## The Development and Optimization of a Deep-Learning Strategy for COVID-19 Classification in Chest X-ray Radiography

by

#### Dalton Griner

A dissertation submitted in partial fulfillment of the requirements for the degree of

> Doctor of Philosophy (Medical Physics)

at the University of Wisconsin–Madison 2023

Date of final oral examination: May 17, 2023

The dissertation is approved by the following members of the Final Oral Committee: Guang-Hong Chen, Professor, Medical Physics Michael A. Speidel, Associate Professor, Medical Physics Ke Li, Associate Professor, Medical Physics John W. Garrett, Assistant Professor, Radiology Ran Zhang, Assistant Scientist, Medical Physics

© Copyright by Dalton Griner May 17, 2023 All Rights Reserved If I have seen [anything at all], it is by standing on the shoulders of giants.

— Sir Isaac Newton Adapted by author

#### Acknowledgments

The quote on the preceding page resonates deeply with me and my pursuit of a PhD. It's no overstatement to say that my graduate journey would have been impossible without the invaluable support from numerous mentors, friends, and family members. Unfortunately, the meager words I can write in this acknowledgments will not come close to conveying my immense gratitude, but nevertheless, I will try.

A mentor is someone who sees more talent and ability within you, than you see in yourself, and helps bring it out of you.

Bob Proctor

I am truly grateful and fortunate to have been surrounded by incredible mentors during my time as a graduate student, each of whom has contributed to my growth and development. Many of these influential individuals serve on my dissertation committee, and I would like to recognize the impact each has had on my life.

Mike Speidel ignited my passion for X-ray imaging through his outstanding 567 course. This class laid the groundwork for the majority of my research projects as a graduate student. Mike has consistently been a patient, approachable mentor, and a friendly presence with whom I could always engage in conversation. Thank you, Mike; your teaching and personal discussions have meant more than you could imagine.

I owe much of my decision to attend Wisconsin and join the CT/X-ray group to John Garrett. He was among the first people I encountered during my interview, and to say that I was (and still am) awestruck by him would be an understatement. From that moment, I aspired to emulate him in my own career. Regrettably for John, he occupied the cubicle in front of me during my first year of graduate school, and if I had paid him a dollar for every question I posed, he would be comfortably retired in a warmer locale by now. Thank you, John, for patiently addressing my endless inquiries, serving as a role model, and being a true friend.

Ke Li has played a pivotal role in my growth as both a student and a researcher. Collaborating with Ke on various tasks—ranging from working in the machine shop and collecting data to analyzing results and engaging in insightful conversations—has been a truly remarkable and enjoyable experience. Ke has devoted countless hours and unwavering patience in guiding me to become a more skilled researcher, assisting me in every aspect of my graduate journey even though he isn't my direct advisor. Thank you, Ke, for the privilege of working alongside you and for the kindness you've shown me. I am immensely grateful for everything you have done for me.

Words fall short in expressing the profound impact Ran Zhang has had on me, especially with regard to this thesis. Every word and figure in this work bears the mark of Ran's influence and guidance. Ran is one of the most brilliant individuals I have ever encountered, and I am constantly in awe of his capabilities. No problem has proven too challenging for him, whether it involves locating the appropriate GPU drivers or spearheading the majority of innovations in this thesis. Ran is unfailingly approachable, greeting me with a smile whenever I knock on his door (which is a daily occurrence), attentively listening to my ideas and thoughts, and patiently explaining and teaching whenever I am mistaken. Collaborating closely with Ran on this thesis over the years has been a true highlight of my graduate experience. Thank you, Ran; this thesis owes its existence to your expertise, and you have inspired me as a researcher, educator, and human being.

Lastly, I am profoundly grateful to Guang-Hong Chen, who recognized and nurtured potential in me that I never knew I had. I am still amazed and eternally appreciative that Guang-Hong saw promise in me and accepted me as a student. A good mentor teaches you how to think, not what to think, and Guang-Hong exemplifies this philosophy flawlessly. Despite initially being one of his least impressive candidates on paper, Guang-Hong transformed my entire learning paradigm by instilling in me his learning style and discipline. Now, six years later, I feel confident in my abilities as a scientist, something I never thought possible. Reflecting on the person I was when I entered the program, I am astounded by how far I've come and it is because of Guang-Hong. The journey has been challenging and required immense patience from Guang-Hong as my advisor, but he has truly changed my life, and words cannot express my gratitude. He has been a mentor not only in academics and research, but also in life, supporting me through every personal challenge I've faced. I owe everything to him. Thank you, Guang-Hong, for believing in me when I didn't believe in myself and for shaping me into the person I am today.

Each member of the CT/X-ray group has served as a source of support and inspiration. Senior members Yinsheng Li, Daniel Cardona, John Hayes, Juan-Pablo Cruz Batista, Juan Montoya, and Xu Ji have set high standards for me to follow, and I am grateful to be associated with them. My fellow classmates Chengzhu Zhang and Mang Feng provided much-needed support during my initial years of classes and research. Their kindness and camaraderie sustained me through challenging times, and I cherish their friendship. Dan Bushe and Kevin Treb, your support has been indispensable, and I couldn't have accomplished this without you both. Xin Tie, you've come to my rescue countless times throughout this project, offering solutions when I was at a loss. Thank you so much for your invaluable assistance over the years. To the junior members, Christian De Caro, Nikou Lei, and Linying Zhan, your talent and enthusiasm have inspired me to be a better role model. It has been a pleasure working with each of you.

To my dear friends Dan and Anna Aderton, you have been the best part of my time in Madison.

Thank you for your unwavering support.

Lastly, I want to express my gratitude to my family. Words cannot capture what they mean to me or the efforts they have made to help me reach this point. My parents have made countless sacrifices to provide me with every opportunity, and I wouldn't be who I am without them. Being a graduate student is challenging, but I believe being married to one is even more demanding. Kayleigh, my soulmate and steadfast support, has put her entire life on hold to stand by me throughout these years. "Support" barely scratches the surface of what she has done for me; every one of my achievements has her handprint on it. This PhD belongs to her just as much as it does to me. Kayleigh has worked selflessly and without recognition as an ICU nurse, even amidst a pandemic and other incredible challenges, tending to the most critically ill and vulnerable patients. She serves as a constant inspiration, embodying the essence of prioritizing patients' needs. She is the most incredible person I know, and I am immensely fortunate that she continues to share her life with me.

As we express our gratitude, we must never forget that the highest appreciation is not to utter words, but to live by them.

- John F. Kennedy

#### Abstract

This thesis scrutinizes the application of Artificial Intelligence (AI), specifically deep learning, in detecting Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) or COVID-19 using Chest X-ray Radiography (CXR). It explores the development process of AI solutions for healthcare, with a focus on addressing the limitations and enhancing the generalizability of deep learning algorithms for COVID-19 detection through CXR. The study examines CXR as a cost-effective, portable, and readily available diagnostic tool, particularly during peak pandemic periods when PCR testing was insufficient. This study highlights the challenge of 'shortcut learning,' where the presence of hidden shortcuts or spurious correlations in training data affects model generalizability and develops methods to detect shortcut features present in datasets. This comprehensive analysis involves curating training data, designing and optimizing models, and evaluating their generalizability and interpretability. The study includes chapters detailing the clinical background of COVID-19, datasets utilized, investigation of shortcut learning, training and evaluation methods, model interpretability, and conclusions for future work in this area. The objective is to advance the integration of AI into clinical settings and improve the accuracy and speed of COVID-19 detection.

## Contents

A	cknov	vledgm	ients		ii
Ał	ostrac	ct			v
Сс	onten	ts			v
Та	bles				ix
Fi	gures				xi
1	Intro	oductio	n		1
2	Bacl	kground	d: An Ove	erview of X-ray Imaging in COVID-19 Diagnosis and the Foundations	
	of D	eep Lea	arning		4
	2.1	Clinic	al Backgr	ound: COVID-19 and Chest X-ray Radiography	4
		2.1.1	The CO	VID-19 Pandemic	4
		2.1.2	The Rol	e of Diagnostic Imaging in COVID-19 Diagnosis	7
	2.2	Techn	ical Back	ground: The Basics of Deep Learning for Image Classification	10
		2.2.1	A Brief	History of Deep Learning	10
		2.2.2	Supervi	sed Learning	15
		2.2.3	How De	eep Neural Networks Learn	17
		2.2.4	Convolu	utional Neural Network Architecture	20
		2.2.5	Training	g and Optimization	22
			2.2.5.1	Loss Function	22
			2.2.5.2	Backpropagation	24
			2.2.5.3	Gradient Descent	25
		2.2.6	Hyperp	arameters	28
		2.2.7	Limitati	ons of Deep Learning	30
	2.3	Major	Research	Problems of this Thesis Work	33

3	Exp	loring a	and Evaluating Chest X-ray Datasets: An Analysis of Dataset Variability and	
	Dist	inctive	ness	35
	3.1	Datas	et Metadata	37
		3.1.1	Patient-Specific DICOM Tags	37
		3.1.2	Image Acquisition DICOM Tags	38
	3.2	Non-O	COVID-19 Chest X-ray Datasets	40
		3.2.1	ImageNet	40
		3.2.2	ChestX-ray14	40
		3.2.3	MIMIC Chest X-ray	41
	3.3	COVI	D-19 Chest X-ray Datasets	42
		3.3.1	Henry Ford COVID-19 Dataset	42
		3.3.2	Henry Ford Temporal COVID-19 Dataset	46
		3.3.3	Valencian Region Medical ImageBank COVID-19+ Dataset	49
		3.3.4	University of Wisconsin-Madison COVID-19 Dataset	53
		3.3.5	Medical Imaging and Data Resource Center COVID-19 Dataset	56
		3.3.6	COVIDx CXR Dataset	60
	3.4	Analy	rsis and Comparison of COVID-19 Datasets	61
		3.4.1	Comparison of Image Contrast	63
		3.4.2	Comparison of Edges and High-Frequency Content	64
		3.4.3	Deep Learning Feature Extraction	70
		3.4.4	Outlier Detection	80
	3.5	Discu	ssion	95
4	Unc	covering	g Hidden Patterns: Analyzing Shortcut Learning in COVID-19 Chest X-ray	
	Data	asets		96
	4.1	The C	OVID-19 Pandemic: A Perfect Storm for Shortcut Learning	98
	4.2	Shorte	cut Features in COVID-19 CXR Datasets	101
		4.2.1	Dataset Metadata Shortcuts	101
		4.2.2	Dataset Source Shortcuts	104
	4.3	Demo	onstration of Intrinsic Shortcut Features in CXR Imaging	109
	4.4	Detec	tion of Intrinsic Shortcut Features in CXR Imaging	116
		4.4.1	Training and Certification of Shortcut Detective Models	117
		4.4.2	Certified Shortcut Detective Investigations on COVID-19 CXR Datasets	122
		4.4.3	Using Shortcut Detectives To Explain Generalizability	124
	4.5	Discu	ssion	126
5	Mas	stering	Generalizability: Training and Optimizing a Robust Deep Learning Network	
	for (	COVID	-19 Classification	129
	5.1	The E	nigma of Deep Neural Network Generalizability	131
	5.2	Gener	alizability of a Single-Source Dataset	133

		5.2.1	Preparing High-Quality Training and External Evaluation Datasets	134
		5.2.2	Model Evaluation and Generalizability	135
	5.3	Influe	ntial Factors in Model Generalization	136
		5.3.1	Network Architecture and Image Size	136
		5.3.2	Disease Severity Measured by Lung Opacity Score	138
		5.3.3	Patient and Imaging Characteristics	140
		5.3.4	Dataset Size and Pretraining	143
			5.3.4.1 The Value of Pretraining for Small Datasets	146
	5.4	Discu	ssion	152
6	Dece	oding A	AI Decisions: Employing Saliency Methods to Illuminate Model Predictions	155
	6.1	An Ov	verview of Common Saliency Methods	157
		6.1.1	Integrated Gradients	157
		6.1.2	Gradient SHAP	158
		6.1.3	Deep Learning Important FeaTures	159
		6.1.4	Occlusion	160
		6.1.5	Gradient-Weighted Class Activation Mapping	161
		6.1.6	Guided Grad-CAM	163
		6.1.7	Counterexamples	164
	6.2	Analy	rsis of Trained Models Using Saliency	165
		6.2.1	BIMCV+/HF-Model	165
		6.2.2	Contrast and Sharpness Shortcut Models	168
		6.2.3	COVIDx Model	168
		6.2.4	Generalizable HF Model	170
	6.3	Alterr	native Saliency Methods: Beyond Heatmaps in Deep Learning Interpretability	173
		6.3.1	Feature Map Activation	173
		6.3.2	Feature Visualization	174
	6.4	Dimer	nsionality Reduction and Analysis of Extracted Features from a Trained COVID-	
		19 Mo	odel	177
	6.5	Discu	ssion	180
7	Con	clusion	s, Limitations, and Future Directions	182
	7.1	Concl	usions	183
		7.1.1	Dataset Curation and Quality Analysis	183
		7.1.2	Shortcut Feature Detection	183
		7.1.3	Training and Evaluation of a Generalizable Network	185
	7.2	Limita	ations and Future Work	187
A	Dee	p Neur	al Network Architecture, Parameters, and Training Strategies	189
	A.1	Netwo	ork Architecture	189

		A.1.1	VGG-16	190
		A.1.2	ResNet-101	192
		A.1.3	DenseNet-121	195
		A.1.4	EfficientNet-V2 Medium	198
		A.1.5	Swin Transformer	201
		A.1.6	ConvNeXt	204
	A.2	Trainiı	ng Strategies	207
		A.2.1	Training/Validation Split and Class Imbalance	207
		A.2.2	Ensemble Learning	210
		A.2.3	Model Pretaining	210
		A.2.4	Learning Rate Adjustment and Regularization	211
		A.2.5	Optimizer	212
		A.2.6	Image Augmentation	214
В	Ana	lysis of	Deep Learning Extracted Features from CXR-Trained Models	215
C	Add	itional	Saliency Heatmap Examples	238
Re	feren	ces		269

## Tables

2.1	The British Society of Thoracic Imaging (BSTI) report for the plain chest radiographic appearances of potential COVID-19 cases and The Radiological Society of North America (RSNIA) classification of the CT appearance of COVID 19 for standardized reporting
	language
3.1	A summary of the demographic information for the HF COVID-19 dataset used in this
	work
3.2	A summary of the demographic information for the HF temporal COVID-19 dataset used
	in this work
3.3	A summary of the demographic information for the BIMCV COVID-19 dataset used in
	this work
3.4	A summary of the demographic information for the UW COVID-19 dataset used in this
	work
3.5	A summary of the demographic information for the MIDRC COVID-19 dataset used in
	this work
4.1	A comparison of internal test performance (AUC and 95% confidence interval) and
	external test performance was conducted for models trained on biased training datasets. 103
4.2	A comparison of model performance (AUC and 95% confidence interval) for the reference
	shortcut-free model, simulated sharpness, contrast, and marker models as well as their
	combinations.
4.3	Five distinct model architectures that are widely employed for image classification and
	demonstrate state-of-the-art performance on ImageNet classification tasks were used as
	shortcut detectives in this work
4.4	Results of the certification exams for the models listed in Table 4.3
4.5	Results of the trained shortcut detectives for the COVID-19 and RoentGen datasets 125

4.6	To demonstrate that datasets with pronounced shortcuts, as identified by the shortcut
	detectives, suffer from poor generalizability, a comparative analysis of performance was
	carried out between two models: one trained on the COVIDx dataset and the other on
	the HF dataset
5.1	Comparison of the HF trained model performance on internal and external test sets 136
5.2	Results for each architecture and input image size on the test sets
5.3	The performance of the HF-trained COVID-19 detection model evaluated across different
	patient characteristics, including sex, view position, and modality
5.4	The performance of the HF-trained COVID-19 detection model evaluated across patient
	race. Further studies remain to determine whether race is truly correlated with model
	performance
5.5	The generalization performance of models with different training dataset sizes as shown
	in Figure 5.6

# Figures

2.1	An atom-by-atom model of the coronavirus SARS-CoV-2 overlaid on an electron micro-	
	scope image of the virus	5
2.2	The polymerase chain reaction is a method widely used to rapidly make copies (complete	
	or partial) of a specific DNA sample	6
2.3	COVID-19 infects the respiratory system and damages the lungs. To fight off the infection,	
	the immune system causes inflammation, which can also cause damage and allow fluid	
	to leak into the small air sacs of the lungs. This is called pneumonia and is the most	
	common radiographic feature in COVID-19 CXR	8
2.4	A healthy chest X-ray showcasing the major lung zones and relatively radio-transparent	
	lung tissue. Pneumonia is an inflammatory condition where the alveoli fill with fluid	
	resulting in a more opaque appearance in CXR. Example of ground glass opacities where	
	the underlying vasculature and borders are not obscured. Example of consolidation	
	where the opacities obscure the underlying anatomy and organ borders	9
2.5	The hierarchy of artificial intelligence, machine learning, and deep learning	11
2.6	Warren Sturgis McCulloch was an American neurophysiologist and cybernetician who,	
	along with Walter Pitts, created computational models based on mathematical algorithms	
	called threshold logic. A biological neuron compared to an artificial neural network	12
2.7	One of the first convolutional neural networks, the Neocognitron developed by Kunihiko	
	Fukushima, compared to a modern DenseNet convolutional neural network	13
2.8	An image created by the author using a cycleGAN where a chest radiograph is reimagined	
	in different modern art styles.	15
2.9	Classification is the process of finding or discovering a model or function which helps	
	separate the data into multiple categorical classes, i.e., discrete values. Regression is the	
	process of finding a model or function for distinguishing the data into continuous real	
	values instead of using classes or discrete values	17

2.10	A neural network is composed of an input layer, hidden layers, and an output layer. Each	
	layer consists of several neurons which perform simple mathematical operations. The	
	layers are connected through weighted edges, representing the strength of the connections	
	between neurons. During the learning process, input data is passed through the network	
	in a forward direction from the input layer to the output layer. Each neuron in the	
	network calculates a weighted sum of its inputs, applies an activation function, and	
	passes the result to the next layer	18
2.11	Activation functions in artificial neural networks introduce non-linearity into the models	
	and determine whether a neuron should be activated. Common activation functions	
	include sigmoid, hyperbolic tangent, and ReLU	19
2.12	A $5 \times 5 \times 1$ image convolved with a $3 \times 3 \times 1$ kernel with a stride of 2. The kernel slides	
	across the image (convolution), and an element-wise multiplication operation is taken	
	between the kernel and the portion of the image over which the kernel is hovering	21
2.13	Relatively simple compared to modern CNNs, AlexNet highlights the main components	
	of a CNN: alternating convolutional and pooling layers followed by fully connected	
	layers and an output layer	22
2.14	An example of the backpropagation algorithm, which computes the gradient of the cost	
	function $C$ to certain parameters. This information is used to adjust the weights and	
	biases in the direction of the negative gradient to minimize the loss, effectively reducing	
	the error in the network's predictions.	26
2.15	Gradient descent is a first-order iterative optimization algorithm used to minimize an	
	objective function, typically used for finding the minimum of a convex function or a local	
	minimum of a non-convex function.	27
2.16	Feature visualization answers questions about what a network, or parts of a network,	
	are looking for by generating images. Above are some examples of features learned by	
	GoogLeNet trained on the ImageNet dataset. This technique can help understand what	
	the DNN learned, but often the features are abstract and not easily interpreted	31
2.17	Deep learning models are prone to learning the most discriminating features, even if	
	these features are not useful for the overall task. Shortcut learning occurs when spurious	
	correlations exist between the image features irrelevant to the task and the corresponding	
	training labels	32
3.1	In September 1956, IBM launched the 305 RAMAC with a disk storage unit weighing	
0.12	over 2.000 lbs and storing approximately 5 megabytes of data. Today, micro-SD cards	
	over a terabyte are commonly used in electronics, over 200.000 times more storage and	
	orders of magnitude faster than the RAMAC.	36
3.2	Comparison of AP and PA view positions. Due to more geometric magnification, AP	
	imaging results in an image where the heart and other mediastinal structures appear	
	larger than they actually are. Comparison of DX and CR. DX uses a digital detector to	
	directly capture the X-ray image, while CR uses a photostimulable phosphor plate	39

3.3	The Henry Ford Health System covers the greater Detroit region in Michigan, USA.	43
3.4	Dataset population characteristics of the Henry Ford COVID-19 dataset.	44
3.5	Average image (with 25 random examples) from 5,000 randomly sampled images from	
	the HF dataset for COVID-19 positive (left) and COVID-19 negative (right).	45
3.6	Average image pixel intensity distribution for the HF dataset and the four most contribut-	
	ing vendors Agfa, Carestream, GE Healthcare, and Konica Minolta.	45
3.7	Dataset population characteristics of the HF temporal COVID-19 dataset.	47
3.8	Average image (with 25 random examples) from 600 randomly sampled images from the	
	HF temporal dataset for COVID-19 positive and COVID-19 negative.	48
3.9	Average image pixel intensity distribution for the HF temporal dataset and the four most	
	contributing vendors Agfa, Carestream, GE Healthcare, and Konica Minolta.	48
3.10	The BIMCV dataset was collected from 11 hospitals in the Valencian Region, Spain	50
3.11	Dataset population characteristics of the BIMCV COVID-19 dataset	51
3.12	Average image (with 25 random examples) from 600 randomly sampled images from the	
	BIMCV dataset for COVID-19 positive and COVID-19 negative	52
3.13	Average image pixel intensity distribution for the BIMCV dataset and the four most	
	contributing vendors Agfa, Carestream, Konica Minolta, and Philips	52
3.14	UW Health Hospital in Madison, Wisconsin.	53
3.15	Average image (with 25 random examples) from 600 randomly sampled images from the	
	UW dataset for COVID-19 positive and COVID-19 negative.	54
3.16	Average image pixel intensity distribution for the UW dataset and the four most con-	
	tributing vendors Philips, Fujifilm, Siemens, and Samsung	54
3.17	Dataset population characteristics of the UW COVID-19 dataset	55
3.18	MIDRC is a multi-institutional collaborative initiative hosted at the University of Chicago	
	with images from hundreds of institutions across the United States	56
3.19	Average image (with 25 random examples) from 6,000 randomly sampled images from	
	the MIDRC dataset for COVID-19 positive and COVID-19 negative	57
3.20	Average image pixel intensity distribution for the MIDRC dataset	57
3.21	Dataset population characteristics of the MIDRC COVID-19 dataset.	59
3.22	Average image (with 25 random examples) from 5,000 randomly sampled images from	
	the COVIDx dataset for COVID-19 positive (left) and COVID-19 negative (right)	60
3.23	Average image pixel intensity distribution for the COVIDx dataset	61
3.24	Table of the available metadata for the COVID-19 datasets used in this work.	62
3.25	Average pixel intensity distribution for COVID-19 positive and negative images based	
	on data source.	63
3.26	Some examples of image differences based on dataset source, including image background	
	color, image text, or markers covering text, as well as window and leveling protocols.	64

3.27	A comparison of the average images (obtained by averaging pixel intensities over all	
	images) for each dataset. The left half of each individual image corresponds to the row	
	label and the right to the column label	65
3.28	A comparison of the average image difference images (obtained by averaging pixel	
	intensities over all images) for each dataset. Each individual image is obtained by taking	
	the row label and subtracting the column label	65
3.29	A comparison of the standard deviation images (obtained by taking the standard deviation	
	of pixel intensities over all images) for each dataset. The left half of each individual	
	image corresponds to the row label and the right to the column label	66
3.30	A comparison of the standard deviation difference images (obtained by taking the	
	standard deviation of pixel intensities over all images) for each dataset. Each individual	
	image is obtained by taking the row label and subtracting the column label	66
3.31	Example CXR and the extracted edges using the Canny method	67
3.32	The mean edge intensity distribution, obtained by averaging the binary edge image, for	
	the COVID-19 datasets used in this work.	68
3.33	Comparison of the average Fourier transform for each dataset (log transform taken to	
	compress dynamic range). The left half of each individual image corresponds to the row	
	label and the right to the column label	69
3.34	Comparison of the average Fourier transform difference for each dataset. Each individual	
	image is obtained by taking the row label and subtracting the column label	69
3.35	An example of how a CNN can be used to extract high-level features from images. The	
	convolutional layers extract image features which are then classified using fully connected	
	layers. By removing the fully connected layers, the high-dimensional features extracted	
	by the network can be obtained to be further analyzed	70
3.36	UMAP clustering of the ImageNet pretrained ResNet-101 extracted features of the	
	COVID-19 datasets labeled according to disease label, dataset source, and K-means	
	cluster index	72
3.37	Unsupervised and supervised UMAP reduction of the ImageNet pretrained ResNet-101	
	extracted features of the COVID-19 datasets labeled according to dataset source and	
	K-means cluster centroids ( $K = 7$ )	73
3.38	To explore what features are prominent in the UMAP reduction, 25 randomly sampled	
	images are shown for six different areas in the total data main cluster, six satellite clusters,	
	as well as the seven clusters grouped by K-means clustering.	75
3.39	UMAP clustering of the NIH ChestX-ray14 pretrained EfficientNet extracted features of	
	the COVID-19 datasets labeled according to disease label, dataset source, and K-means	
	cluster index	76
3.40	UMAP reduction of features extracted from the COVID-19 datasets using the ImageNet	
	pretrained ResNet-101 model, with labels according to dataset source. Each dataset	
	source is displayed individually to enhance the visualization of their distinct structures.	77

xv

3.41	To explore what features are prominent in the UMAP reduction, 25 randomly sampled	
	images (framed by the label color) are shown for the one outlier cluster as well as the	
	seven clusters grouped by K-means clustering.	78
3.42	Unsupervised and supervised UMAP reduction of the ChestX-ray14 pretrained Efficient-	
	Net extracted features of the COVID-19 datasets labeled according to dataset source and	
	K-means cluster centroids ( $K = 7$ )	79
3.43	An example of random partitioning in a 2D dataset of normally distributed points for a	
	non-anomalous point and for a point that's more likely to be an anomaly. It is apparent	
	from the plots how anomalies require fewer random partitions to be isolated, compared	
	to normal points.	81
4.1	Clever Hans, a horse who demonstrated mathematical genius which was later discovered	
	to be giving the right answers by watching the reactions of the people who were watching	
	him	97
4.2	Much like Hans, "Clever AI" has learned to detect a metal token that radiology technicians	
	place on the patient in the corner of the image instead of the true imaging features of	
	pneumonia.	99
4.3	To identify potential shortcut features in COVID-19 CXR classification, biased training	
	datasets containing various shortcuts were created.	102
4.4	Randomly selected images corresponding to the training datasets described in Table 4.1.	
	Apart from patient sex, primarily due to differences in breast tissue, none of the shortcut	
	variables exhibit obvious discriminating features upon inspection	104
4.5	When using data from different sources as the classification label, a network will simply	
	learn the differences between the datasets and fail to generalize to outside data. By	
	removing the correlation between the dataset source and classification label, the network	
	learned more generalizable features and relied less on dataset-specific shortcuts	105
4.6	Even with 70% of the image masked out, when using data from different sources as the	
	classification label, a network will primarily learn the differences between the datasets	
	and fail to generalize to outside data	106
4.7	Efforts to mitigate shortcut learning include methods to harmonize data between dif-	
	ferent datasets to remove biases. Histogram equalization addresses image contrast and	
	window/leveling differences, and segmentation can isolate relevant features	107
4.8	In comparison to the model without segmentation and histogram equalization, the model	
	trained on equalized and segmented images still exhibits considerable shortcut learning.	108
4.9	For chest CXRs employed in COVID-19 classification tasks, image contrast and sharpness	
	may vary across hospitals due to a variety of reasons. These inherent characteristics of	
	the CXRs can serve as shortcuts.	110
4.10	Methods for adjusting the image sharpness and contrast to the COVID-19 positive images.	
	Contrast was adjusted using parameter $c = 0.02$ and sharpness was adjusted using	
	parameter $\alpha$ = 1.1	111

4.11	Methods for adjusting the image sharpness and contrast to the COVID-19 positive images.112	
4.12	Grad-CAM activation maps were generated for several examples from the reference,	
	sharpness, contrast, and marker trained models	
4.13	Examples of contrast (c) and sharpness (s) adjusted images according to the process	
	shown in Figure 4.10. These adjustments only introduce very subtle changes to the	
	original image, as demonstrated in these images	
4.14	In this analogy, the shapes represent CXR disease features, while the color symbolizes	
	contrast or sharpness shortcuts. A shortcut detective trained on added sharpness or	
	contrast features (analogous to the color) should not be able to differentiate between	
	COVID-19 positive/negative features (analogous to the shape)	
4.15	Overview of the training and certification of shortcut detective deep neural network	
	models	
4.16	Real CXR images from the MIMIC dataset compared to synthetic CXRs generated by	
	the RoentGen model, a GAN that produces highly realistic images that are visually	
	challenging to differentiate from authentic CXR images	
4.17	The detective framework effectively serves as a litmus test for detecting shortcut features	
	in a dataset. Analogous to the color change in litmus paper that helps determine whether	
	a solution is acidic or basic (alkaline), the shortcut detectives function like a litmus sheet	
	when tested on a dataset (solution), and the AUC measurement (color change) indicates	
	whether the dataset is biased toward a particular shortcut feature 127	,
	mether the dualet is blaced toward a particular shorear feature	
5.1	The growth of AI-approved medical devices and algorithms by the FDA has surged in	
5.1	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than	
5.1	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
5.1 5.2	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
5.1 5.2	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
<ul><li>5.1</li><li>5.2</li><li>5.3</li></ul>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
5.1 5.2 5.3	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
5.1 5.2 5.3	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
<ul><li>5.1</li><li>5.2</li><li>5.3</li><li>5.4</li></ul>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
<ul> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> <li>5.5</li> </ul>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
<ul> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> <li>5.5</li> </ul>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
<ul> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> <li>5.5</li> <li>5.6</li> </ul>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
<ol> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> <li>5.5</li> <li>5.6</li> </ol>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
<ol> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> <li>5.5</li> <li>5.6</li> <li>5.7</li> </ol>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	1
<ol> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> <li>5.5</li> <li>5.6</li> <li>5.7</li> </ol>	The growth of AI-approved medical devices and algorithms by the FDA has surged in         the past decade. Of the 521 submissions the FDA has authorized to date, greater than         75% have been in radiology.       130         For reasons not well understood, DNNs generally do not memorize real data but instead         learn simple patterns before resorting to memorization.       132         Excluding asymptomatic cases allows the training dataset to focus on relevant and         informative cases, ultimately leading to more accurate and robust disease classification         models.       139         The opacity score distribution for the HF training dataset, as well as the test sets.       140         Using the opacity scores from the trained opacity model, the AUC from the HF-trained       141         Model performance in terms of AUC on the four test sets as a function of the training       145         Loss curves for a DenseNet model trained on the HF dataset under three conditions: no       147	
<ol> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> <li>5.5</li> <li>5.6</li> <li>5.7</li> <li>5.8</li> </ol>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
<ol> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> <li>5.5</li> <li>5.6</li> <li>5.7</li> <li>5.8</li> </ol>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	
<ol> <li>5.1</li> <li>5.2</li> <li>5.3</li> <li>5.4</li> <li>5.5</li> <li>5.6</li> <li>5.7</li> <li>5.8</li> </ol>	The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology	

5.9	Model performance in terms of AUC on the four test sets as a function of the training data size and ImageNet and NIH pretraining	151
5.10	Utilizing the strategies described in this thesis, Ran Zhang, Ph.D., and the author submitted a model to the MIDRC COVIDx Challenge.	154
6.1	Integrated Gradients is a technique for attributing a classification model's prediction to its input features. Example heatmaps for the marker-trained model with a positive case	4 = 0
6.2	Gradient SHAP is an interpretability method that combines the principles of IG and Shapley values to provide explanations for deep learning models. The method highlights the marker as a key feature in the positive case. The baselines used were a random sample	158
6.3	of negative images, which did not contain the added marker	159
6.4	added marker	160
6.5	different parts of the image and observing the impact on the model's prediction Grad-CAM generates a coarse heatmap (corresponding to the output size of the target layer) that highlights the areas in the input image that contribute the most to the model's	161
6.6	prediction. The heatmap is generally upsampled to the original image size Guided Grad-CAM is a combination of two interpretability methods, Grad-CAM and Guided Backpropagation, to address the localization issues with Grad-CAM. The method highlights the marker with much more accuracy as a key feature in the positive case and also highlights the shoulder regions in the negative case suggesting that this region was	162
6.7	important to the model's prediction, which follows intuition	163
	seemingly irrelevant features.	164

6.8	Example heatmaps for the saliency methods described in this chapter are shown for each	
	of the three BIMCV+/HF- models outlined in Chapter 4. The saliency maps for the top	
	two panels indicate that shoulder markers may serve as a shortcut in the BIMCV+/HF-	
	training dataset. However, when these features are removed through segmentation,	
	shortcut learning still persists, and the saliency maps provide no immediate explanation	
	for what the shortcut may be	166
6.9	Example heatmaps for the sharpness and contrast shortcut models developed in Chapter 4.	
	Drawing conclusions from these heatmaps is challenging because the added contrast	
	and sharpness are not only nearly invisible but also globally distributed	169
6.10	The COVIDx dataset is predisposed to shortcut learning due to the fact that most negative	
	cases stem from a pre-pandemic dataset, while positive cases originate from various	
	sources with differing quality levels. In these examples, image markers are accentuated	
	by the saliency maps suggesting these may serve as potent shortcut features	170
6.11	Example heatmaps for the five separate models in the HF-trained ensemble developed in	
	Chapter 5. The saliency maps are consistent across each model, suggesting that the same	
	features are learned by every individual model.	171
6.12	Example heatmaps for the HF-trained DenseNet-121, EfficientNet, and Swin Transformer	
	models. The saliency maps remain consistent across each network architecture, implying	
	that the models learn the same features	172
6.13	Example feature map activations and convolutional kernels from the HF-trained	
	DenseNet-121 model. The intensity of each pixel in the extracted feature map cor-	
	responds to the activation strength at that location. Brighter pixels indicate higher	
	activation, suggesting the presence of the corresponding feature.	174
6.14	Feature visualization aims to reveal the features and patterns that the model has learned	
	to recognize during the training process. The process typically involves optimizing input	
	images to maximize the activation of specific neurons or layers within the model. These	
	images typically represent the types of patterns that excite the selected neurons or layers	
	the most	175
6.15	While primarily an artistic tool, DeepDream feature visualization also offers insights into	
	the features and patterns learned by CNNs, contributing to the understanding of their	
	internal representations.	176
6.16	To further explore HF-trained models, feature vectors from test images in external datasets	
	were analyzed using UMAP dimensionality reduction and K-Means clustering. Different	
	model architectures uniquely embed features, leading to diverse UMAP graph shapes.	
	Distinct clusters in the graph represent groups of similar high-dimensional data points,	
	while denser regions imply greater similarity.	178
6.17	When examining the sample images associated with K-Means clustering labels, no	. –
	discernible patterns in cluster groupings are detected	179
A.1	VGG architecture.	190

A.2	Resnet architecture	<i>)</i> 2
A.3	DenseNet architecture	<i></i> 95
A.4	Structure of MBConv and Fused-MBConv	98
A.5	Swin Transformer architecture	)1
A.6	ConvNeXt architecture	)4

# Chapter 1

### Introduction

Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), commonly known as COVID-19, has infected approximately 700 million people worldwide, leading to nearly 7 million fatalities as of April 2023.<sup>1</sup> The reverse transcription polymerase chain reaction (RT-PCR) test serves as the gold standard for identifying and diagnosing COVID-19 infections. In addition to PCR testing, diagnostic imaging has played a crucial role in detecting and managing suspected COVID-19 patients.

Chest X-ray radiography (CXR) is widely employed alongside PCR testing for diagnosis and monitoring treatment responses in suspected COVID-19 cases.<sup>2,3</sup> When compared to other imaging techniques, such as computed tomography (CT), CXR offers several advantages, particularly during a global pandemic: (1) CXR is readily available in most hospitals and clinics worldwide and is significantly more cost-effective than alternative imaging methods; (2) CXR is fast, portable, and easy to sterilize, facilitating high patient throughput while reducing the risk of disease transmission.

In certain situations, particularly during the pandemic's peak, PCR response times were insufficient for timely triaging and management of the overwhelming number of suspected COVID-19 cases. As a result, CXR became an alternative tool to address this challenge.

The rapid progression of COVID-19 and the sudden influx of cases placed immense pressure on radiologists to quickly learn the distinct and common imaging features of the disease. This challenge was exacerbated by the nonspecific radiographic presentation of COVID-19, which shares similarities with other viral illnesses. Several retrospective studies<sup>4–6</sup> have assessed the diagnostic performance of radiologists during the pandemic, revealing considerable variability in their performance based on personal experience and disease prevalence. Notably, radiologists exhibited relatively poor performance at the pandemic's onset.

Artificial intelligence (AI) algorithms, particularly deep learning, were rapidly implemented for COVID-19 detection to address these limitations and alleviate the burden on radiologists. Deep learning methods have demonstrated exceptional performance in various image classification tasks, often surpassing human capabilities.<sup>7</sup> These techniques also offer the advantage of a quick and tireless learning process, allowing for rapid training and implementation—an essential feature in a pandemic setting. Numerous AI models have been developed for COVID-19 detection,<sup>8</sup> with some boasting near-perfect accuracy—an impressive improvement over human radiologists.

However, these models have not yet seen widespread clinical adoption due to a significant decrease in performance when applied to "external data" (i.e., data from other vendors, hospitals, or imaging protocols). Recent attention has focused on "shortcut learning" as a key factor in model generalizability. Shortcut learning occurs when the training dataset contains hidden shortcuts or spurious correlations between image features unrelated to the task and corresponding training labels. Additionally, limited training and test data can result in poor generalization. Addressing these shortcomings is crucial for the successful integration of deep learning algorithms into clinical settings.

This thesis explores the process of developing AI solutions for healthcare, using CXR-based COVID-19 detection as a foundation. The study encompasses the entire pipeline, from curating training data and designing models to evaluating their generalizability and interpretability. The thesis is organized as follows:

Chapter 2 delves into the clinical background of COVID-19 and the utilization of medical imaging for diagnosing and managing the disease. This chapter also examines AI, specifically deep learning, in medical imaging analysis, discussing current achievements and challenges for eventual clinical implementation.

Chapter 3 introduces the CXR datasets employed in this research. The development, training, optimization, and comprehensive testing of a deep learning model for COVID-19 classification hinge on large and diverse datasets. This chapter elaborates on and highlights the features of both COVID-19 positive and negative CXR datasets.

Chapter 4 investigates shortcut learning in medical imaging AI and potential shortcuts in the datasets mentioned in Chapter 3. A general framework for detecting and identifying potential

shortcut fatures is presented.

Chapter 5 describes the methods for training and evaluating a deep learning model, analyzing its generalizability. The impact of training data size on model performance and generalizability is also explored, addressing the question of how large "big data" needs to be. The effects of pretraining with different datasets are also examined.

Chapter 6 concentrates on model interpretability, employing saliency and other explainable AI techniques to create a more transparent and trustworthy model for clinical applications.

Finally, Chapter 7 summarizes the key findings and conclusions of this thesis, offering insights into future work in this area.

# Chapter 2

# Background: An Overview of X-ray Imaging in COVID-19 Diagnosis and the Foundations of Deep Learning

#### 2.1 Clinical Background: COVID-19 and Chest X-ray Radiography

This section provides a clinical overview of the COVID-19 pandemic to underscore the significance of this thesis work. Furthermore, the role of diagnostic imaging in detecting the disease and the radiographic manifestations of COVID-19 will be discussed to clarify the nuances involved in the deep learning approach presented in this thesis.

#### 2.1.1 The COVID-19 Pandemic

In late 2019, a new coronavirus emerged as the source of a cluster of severe pneumonia cases in Wuhan, a city in China's Hubei province. The virus, now known as severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), led to a disease that the World Health Organization (WHO) named COVID-19 in February 2020, which stands for coronavirus disease 2019. This disease quickly spread across Europe and the rest of the world, resulting in the most severe pandemic in the past century. The WHO officially declared the outbreak a pandemic on March 11, 2020. Although no consensus has been reached on a definitive  $R_0$  value (basic reproduction number),<sup>9</sup> studies and observations have shown COVID-19 to be more contagious and deadly than the seasonal flu,<sup>10</sup> causing widespread economic and social disruptions. Moreover, as the pandemic has evolved, multiple variants of the virus have emerged, and COVID-19 is likely to remain a differential



diagnosis for years to come as the disease transitions to becoming endemic.

While understanding of the disease is continually evolving, some epidemiologic and radiographic characteristics of COVID-19 have emerged from studies conducted throughout the pandemic. Studies have estimated a mean incubation period for COVID-19 of 6.0 days, <sup>11</sup> and a vast majority of symptomatic individuals showed symptoms within 12 days. <sup>12</sup> The majority of COVID-19 patients were asymptomatic or mild cases <sup>13</sup> with more severe and symptomatic cases unsurprisingly affecting the older population, with studies establishing the median age between 47 and 59 years. <sup>14,15</sup> COVID-19 typically presents with systemic (multiple organ dysfunction, MODS-SARS-CoV-2)<sup>16</sup> and/or respiratory manifestations. <sup>15,17</sup>

Leading radiological societies have established reverse-transcription polymerase chain reaction (RT-PCR) testing as the reference standard for COVID-19 diagnostic screening.<sup>18</sup> This allows searching for specific sequences from the viral genome: E (envelope protein gene), N (nucleocapsid protein gene), and RdRP (RNA-dependent RNA polymerase gene).<sup>19</sup> The PCR test is highly specific, and pooled sensitivity has been reported to be 89%,<sup>20</sup> however sensitivity as low as 60-70% has also been reported<sup>21</sup> making false negatives a clinical concern, and several negative tests may be required to be confident about excluding the disease. The sensitivity of a PCR test is also predicated on time since exposure to COVID-19, with a false-negative rate of 100% on the first day after exposure, dropping to 67% on the fourth day. On the day of symptom onset (around four days after exposure



**Figure 2.2:** The polymerase chain reaction is a method widely used to rapidly make millions to billions of copies (complete or partial) of a specific DNA sample, allowing scientists to take a tiny sample of DNA and amplify it (or a part of it) to a large enough amount to study in detail. Image from Wikipedia https://en.wikipedia.org/wiki/Polymerase\_chain\_reaction.

for most of the population), the false-negative rate remains at 38%, and it reaches its low of 20% about three days after symptoms begin (8 days post-exposure). After this, the false-negative rate starts to climb again, reaching 66% on day 21 after exposure.<sup>22</sup>

As the virus began to spread globally in March 2020, an unanticipated issue arose due to a worldwide shortage of chemical reagents required for PCR testing. The rapid progression of the disease and the large number of individuals requiring testing led to PCR testing becoming a significant bottleneck in diagnosing and triaging suspected COVID-19 patients. Researchers quickly developed and deployed alternative molecular detection methods in response to this shortage.<sup>23–26</sup> However, even two years after the pandemic, the COVID-19 testing supply chain, especially in the United States, struggled to maintain consistent and equitable testing across the population. This has hindered the public health response and efforts to control the spread of the virus.<sup>27</sup>

Numerous factors contributed to the intensity of the COVID-19 pandemic. First, although the virus responsible for COVID-19 is not as lethal as Ebola or HIV, nor is it as contagious, it posed a unique threat due to its novelty. The virus had ample opportunity to spread without existing immunity or immediate vaccines. The infection targets the upper respiratory system, allowing transmission through coughs, sneezes, and other respiratory interactions. Asymptomatic carriers further complicated containment efforts, as they unknowingly contributed to the virus's propagation. The sudden emergence of the disease quickly overwhelmed mask production and PCR testing capacities, leading to rapid and widespread infection. Hospitals struggled to manage the influx of patients, lacking the necessary resources to address the crisis adequately. Finally, the novel coronavirus's unprecedented nature and rapid spread caught the world off guard, exacerbating the pandemic's severity.

#### 2.1.2 The Role of Diagnostic Imaging in COVID-19 Diagnosis

In addition to PCR testing, diagnostic imaging has been used in the diagnosis and treatment-response monitoring of persons under investigation for COVID-19. While widely used at the beginning of the pandemic due to high patient throughput and PCR test shortages, the radiological community has discouraged thoracic computed tomography (CT) imaging as a screening tool. Instead, it is recommended that CT be reserved for managing patients with COVID-19 with worsening and severe respiratory symptoms.<sup>28</sup> Additionally, in a resource-constrained environment, imaging is indicated for the medical triage of patients with suspected COVID-19 who present with moderate-severe clinical features and a high pretest probability of disease.<sup>29</sup> Chest X-ray radiography (CXR), already widely used for respiratory screening, has several advantages over CT imaging in the setting of a pandemic: it is available in almost every hospital and clinic around the world, chest radiography units are easily protected from contamination, and easier to disinfect after use, and they can also be directly used in a contained clinical environment without having to move patients to a dedicated imaging suite. Although less sensitive than chest CT, chest radiography is typically the first-line imaging modality for patients with suspected COVID-19.<sup>30</sup>

The primary findings of COVID-19 on CXR and CT are those of atypical pneumonia<sup>2,31</sup> or organizing pneumonia<sup>21,32</sup> with the presence of patchy and confluent, band-like ground-glass opacity or consolidation in a peripheral and mid to lower lung zone distribution on a chest radiograph to be highly suggestive of COVID-19 infection.<sup>33</sup> However, chest radiographs may not indicate the disease in early/mild cases. In those COVID-19 cases requiring hospitalization, 69% had an abnormal chest radiograph at the initial time of admission, and 80% had radiographic abnormalities sometime during hospitalization, with most extensive findings about 10-12 days after symptom onset.<sup>30</sup> A summary of the radiographic appearances of potential COVID-19 cases in CXR<sup>34</sup> and CT<sup>35</sup> are presented in Table 2.1.



**Figure 2.3:** COVID-19 is a respiratory illness caused by the virus SARS-CoV-2. The virus typically infects the respiratory system and damages the lungs. To fight off the infection, the immune system causes inflammation, which can also cause damage and allow fluid to leak into the small air sacs of the lungs. This is called pneumonia and is the most common radiographic feature in COVID-19 CXR. Image made in BioRender.

The lungs are the primary organs of the respiratory system, and due to their primary function of gas exchange, they are relatively sparse and do not absorb X-rays as readily as the surrounding tissue and bone of the thoracic cavity. As a result, they appear as two large, dark cavities in a typical chest radiograph and are roughly defined into upper, middle, and lower zones, as well as central and peripheral zones (see Figure 2.4). In the radiologic sense, opacity refers to any area that preferentially attenuates the X-ray beam and appears more opaque than the surrounding area. Pneumonia is an inflammatory lung condition primarily affecting the tiny air sacs known as alveoli. In an infection, the air sacs may fill with fluid or purulent material, resulting in more radio-opaque areas, i.e., opacities, which appear brighter compared to healthy lung tissue. Pneumonia opacities are differentiated from other opacities (tumors, pleural effusion, nodules, etc.) by poorly defined borders and a diffuse, hazy appearance. It is termed consolidation if the opacity obscures the margins of the lungs, heart, and/or underlying vasculature. If the area of increased attenuation does not conceal the local vasculature, it is termed ground-glass opacity and less opaque than

**Table 2.1:** The British Society of Thoracic Imaging (BSTI) report for the plain chest radiographic appearances of potential COVID-19 cases and The Radiological Society of North America (RSNA) classification of the CT appearance of COVID-19 for standardized reporting language.

Radiology Report: Chest Radiography				
Classic/Probable COVID-19	Lower lobe and peripheral predominant multiple opacities that are bilateral			
Indeterminate for COVID-19	Does not fit classic or non-COVID-19 descriptors			
Non-COVID-19	Pneumothorax / lobar pneumonia / pleural effusion(s) / pulmonary edema / other			
Normal	COVID-19 not excluded			
Radiology Report: Computed Tomography				
	• Peripheral, bilateral, GGO +/- consolidation or visible intralobular lines ("crazy paving" pattern)			
Typical Appearance	<ul> <li>Multifocal GGO of rounded morphology +/- consolidation or visible intralobular lines ("crazy paving" pattern)</li> <li>Reverse halo sign or other findings of organizing pneumonia</li> </ul>			
	Absence of typical CT findings and the presence of			
Indeterminate Appearance	<ul> <li>Multifocal, diffuse, perihilar, or unilateral GGO +/-consolidation lacking a specific distribution and are non-rounded or non-peripheral</li> <li>Few very small GGO with a non-rounded and non-peripheral distribution</li> </ul>			
	Absence of typical or indeterminate features and the presence of			
Atypical Appearance	<ul> <li>Isolated lobar or segmental consolidation without GGO</li> <li>Discrete small nodules (e.g. centrilobular, tree-in-bud)</li> <li>Lung cavitation</li> <li>Smoother interlobular septal thickening with pleural effusion</li> </ul>			
Negative for Pneumonia	No CT features to suggest pneumonia, in particular, absent GGO and consolidation			



**Figure 2.4:** (Left) A healthy chest X-ray showcasing the major lung zones and relatively radio-transparent lung tissue. (Middle left) Pneumonia is an inflammatory condition where the alveoli fill with fluid resulting in a more opaque appearance in CXR. (Middle right) Example of ground glass opacities where the underlying vasculature and borders are not obscured. (Right) Example of consolidation where the opacities obscure the underlying anatomy and organ borders. From Cleverley et al., "The role of chest radiography in confirming COVID-19 pneumonia." *BMJ*, 2020.

consolidation.

CXR screening and diagnosis for COVID-19 infection is complicated by the fact that a large percentage of COVID-19 cases may be asymptomatic,<sup>36</sup> and even with symptomatic infection, there exists a spectrum in the severity and presentation of the disease. Furthermore, many patients diagnosed with COVID-19 do not develop pneumonia, and many of the radiographic presentations of the disease are non-specific<sup>37</sup> resulting in a significant challenge in establishing a diagnosis of COVID-19 in CXR. In addition to these factors, radiologists, particularly at the start of the pandemic, had little time to learn the standard and unique radiological features of COVID-19 infection, making the diagnosis even more challenging.

## 2.2 Technical Background: The Basics of Deep Learning for Image Classification

This section provides an overview of the technical aspects of the deep learning approach developed in this thesis. First, a concise history of deep learning, emphasizing key contributions, will offer context for contemporary deep learning strategies. Next, the foundation of the deep learning algorithm for classification tasks will be outlined, followed by a brief technical review of the neural network architectures, hyperparameters, and training strategies employed in this work.

First, it may be helpful to define some commonly used terms. Broadly, artificial intelligence (AI) describes when a machine mimics cognitive functions that humans associate with other human minds, such as learning and problem-solving. Machine learning (ML), a subset of AI, is a series of algorithms that analyze data, learn from it, and make informed decisions based on those learned insights. Finally, deep learning (DL) is a subfield of machine learning based on artificial neural networks. Artificial neural networks have unique capabilities that enable deep learning models to solve tasks that machine learning models could never solve. The terms "model" and "network" will also be used interchangeably in this work and describe the trained deep learning algorithm.

#### 2.2.1 A Brief History of Deep Learning

Deep learning is a machine learning subfield inspired by the human brain's structure and function, specifically neural networks. It uses these neural connections, or layers of algorithms, to process



**Artificial Intelligence (AI):** Any technique that enables computers to mimic human intelligence. It includes machine learning.



**Machine Learning (ML):** A subset of AI that includes techniques that enable machines to improve at tasks with experience. It includes deep learning.



**Deep Learning (DL):** A subset of ML based on neural networks that permit a machine to train itself to perform a task.

Figure 2.5: The hierarchy of artificial intelligence, machine learning, and deep learning.

data and extract features to develop abstractions that loosely imitate a human's thinking process. The history of deep learning can be traced back to the 1940s with the introduction of the first artificial neural networks when Walter Pitts and Warren McCulloch created a computer model based on the human brain's neural networks.<sup>38</sup> They used a combination of algorithms and mathematics called "threshold logic" to mimic the thought process of a human learner.

During the 1960s, researchers began experimenting with neural network architectures, such as the perceptron, a single-layer neural network. This was made possible due to the backpropagation model developed by Henry J. Kelley and Stuart Dreyfus.<sup>39,40</sup> Unfortunately, backpropagation at this time was inefficient and would not be practically useful for another couple of decades. The first general, working learning algorithm for supervised, deep, feedforward, multilayer perceptrons was published by Alexey Ivakhnenko and Valentin Lapa in 1967.<sup>41</sup> However, these early neural networks were limited in their ability to learn and generalize from data.

The first "AI winter" occurred during the 1970s when the lofty promises of AI were not achieved, resulting in a dramatic slash of funding and research in AI. However, many kept researching throughout this time. At the end of this decade, the first convolutional neural networks were developed by Kunihiko Fukushima. These networks, termed "Neocognitrons,"<sup>42</sup> used a hierarchical, multilayered design with multiple convolutional and pooling layers to learn visual patterns in images. The networks were trained with a reinforcement strategy of recurring activation in multiple layers, and important features could be manually adjusted by increasing the weight of specific



**Figure 2.6:** (Left) Warren Sturgis McCulloch was an American neurophysiologist and cybernetician who, along with Walter Pitts, created computational models based on mathematical algorithms called threshold logic. (Right) A biological neuron compared to an artificial neural network: (a) human neuron; (b) artificial neuron; (c) biological synapse; and (d) artificial neural network synapses. Adapted from *Artificial Neural Networks* by Kenji Suzuki.

connections. Many of the concepts of the Neocognitron are still used in modern neural networks today.

By the late 1980s, backpropagation was gaining traction again. At Bell Labs in 1989, Yann LeCun demonstrated the feasibility of convolutional neural networks in conjunction with backpropagation to read handwritten digits.<sup>43</sup> An impressive feat, but the training required over three days, demonstrating that computational resources were insufficient to implement these algorithms practically. These initial successes once again created overly optimistic expectations of AI and created a backlash against the community, insomuch that "Artificial Intelligence" was regarded as pseudo-science. Nevertheless, another AI winter shortly followed with some notable advances, such as the development of the support vector machine (SVM)<sup>44</sup> in 1995 and the long short-term memory (LSTM)<sup>45</sup> used in recurrent neural networks in 1997.

In the 2000s, advances in computer hardware, such as the development of graphical processing units (GPUs) and the increasing availability of large amounts of data, made it feasible to train deep neural networks (DNNs). This led to breakthroughs in image and speech recognition and the development of deep learning architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

In 2009 ImageNet,<sup>46</sup> a free database of more than 14 million labeled images, was established



**Figure 2.7:** (Left) One of the first convolutional neural networks, the Neocognitron developed by Kunihiko Fukushima, compared to a modern DenseNet convolutional neural network (right).

and marked the beginning of the "big data" era. In the 2010s, GPUs increased computational speeds by orders of magnitude, and with ever-increasing data, deep learning was finally beginning to reach its potential. Many astonishing benchmarks were made in this decade as computers were able to beat humans in a variety of different tasks. Perhaps the quintessential example came in 2017 when a neural network named AlphaGo,<sup>47</sup> leveraging reinforcement learning, beat the current Go world champion Ke Jie.

Deep learning continues to achieve state-of-the-art performance in various tasks, including image and speech recognition, natural language processing, and game playing. Furthermore, with new architectures, such as transformer networks and GPT-3,<sup>48</sup> deep learning has become one of AI's most active and promising research areas. Popular modern deep learning networks number in the millions of trainable parameters, with corporate-funded models reaching trillions of trainable parameters.

Today it is hard to find an industry that has not been dramatically affected by deep learning. From online shopping and fraud detection to stock trading, deep learning has seamlessly been integrated into many facets of our daily lives. In fact, deep learning software was used to help check the spelling and grammar of this thesis! Deep learning has had a meteoric rise, and there does not seem to be signs of it slowing down anytime soon, with Geoffry Hinton stating, "The point about this approach is that it scales beautifully. Basically, you just need to keep making it bigger and faster, and it will get better. There's no looking back now."<sup>49</sup>

While deep learning holds much promise, it does not come without its challenges. With each AI success comes a multitude of failures and ethical dilemmas. As a result, it becomes critically

important to understand why and how it works and, perhaps even more importantly, understand when it does not work.

A couple of interesting, somewhat related examples related to COVID-19 are presented here to demonstrate some of the incredible feats of modern deep learning. The first comes from the field of natural language processing. ChatGPT,<sup>48</sup> developed by OpenAI, has pushed the boundaries of what was thought possible in natural language processing. ChatGPT mimics a human conversationalist with astonishing versatility and improvisation, including writing and debugging computer programs, composing music, teleplays, fairy tales, and can answer test questions (sometimes, depending on the test, at a level above the average human test-taker). For example, when given the prompt "compose a poem about the radiographic presentation of COVID-19," ChatGPT wrote the following poem (in about three seconds!):

In lungs once pink and full of air Now shadows linger, dark and spare A virus unseen, yet all too real leaves its mark, a stark reveal

The X-ray shows the tale it tells of alveoli filled with fluid swells The damage done, a somber sight

A reflection of a global plight

But hope is not yet lost, we see the promise of vaccines setting us free A path to healing, a way to mend and with each shot,

A new hope to tend

So let us not forget the fight against this virus, day and night For though the image may be grim

We will overcome, and heal within

-ChatGPT (2023)

Another example is an image created by the author using a type of deep convolutional neural network known as a style-transfer cycleGAN.<sup>50</sup> This network "learns" different art styles and translates them to an input image, in this case, a CXR image.

Hopefully, these couple of examples can help demonstrate the incredible quantum leap of deep


**Figure 2.8:** An image created by the author using a cycleGAN where a chest radiograph is reimagined in different modern art styles.

learning, from simple digit recognition to nearly indistinguishable human-like creativity, all within the span of just 30 years!

# 2.2.2 Supervised Learning

Distinguishing COVID-19 infection from other causes of lung abnormalities can be approached as a supervised learning binary classification task. Consequently, implementing AI, especially deep

learning, becomes a natural choice, as these methods have demonstrated exceptional performance in image classification tasks.<sup>51–53</sup> Additionally, deep learning offers an advantage in terms of learning speed compared to human learners. The training process for deep learning models is significantly shorter, allowing for the development of high-performing models in a fraction of the time it takes for a human to learn to perform the same task.

Supervised learning is a type of machine learning where the model is trained on labeled data. In supervised learning, the training dataset consists of input-output pairs, where the model is provided with inputs (also known as features or predictors) and their corresponding outputs (also known as labels or targets). The model learns to map the inputs to the correct outputs during training, and this learned mapping is then used to make predictions on unseen data.

There are two main types of supervised learning: classification and regression. In classification, the output is a discrete label, and the goal is to predict the class label of new inputs. For example, a model is trained based on millions of emails on different parameters, and whenever it receives a new email, it identifies whether the email is spam. In regression, the output is a continuous value, and the goal is to predict the value of a continuous output variable given the input features. For example, in weather prediction, the model is trained on historical data, and once the training is completed, it is used to predict the temperature for future days.

Supervised learning is termed "supervised" because, during the training process, the model is constrained by labeled examples that provide the correct output for a given input. The model is trained to learn a mapping between inputs and outputs and generalize this mapping to new, unseen data. The supervision comes in providing the model with the correct output for each input during the training process. The idea behind supervised learning is that the model should be able to predict the output for new unseen data based on the patterns it has learned from the labeled training data. This is in contrast to unsupervised learning, where the model is not given any labeled examples and must discover the underlying structure of the data on its own.

Supervised learning is prevalent in deep learning research because it has been around for decades and is a well-established and well-understood technique. In addition, it is a more straightforward and intuitive concept (say, compared to reinforcement learning) that is easy to understand and implement. Supervised learning is also well-suited to many real-world problems and is easy to evaluate using metrics such as accuracy, precision, and recall, which makes it easy to compare



**Figure 2.9:** Classification is the process of finding or discovering a model or function which helps separate the data into multiple categorical classes, i.e., discrete values. Regression is the process of finding a model or function for distinguishing the data into continuous real values instead of using classes or discrete values.

different models and select the best one.

However, there are some notable limitations of supervised learning, many of which will be the focus of the work in this thesis. Supervised learning, generally, can only predict outcomes based on the patterns it has seen during training, and so it is limited by the quality and quantity of the labeled training data. If the data is biased, the model will learn the bias and replicate it in its predictions; this is a problem known as "bias amplification."<sup>54</sup> In many applications, the availability of labeled data can be a limiting factor in developing a deep learning model. A trained model can be prone to overfitting if the training data is too small or similar, leading to poor generalization performance on new unseen data.

## 2.2.3 How Deep Neural Networks Learn

The task of the deep learning strategy in this thesis can be described simply: given a chest X-ray, does the patient have COVID-19 or not? While this is a highly non-trivial task to actualize in reality, it boils down to a simple "yes" or "no" classification. This is known as a binary classification problem, and many algorithms have been developed to perform this task (and its extension, multiclass classification). An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function implemented by a classification algorithm that maps input data to a category. Classification is a subset of the more general problem of pattern recognition, which is assigning some sort of output value to a given input value.



**Figure 2.10:** A neural network is composed of an input layer, hidden layers, and an output layer. Each layer consists of several neurons which perform simple mathematical operations. The layers are connected through weighted edges, representing the strength of the connections between neurons. During the learning process, input data is passed through the network in a forward direction from the input layer to the output layer. Each neuron in the network calculates a weighted sum of its inputs, applies an activation function, and passes the result to the next layer.

Given data, a DNN can learn specific patterns or attributes from it. But how does it "learn" exactly? This section provides a (very) brief overview of the critical components of the learning process of neural networks, with more details in the following subsections.

A DNN is composed of an input layer, multiple hidden layers, and an output layer. Each layer consists of several artificial neurons, also known as nodes or units, which perform simple mathematical operations. The layers are connected through weighted edges, representing the strength of the connections between neurons. During the learning process, input data is passed through the network in a forward direction from the input layer to the output layer. Each neuron in the network calculates a weighted sum of its inputs, applies an activation function (e.g., ReLU, sigmoid, or tanh), and passes the result to the next layer. Finally, the output layer generates a prediction based on the processed input data. A simple example is shown in Figure 2.10.

The primary purpose of an activation function in artificial neural networks is to introduce non-linearity into the model. This non-linearity allows the network to learn and approximate complex, non-linear relationships between inputs and outputs, making it capable of handling various tasks. They also act as a thresholding mechanism determining whether a neuron should be activated. By applying the activation function to the weighted sum of inputs and bias, a neuron



**Figure 2.11:** Activation functions in artificial neural networks introduce non-linearity into the models and determine whether a neuron should be activated. Common activation functions include sigmoid, hyperbolic tangent, and ReLU.

decides whether to pass the signal forward or suppress it. This enables the network to focus on relevant features and filter out noise. Common activation functions are compared in Figure 2.11.

A loss function is used to quantify the difference between the predictions made by the DNN and the actual target values. Standard loss functions include mean squared error for regression tasks and cross-entropy loss for classification tasks. The learning process involves adjusting the neural network weights to minimize the loss function. This is achieved through an algorithm called backpropagation, which computes the gradients of the loss function with respect to each weight by applying the chain rule of calculus. The backpropagation algorithm starts from the output layer and moves backward through the network (thus the name "back"-propagation), calculating the error gradients for each neuron and weight. Once the gradients are calculated, an optimization algorithm, such as gradient descent or its variants (e.g., Adam, RMSprop), is used to update the weights in the network. The optimization algorithm adjusts the weights in a direction that minimizes the loss function using the calculated gradients.

The learning process consists of multiple iterations, or epochs, where the entire dataset is passed through the network multiple times. With each iteration, the weights are updated, and the model becomes better at making predictions. In addition, various regularization techniques can be employed to prevent overfitting and improve the DNN's ability to generalize to unseen data, such as L1 or L2 regularization, dropout, and early stopping. These techniques help the model learn more robust features and prevent it from relying too heavily on specific training examples.

## 2.2.4 Convolutional Neural Network Architecture

Convolutional neural networks are a specialized type of deep learning architecture primarily designed for tasks involving grid-like data, such as images, where spatial (or temporal) relationships are crucial. This type of architecture performs a better fitting to image data due to the reduction in the number of parameters involved and the reusability of weights. For example, a standard RGB image 1280×720 pixels contains almost 3 million inputs! However, the role of a CNN is to reduce the images into a form that is easier to process without losing features critical for getting a good prediction and is also scalable to massive datasets.

Convolutional layers are the core building blocks of CNNs. They apply a set of learnable filters (kernels) to the input, capturing local patterns and spatial features, such as edges, corners, and textures. The filters slide over the input (convolve), computing element-wise multiplications (Hadamard product) and aggregating the results into a feature map. The number and size of filters and the stride and padding can be adjusted to control the output dimensions and receptive field. A simple example of this process is shown in Figure 2.12 for a  $5 \times 5 \times 1$  image convolved with a  $3 \times 3 \times 1$  kernel with a stride of 2.

The convolution operation aims to extract high-level features, such as edges, from the input image. CNNs usually consist of many convolutional layers, and the initial convolutional layers capture low-level features, including edges, color, and gradient orientation.<sup>55</sup> As more layers are added, the architecture becomes capable of recognizing high-level features, resulting in a network that comprehensively understands images in the dataset, much like a human would.

After the convolution operation, an activation function, such as Rectified Linear Unit (ReLU),<sup>56</sup> is applied element-wise to the feature maps. This introduces non-linearity into the model, enabling it to learn complex patterns and representations.

Another key feature of CNNs is pooling layers. Pooling layers are used to downsample the feature maps, reducing their spatial dimensions and computational complexity while retaining important information. Furthermore, they are useful for extracting dominant features, which are



**Figure 2.12:** A  $5 \times 5 \times 1$  image convolved with a  $3 \times 3 \times 1$  kernel with a stride of 2. The kernel slides across the image (convolution), and an element-wise multiplication operation (known as the Hadamard Product) is taken between the kernel and the portion of the image over which the kernel is hovering.

rotational and positional invariant. Common pooling operations include max pooling, average pooling, and global average pooling. Pooling layers often follow convolutional layers, together forming the "i-th" layer of a CNN. There are many ways to use convolutional layers, and different architectures combine them for different purposes.

For classification tasks, after many alternating convolutional and pooling layers, the final feature maps are flattened and connected to one or more fully connected (dense) layers. These layers are responsible for integrating the high-level features learned by the previous layers and making predictions or decisions based on the entire input. In a classification task, the final fully connected layer has as many output units as there are classes, and an activation function is applied to generate class probabilities. The output layer produces the final predictions or decisions based on the learned features and is typically a probability distribution across the classes, and the class with the highest probability is selected as the prediction.

AlexNet,<sup>57</sup> a deep CNN, won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a prestigious computer vision competition, by achieving a top-5 error rate of 15.3%, significantly outperforming the second-best entry, which had an error rate of 26.2%. Relatively



**Figure 2.13:** Relatively simple compared to modern CNNs, AlexNet highlights the main components of a CNN: alternating convolutional and pooling layers followed by fully connected layers and an output layer. Image from Krizhevsky et al., "ImageNet classification with deep convolutional neural networks." *NIPS*, 2012.

simple compared to modern CNNs, AlexNet architecture (see Figure 2.13) highlights the main components of a CNN: alternating convolutional and pooling layers followed by fully connected layers and an output layer.

## 2.2.5 Training and Optimization

In image classification tasks, an input image is fed to the CNN, where the convolutional layers extract information (features) about the image, which are used to predict the image class. The CNN must "learn" which features are important for each class from the training data. The process in which CNNs "learn" is called training and involves two fundamental algorithms: backpropagation and gradient descent. They work together to adjust the model's parameters to minimize the loss function, allowing the model to learn patterns in the data. A loss function quantifies the difference between the predictions made by the network and the true target values.

## 2.2.5.1 Loss Function

Each example  $(x_i, y_i)$  in binary classification belongs to one of two complementary classes—for example, a CXR image x with a positive PCR test y = 1. Say a particular neural network is trained to solve the problem of classifying a CXR image as either COVID-19 positive ("1") or COVID-19 negative ("0"). Given an input image x, this network seeks to give the probability that the patient is COVID-19 positive, P(y = 1|x) = p (the negative class can be derived as P(y = 0|x) = 1 - P(y = 1|x) = 1 - p). A good classifier should produce a high value p when the input x has a positive label y = 1 and, conversely, a low value p for a negative label y = 0. In other words, the goal is to maximize p when y = 1 and maximize 1 - p when y = 0. This can be expressed as  $P(y|x) = p^y(1-p)^{1-y}$ , where y = 1 reduces to p and y = 0 reduces to 1 - p. This is the probability mass function (PMF) of the Bernoulli distribution, named after Swiss mathematician Jacob Bernoulli, and calculating it for a single data point is called a Bernoulli trial. The goal is to maximize the Bernoulli distribution for every trial (or training samples  $x_i$ ). One way to accomplish this is by using maximum likelihood estimation.

In statistics, maximum likelihood estimation (MLE) is a method of estimating the parameters of an assumed probability distribution given some observed data. This is achieved by maximizing a likelihood function so that the observed data is most probable under the assumed statistical model. It is a function that measures a statistical model's "goodness of fit" to the observed data, given a set of parameter values. For example, given a dataset with N samples, for each sample  $x_i$ , the target (true) values are represented by  $y_i$  (0 or 1), and the predicted probabilities for the positive class are  $p_i$ . The likelihood function can be defined as the product of the probabilities of the true class labels for each sample:

$$\mathcal{L}(p) = \prod_{i=1}^{N} \left[ p_i^{y_i} (1 - p_i)^{(1 - y_i)} \right],$$
(2.1)

where the goal is to maximize each probability  $p_i$ , given the samples  $(x_i, y_i)$ . The likelihood in its current form is somewhat cumbersome to work with, but taking the logarithm can help simplify calculations:

$$\log \left(\mathcal{L}(p)\right) = \sum_{i=1}^{N} \left[y_i \log \left(p_i\right) + (1 - y_i) \log \left(1 - p_i\right)\right].$$
(2.2)

Note that maximizing a function is the same as minimizing the negative of the function, and averaging over all *N* samples, a loss function can be derived to guide a neural network's learning:

Binary Cross-Entropy Loss = 
$$-\frac{1}{N} \sum_{i=1}^{N} [y_i \log (p_i) + (1 - y_i) \log (1 - p_i)].$$
 (2.3)

This is known as the binary cross-entropy loss function and is one of the most common loss functions used for model training. Cross-entropy refers to the measure of the dissimilarity between two probability distributions. In the context of classification, it compares the predicted probability distribution (generated by the model) with the true probability distribution (represented by the target variable). The cross-entropy loss is derived from the concept of information theory, where it is used to quantify the average number of bits required to encode events from one distribution using the optimal code for another distribution.

#### 2.2.5.2 Backpropagation

Once a loss is quantified by way of the loss function, backpropagation and gradient descent can be used to update the learnable parameters of the network such that better predictions can be learned. Backpropagation<sup>39,40</sup> is a supervised learning algorithm used to compute the gradients of the loss function with respect to each model parameter (weights and biases). It is essentially an efficient implementation of the chain rule from calculus applied to the computation graph of the neural network. Backpropagation proceeds in two passes:

- 1. **Forward Pass**: Input data is passed through the network to compute predictions. Intermediate outputs and activations are stored for use in the backward pass.
- 2. Backward Pass: The error between the predictions and true target values is calculated using the loss function. The gradients of the loss function with respect to each parameter are computed, starting from the output layer and moving backward through the network to the input layer. The chain rule calculates each layer's gradients based on the subsequent layer's gradients.

The gradient, or partial derivative of a function with respect to a variable, measures the sensitivity to the change of the function value with respect to a change in the argument variable. This information is used to adjust the weights and biases in the negative gradient direction to minimize the loss, effectively reducing the error in the network's predictions. Gradients can be calculated following the chain rule; for a single weight  $w_{jk}^{(l)}$ , where w is the weight from the k-th neuron in the (1 - l)-th layer to the j-th neuron in the l-th layer, and cost function C(w, b):

$$\frac{\partial C(w,b)}{\partial w_{jk}^{(l)}} = \frac{\partial C(w,b)}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial w_{jk}^{(l)}},$$
(2.4)

(1)

where z is the weighted sum of its n inputs:

$$z_j^{(l)} = \sum_{k=1}^n w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)},$$
(2.5)

and *a* is the specific activation function used. Using the definition of *z*, the partial derivative of  $z_j^{(l)j}$  with respect to  $w_{jk}^{(l)}$  can be expressed as

$$\frac{z_j^{(l)}}{w_{jk}^{(l)}} = a_k^{(l-1)},\tag{2.6}$$

and thus the gradient of the cost function C with respect to weight w can be written as

$$\frac{\partial C(w,b)}{\partial w_{jk}^{(l)}} = \frac{\partial C(w,b)}{\partial z_j^{(l)} a_k^{(l-1)}}.$$
(2.7)

A similar process can be followed to find the gradient of *C* with respect to the bias *b*:

$$\frac{\partial C(w,b)}{b_j^{(l)}} = \frac{\partial C(w,b)}{z_j^{(l)}}.$$
(2.8)

Figure 2.14 gives an example of this process. While backpropagation is straightforward in theory, many technical limitations exist, such as memory consumption, different tensor types and precision, and undefined gradients. Accordingly, much technical effort has been made to implement this algorithm effectively and efficiently.

## 2.2.5.3 Gradient Descent

Gradient descent is an optimization algorithm commonly used in DNN training that adjusts the network's parameters iteratively to minimize the loss function.<sup>58,59</sup> It uses the gradients computed by backpropagation to update the weights and biases in the direction that reduces the loss or the difference between the network prediction and the label. The idea is to take repeated steps in the opposite direction of the gradient (or approximate gradient) of the function at the current point, because this is the direction of steepest descent. Gradient descent is used in DNN training because most deep learning optimization problems are highly non-convex and impossible to solve



**Figure 2.14:** An example of the backpropagation algorithm, which computes the gradient of the cost function *C* to certain parameters. This information is used to adjust the weights and biases in the direction of the negative gradient to minimize the loss, effectively reducing the error in the network's predictions.

in a closed form; therefore, they must be approximately solved using a numerical optimization procedure.

To understand what the algorithm is doing, imagine a person stranded high on a mountain who wants to descend to their house at the base of the mountain (i.e., locate the global minimum). Due to heavy fog, visibility is severely limited, rendering the trail invisible. As a result, the individual must rely on local information to identify the minimum. They can employ the gradient descent method, which entails assessing the slope of the ground at their current location and then moving in the direction of the steepest descent (i.e., downhill). Using this method, they would eventually descend the mountain or get stuck in a small ravine (local minimum). However, assume that the hill's steepness is not immediately apparent with simple observation, but rather it requires using a level (which this person happens to have on them). Since using the level to measure the ground is tedious, this person wants to minimize their use of the level, especially if they want to get down before nightfall. The difficulty is choosing the frequency at which they should measure the hill's steepness so as not to go off track.

In this analogy, the person symbolizes the algorithm, and the path they take down the mountain represents the series of parameter settings the algorithm will investigate. The slope of the hill at a specific location corresponds to the function's slope at that point. Differentiation serves as the



**Figure 2.15:** Gradient descent is a first-order iterative optimization algorithm used to minimize an objective function, typically used for finding the minimum of a convex function or a local minimum of a non-convex function. Here the red line represents the path of parameter values the algorithm uses to minimize the loss function.

tool for measuring steepness. The chosen direction aligns with the function's gradient at that point. The time between measurements reflects the step size, and the challenge lies in determining the optimal frequency for measuring the hill's steepness to stay on track without excessive use of the instrument. Figure 2.15 illustrates gradient descent, where the x and y axes represent different parameter values, and the z axis is the associated loss value for the x, y parameters. The red line is the series of parameter values the algorithm chooses on its path to find a minimum loss value. Notice that the minimum achieved is not always the global minimum.

Mathematically, gradient descent can be summarized as follows

Repeat until convergence {  

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} C(w)$$
 (2.9)  
}

The learning rate  $\alpha$  is a hyperparameter that controls the step size of the updates. A smaller learning rate may result in a more accurate solution, but it may take longer to converge. Conversely, a larger learning rate can speed up convergence but may overshoot the minimum and cause oscillations or divergence.

Batch gradient descent uses the entire dataset to compute the gradients and update the parameters in a single step; however, this approach can be computationally expensive for large datasets. On the other hand, stochastic gradient descent (SGD) uses the gradients computed for a single training example to update parameters. This introduces randomness and can help the model escape local minima, but it may cause more fluctuations and instability in the learning process. A compromise between batch and stochastic gradient descent, mini-batch gradient descent, computes gradients and updates parameters based on a small subset (mini-batch) of the training data. This balances the computational efficiency and stability of the learning process and is the most commonly used method in modern deep-learning training.

Several issues with gradient descent optimization in modern deep learning include vanishing and exploding gradients, local minima, and saddle points. Several improvements and variants of gradient descent have been proposed to enhance the training process, such as momentum,<sup>60</sup> Nesterov accelerated gradient,<sup>61</sup> AdaGrad,<sup>62</sup> RMSprop,<sup>63</sup> and Adam.<sup>64</sup> These methods often adapt the learning rate, dampen oscillations, or accelerate convergence.

## 2.2.6 Hyperparameters

Hyperparameters are adjustable parameters that govern the learning process and model architecture in DNNs. They are generally set before the training process and can play a crucial role in the model's overall performance. Hyperparameters are used by the DNN when it is learning, but they are not part of the resulting model. There are many different hyperparameters, and some play a bigger role depending on the type of network or the task. For the sake of completeness, some important hyperparameters for image classification are briefly discussed:

- Network Depth: The number of layers in a CNN, including convolutional, pooling, and fully connected layers. Deeper networks can capture more complex features but are computationally expensive and prone to overfitting.
- **Number of Filters**: The number of filters (kernels) in a convolutional layer. More filters allow the model to learn a larger variety of features, but it increases the computational complexity.
- Filter Size: The spatial dimensions (height and width) of the filters in a convolutional layer.

Smaller filters capture fine-grained local features, while larger filters capture more global features.

- **Stride**: The step size taken by filters when sliding across the input image during the convolution operation. Larger strides result in smaller output feature maps and faster computation but may reduce the model's ability to capture fine-grained features.
- **Pooling**: The pooling operation (max pooling, average pooling, or global average pooling) and its window size used in pooling layers. Pooling layers help reduce spatial dimensions and improve translational invariance.
- Activation Functions: The non-linear activation functions used in the network, such as ReLU, sigmoid, or tanh. Activation functions introduce non-linearity, enabling the model to learn complex patterns.
- Learning Rate: A crucial parameter in gradient-based optimization algorithms that determines the step size of parameter updates. The learning rate should be set appropriately to ensure convergence and stability during training.
- **Batch Size**: The number of training examples used in each mini-batch gradient descent update step. Smaller batch sizes lead to faster training but may introduce more fluctuations, while larger batch sizes provide more stable updates but are computationally expensive.
- **Number of Epochs**: The number of times the entire training dataset is passed through the network during training. Too few epochs may result in underfitting, while too many epochs can lead to overfitting.
- **Regularization**: Techniques such as L1, L2, or dropout are used to prevent overfitting and improve the model's generalization capabilities.
- Weight Initialization: The method used to initialize the model's weights, such as Glorot<sup>65</sup> or He<sup>51</sup> initialization, which can affect the training convergence.

Choosing the appropriate values for these hyperparameters is essential for achieving good performance and generalization. The community has adopted standards for most hyperparameters;

however, grid search, random search, or more advanced techniques such as Bayesian optimization can be used to find the optimal combination of hyperparameters for a specific task and dataset.

## 2.2.7 Limitations of Deep Learning

With a renewed interest in AI, the surge of deep learning spurned by the ImageNet challenge, the development of powerful computing hardware, and the advent of new novel deep neural networks, it's hard to find an application where deep learning has yet to be applied. Specifically, in the medical imaging domain, deep learning has revolutionized medical imaging, <sup>66–68</sup> particularly in detection, <sup>69,70</sup> classification, <sup>70,71</sup> segmentation, <sup>70,72</sup> and reconstruction. <sup>73–76</sup> It is apparent deep learning has taken the community by storm. However, deep learning is not without its flaws, and it is of paramount importance that these flaws are recognized, studied, and understood so that they can be sufficiently addressed if deep learning is to be truly beneficial in a clinical setting.

Deep learning is very dependent on data, with models typically requiring large amounts of labeled data for training. Unfortunately, gathering and labeling sufficient data for DNN training can be time-consuming, expensive, and sometimes infeasible, especially for rare events or small sample sizes. Chapter 3 discusses this in more detail. In addition to being data-intensive, deep learning models can be computationally expensive and resource-intensive, requiring specialized hardware like GPUs or TPUs. This may limit the accessibility and scalability of deep learning solutions.

DNNs work almost like magic, input data is fed into the network, and seemingly impossible results are obtained. This "magic" is, in fact, one of the main concerns of modern deep learning and is why DNNs are often called "black boxes": input data is placed into the box (network), and results are obtained, but the transformation mechanism is poorly understood. The problem lies in the immense complexity of the networks. Neural networks work by approximating functions: they learn an approximate mapping from input to desired output. However, even the simplest DNNs contain millions of parameters, making them essentially intractable to human comprehension. While the general mechanisms and processes by which DNNs are constructed have been comprehensively studied and are relatively well understood, there remains no simple link between a network's structure and the function it approximates. Deep neural networks are thus very difficult to interpret and, accordingly, very difficult to trust. This can lead to unexpected and unpredictable behavior from neural networks, so a DNN must be thoroughly and comprehensively studied and tested.



**Figure 2.16:** Feature visualization answers questions about what a network, or parts of a network, are looking for by generating images. Above are some examples of features learned by GoogLeNet trained on the ImageNet dataset. This technique can help understand what the DNN learned, but often the features are abstract and not easily interpreted. Image from Olah et al., "Feature visualization." *Distill*, 2017.

This "black box problem" is an active area of research, and many methods have been developed to aid in "glimpsing into the black box." Most of these methods are post hoc, meaning they try to make sense of a network by examining its output and parameter values. Techniques such as class activation mapping<sup>77</sup> or saliency mapping<sup>78</sup> generate heatmaps to highlight important features used by the network for its task. These heatmaps allow a visual representation of what parts of an image are most important, helpful for identifying confounding features and highlighting common and unique features of a specific class. While these methods are useful, they still do not fully address the black-box nature of neural networks.<sup>79</sup> Model interpretability is further explored in Chapter 6.

A significant challenge limiting the real-world implementation of neural networks in deep learning is that a model that performs well on training data may not perform equally well on unseen test or external data. This decrease in performance when transitioning from training data to test data is called the generalization gap, which is currently an active area of research within the AI community. The generalization gap is a multifaceted issue, as numerous factors contribute to its complexity. A prevalent cause is overfitting, which occurs when the model learns the noise present in the training data instead of the underlying patterns. Consequently, the model's ability to generalize to unseen data is negatively impacted.

Deep learning models learn from the data they are trained on. If the training data is biased or unrepresentative of the real-world distribution, the model will not generalize well to new, unseen data. This is particularly problematic when the training data is collected from a limited context or contains inherent biases. These can create shortcuts where a network bypasses relevant features



(a) Husky classified as wolf







(A) **Cow: 0.99**, Pasture: (B) No Person: 0.99, Water:

**Figure 2.17:** Deep learning models are prone to learning the most discriminating features, even if these features are not useful for the overall task. Shortcut learning occurs when spurious correlations exist between the image features irrelevant to the task and the corresponding training labels. For example, snowy and grassy background settings became shortcuts for identifying wolves (left) and cows (right) instead of useful discriminating features. Because the networks learned shortcuts, their generalizability to unseen data is severely limited. Images from Ribeiro et al., "Why should I trust you?" *SIGKDD*, 2016 and Beery et al., "Recognition in terra incognita." *ECCV*, 2018.

and focuses on correlations. Recently, the concept of shortcut learning has drawn the attention of the deep learning community.<sup>80</sup> It has been found that poor generalizability may be attributed to when the training dataset has hidden shortcuts, i.e., spurious correlations between the image features that are irrelevant to the task and the corresponding training labels. When these spurious correlations exist, models quickly pick these spurious correlations over the desired image features to establish the connections between input image data and output labels. For example, a neural network was trained to differentiate dogs from wolves with high accuracy.<sup>81</sup> However, when testing data consisting of husky dogs were applied, the performance dropped dramatically. The answer to this unexpected drop can be explained by the fact that the network relied on a shortcut when making a prediction: the presence of a snowy background. It was found that most images of wolves contained snow, and similarly, most pictures of husky dogs contained snow. Similarly, a network detected and classified cows correctly in "common" contexts (e.g., alpine pastures), while cows in uncommon contexts (e.g., beach) were not detected.<sup>82</sup> Example figures from these authors are shown in Figure 2.17.

Shortcut learning severely hinders the translation and deployment of deep learning models for high-stakes tasks such as those in healthcare applications.<sup>83</sup> This was made apparent at the onset of the pandemic, where despite thousands of COVID-19 prediction machine learning models being developed, very few performed well on real-world clinical tests.<sup>84–86</sup> For example, early studies have shown that deep learning models can learn to differentiate chest X-rays from different

hospitals and patient groups.<sup>83</sup> These findings indicated that different data sources and patients' population characteristics, such as gender, age, and race, could also become shortcuts. DeGrave et al.<sup>87</sup> also showed that a model only learned the source label as a shortcut for prediction when COVID-19 positive and negative training data were collected from two different sources. As a result, the trained model does not have the desired prediction power in real-world clinical scenarios. The authors also showed that the trained model used extrinsic image features such as lead markers for prediction. Yet, these markers were only introduced to label the orientation of patients in X-ray image acquisitions and did not correspond to any disease features. If shortcuts can be identified, a challenge in its own right, efforts to remove them can be difficult and may not actually remove shortcut learning.<sup>88</sup>

Shortcut learning and, therefore, poor generalizability can often be attributed to the quality of the training data. Spurious confounding factors, i.e., shortcuts, may exist in the training data without proper data collection and curation strategies. This problem is compounded by the "big data" mantra that "more data is better." It is often assumed that increasing the data size and collecting data from diversified sources, i.e., multiple institutions, will ensure generalizable models.<sup>89</sup> While it is well known that modern DNNs can improve performance with increasing data, adding data can further introduce shortcut learning. This is extensively studied in Chapters 4 and 5.

Despite these drawbacks, deep learning has proven to be a powerful tool for image classification when proper precautions are taken. In the setting of a pandemic, DNNs can quickly learn the unique features of a disease and aid in screening for patients suspected of infection to help prevent the spread of the disease. Even with the development and administration of COVID-19 vaccines, COVID-19 is likely to remain an important differential diagnosis for the foreseeable future for any person presenting flu-like symptoms for which CXR is a common prescription.

# 2.3 Major Research Problems of this Thesis Work

In the past decade, deep learning has shown unprecedented performance in object detection, image classification, and natural language processing thanks to its powerful capability to learn complex and high-level features from data. While the performance of deep learning models on various benchmark test data sets continues to show improvement, limitations and practical value of the

models in real-world scenarios are also being revealed and have received broad attention in research and application communities.

The primary research problem addressed in this thesis is developing a generalizable deeplearning strategy for the classification of COVID-19 in chest X-ray radiography. This work will address the entire pipeline of a deep-learning strategy from data curation, model training, evaluation, model interpretation, and performance monitoring. The main issue of model generalizability limiting the clinical impact of deep learning algorithms will be studied by detecting and identifying shortcut learning and dataset size and curation. This thesis aims to provide a framework for developing and optimizing generalizable AI solutions for image classification in medical imaging.

# Chapter 3

# Exploring and Evaluating Chest X-ray Datasets: An Analysis of Dataset Variability and Distinctiveness

Data is a set of values, symbols, or representations of facts, concepts, or instructions that are stored, processed, and communicated by computers and other digital devices. Data can take many forms, including numbers, text, images, audio, and video. The term "data" is often used to refer to information organized in a specific format to facilitate processing, analysis, and communication. It is the foundation of modern computing and is essential for developing applications, algorithms, and systems that support various activities, from communication and entertainment to scientific research and business operations. Data is also critical for developing artificial intelligence and machine learning, as these technologies rely on large amounts of data to train algorithms and make predictions. As discussed in the previous chapter, much of the success of deep learning has come as a consequence of the internet and the sharing and curation of large datasets.

To comprehend the scale of the massive increase in "data," consider the number of photographs you have of yourself or your family members as children. Then, compare that to the number of photographs you have of your child, grandchild, pet, etc. For most, in this context, it is easy to appreciate the staggering growth of data; for example, modern memory cards have increased from megabytes just a decade ago to terabytes being commonplace now. We live in an era where data is abundant, and the volume of data is rapidly increasing. According to recent estimates, approximately 2.5 quintillion ( $2.5 \times 10^{18}$ ) bytes of data are generated every day, and the total amount of data in the world is estimated to be around 50 zettabytes (which is 40 times more bytes



**Figure 3.1:** In September 1956, IBM launched the 305 RAMAC with a disk storage unit weighing over 2,000 lbs and storing approximately 5 megabytes of data. Today, micro-SD cards over a terabyte are commonly used in electronics, over 200,000 times more storage and orders of magnitude faster than the RAMAC. Image credit: Micron.

than the number of stars in the observable universe).<sup>90</sup> The development of the internet, allowing widespread sharing of data, has ushered in the era of "big data."

If we consider neural networks as cars, data can be considered the gasoline that fuels their performance. Just as one would not fill a high-performance sports car like a Lamborghini with low-octane fuel, poor-quality training (and test) data can cause an AI solution to fail. This is where the adage "garbage in, garbage out" comes into play. While it is well-known in deep learning that larger datasets generally result in better performance,<sup>91,92</sup> the size of the dataset alone does not ensure good results. The impact of data quality and dataset size is explored further in Chapters 4 and 5.

This chapter provides a concise overview of the metadata and datasets utilized throughout this thesis work to investigate the efficacy of deep learning in classifying COVID-19 in chest X-ray images. These datasets were utilized during various stages of the model development and evaluation process. The datasets can be categorized into two groups: non-COVID-19 datasets,

curated before the pandemic and used for model pretraining or bias evaluation, and COVID-19 CXR datasets, consisting of chest X-ray images acquired during the pandemic, containing both positive and negative cases.

# 3.1 Dataset Metadata

This section briefly summarizes the dataset metadata used in this work. DICOM (Digital Imaging and Communications in Medicine) is a widely used standard for managing medical images and healthcare-associated information. DICOM metadata refers to the information contained in a DICOM file that describes the characteristics of the image, such as the imaging modality (e.g., CT, MRI), the patient information (e.g., name, age, sex), and the image acquisition parameters (e.g., pixel size, image orientation).

DICOMs are much more than images, and the metadata can be accessed and manipulated using various software tools, such as DICOM viewers and editors. In addition, the metadata can be used to facilitate the exchange of medical images and related information between different healthcare institutions and systems. However, there are several issues with DICOM to consider, especially in the context of curating datasets to train neural networks. First, DICOM metadata can contain sensitive patient information, such as name, date of birth, and medical history, and therefore needs to be de-identified before it can be publicly released.<sup>93,94</sup> A second primary concern with DICOM metadata is the lack of standardization for certain types of metadata. For example, there are some types of metadata, such as image quality metrics and radiation dose information, for which there is currently no standardized way of encoding and storing the data in DICOM files and thus are dependent on the institution from with which the data is acquired.

This work uses a few key DICOM metadata tags, which can be grouped into patient-specific tags and image acquisition tags.

## 3.1.1 Patient-Specific DICOM Tags

Patient-specific tags are keys that contain information about the patient. The main tags used in this work are Patient ID, Study ID, Sex, Age, Race, and Delta. Due to the Health Insurance Portability and Accountability Act (HIPAA), certain identifying patient information cannot be freely shared.

Patient ID acts as a replacement for the patient name to identify the corresponding patient image. Note that there may, and often are, many images associated with one patient. This is important to consider to prevent data leakage where patients' images could be distributed between training and test data (see Appendix A). Another tag, for example, Study ID, can indicate multiple images corresponding to a patient with Patient ID. Somewhat self-explanatory, Sex, Age, and Race define the patient's sex, age, and race. These characteristics are important to consider to study and address bias in curated datasets and are elaborated more in Chapter 4.

It is widely recognized that some COVID-19 patients may display only mild or no symptoms at all and may not show any signs of lung infection on a CXR.<sup>95</sup> To ensure that the imaging data aligns with the RT-PCR test results, CXRs can be filtered based on the time between testing and imaging. Cases with a significant gap between testing and imaging can be excluded from model training and performance evaluation, as the disease information may not be conclusive.

For instance, if a patient tested positive for COVID-19 but was imaged 50 days after testing, it is highly probable that the patient has recovered by then, and the CXR may not exhibit the relevant radiographic features of COVID-19 infection. In this work, we use Delta ( $\Delta$ ) as a specific tag to quantify the time (in days) from a confirmed PCR test to patient imaging.

#### 3.1.2 Image Acquisition DICOM Tags

Image acquisition tags contain information about the imaging system and parameters of a CXR acquisition. The main tags used in this work are Modality, View Position, Vendor, and Location.

In chest radiography, digital radiography (DX) and computed radiography (CR) are two distinct imaging modalities used to acquire and process radiographic images.<sup>96–98</sup> DX uses a digital detector to directly capture the X-ray image, which is then processed and stored digitally. This technique is faster than traditional film-based radiography and produces high-quality images with a wide dynamic range. Conversely, CR uses a cassette containing a photostimulable phosphor plate to capture the X-ray image. The cassette is then inserted into a CR reader, which uses a laser to scan the plate, which is recorded to produce a digital image. While cheaper and more widely adopted than DX, CR is less efficient and can result in lower image quality due to its indirect imaging process.

In this work, frontal CXRs are used. The anterior-posterior (AP) and posterior-anterior (PA) view positions are two different chest X-ray imaging views.<sup>99</sup> In the AP view, the patient stands



**Figure 3.2:** (Left) Comparison of anterior-posterior (AP) and posterior-anterior (PA) view positions. Due to more geometric magnification, AP imaging results in an image where the heart and other mediastinal structures appear larger than they actually are. Image from https://www.radiologymasterclass.co.uk/gallery/chest/quality/chest-x-ray-ap. (Right) Comparison of digital radiography (DX) and computed radiography (CR). DX uses a digital detector to directly capture the X-ray image, while CR uses a photostimulable phosphor plate. Image from https://www.thalesgroup.com.

facing the X-ray machine, and the X-rays travel from the front of the patient towards the back, penetrating through the chest and onto the detector on the opposite side of the patient. In the AP view, the beam enters the patient's anterior (front) chest wall and exits through the posterior (back) chest wall, resulting in an image where the heart and other mediastinal structures appear larger than they actually are. In the PA view, the patient stands with their back facing the X-ray machine, and the X-rays travel from the back of the patient towards the front, penetrating through the chest and onto the detector positioned in front of the patient. Opposite to AP, in PA, the beam enters the patient's posterior chest wall and exits through the anterior chest wall, resulting in an image where the heart and closer to their actual size.

Chest radiography is a widely used imaging technique in contemporary medical practice. Various vendors manufacture CXR units and their sub-components, each with slightly different parameters influencing contrast and resolution, such as detector material, kVp, and mAs. The presence of multiple vendors with distinct parameters contributes to the variations in chest radiography outcomes. These differences can serve as shortcuts, which are further explored in Chapter 4. Moreover, imaging location plays a role in the patient-specific and image acquisition tags, as the patient population demographics and vendor distribution vary by location, with most hospitals and clinics using a specific vendor. Therefore, taking note of the imaging location can help identify potential biases in a dataset and prevent them from being used as shortcuts.

# 3.2 Non-COVID-19 Chest X-ray Datasets

#### 3.2.1 ImageNet

Fei-Fei Li, an AI researcher at Princeton University, started working on the concept of ImageNet in 2006 at a time when most AI research focused on algorithms and models. Li recognized the importance of having a large and diverse dataset to train AI algorithms and wanted to expand and improve the available data. In 2007, Li met with Christiane Fellbaum, a Princeton professor and one of the creators of WordNet, to discuss the project. Li decided to use the word database of WordNet as the foundation for ImageNet and incorporated many of its features to build the dataset.

ImageNet<sup>46</sup> is a vast image database created for research on visual object recognition software. The project has over 14 million images, all of which have been manually annotated to identify the objects depicted in them. With over 20,000 categories, each containing several hundred images, ImageNet provides a comprehensive dataset for training and testing object recognition algorithms. In addition, convolutional neural networks are often associated with ImageNet, as this dataset played a pivotal role in the swift advancement of deep learning research and its widespread success.

ImageNet is an important benchmark for comparing CNN and other deep learning algorithms. Additionally, ImageNet is commonly used for model pretraining, even for tasks outside the scope of natural images, such as medical imaging. This is the primary use of the ImageNet dataset in this work and is explored further in Chapter 5.

## 3.2.2 ChestX-ray14

Chest X-ray exams, while widely used and cost-effective compared to CT scans, can be more challenging to interpret due to the overlapping anatomy present in the images. The lack of large, publicly available datasets with annotations made it challenging to achieve clinically relevant computer-aided detection and diagnosis (CAD) for chest X-rays in real-world medical settings. To address this, the National Institutes of Health (NIH) released over 100,000 anonymized chest X-ray images<sup>100</sup> and their corresponding data to the scientific community. Before releasing the ChestX-ray14 dataset, Openi was the largest publicly available source of chest X-ray images, with only 4,143 images.

ChestX-ray14 represented a significant breakthrough in AI development in medical imaging,

consisting of 112,120 X-ray images (PNG images in 1024×1024 resolution) with disease labels from 30,805 unique patients. The authors used natural language processing (NLP) to extract fourteen disease classifications from the radiological reports associated with the X-rays, resulting in labels that are expected to be over 90% accurate.<sup>100</sup> The fourteen common thoracic pathologies include atelectasis, consolidation, infiltration, pneumothorax, edema, emphysema, fibrosis, effusion, pneumonia, pleural thickening, cardiomegaly, nodule, mass, and hernia.

Similar to ImageNet, ChestX-ray14 is frequently employed for pretraining deep network feature extraction modules, especially in medical imaging, owing to its status as one of the most extensive CXR datasets accessible to the public. Throughout the pandemic, the majority of AI solutions leveraged this dataset, either through pretraining or employing its images as COVID-19 negative cases. In this study, the dataset was used for pretraining to refine the convolutional filters, and its utilization is further examined in Chapter 5.

## 3.2.3 MIMIC Chest X-ray

The MIMIC Chest X-ray (MIMIC) Database<sup>101</sup> is a large publicly available dataset of chest radiographs in DICOM format with free-text radiology reports. The dataset contains 377,110 images corresponding to 227,835 radiographic studies performed at the Beth Israel Deaconess Medical Center in Boston, MA.

The creation of MIMIC required handling three distinct data modalities: electronic health record data, images (chest radiographs), and natural language (free-text reports). The authors queried the BIDMC EHR for chest X-ray studies made in the emergency department between 2011 - 2016, and extracted the set of patient identifiers associated with these studies.

After de-identification, images were exported in the JPEG standard format. First, the image pixels were extracted from the DICOM file, and pixel values were normalized to the range [0, 255] by subtracting the lowest value in the image, dividing by the highest value in the shifted image, truncating values, and converting the result to an unsigned integer. Next, histogram equalization was applied, and the images were written out in the compressed JPEG format with a quality value of 95.

Labels for the images were derived from either the impression section, the findings section (if the impression was not present), or the report's final section (if neither the impression nor findings

sections were present). Of the total 227,835 reports, 189,561 (83.2%) had an impression section, 27,684 (12.2%) had a findings section, and 10,514 (4.6%) had an equivalent section not explicitly labeled as findings or impression.

This work leveraged the MIMIC dataset for two important purposes. First, it provides a vast and diverse collection of cases, including a significant number of "normal" or "no findings" diagnoses. Moreover, the dataset was collected between 2011-2016, ensuring that the COVID-19 pandemic did not influence it. These factors were essential in training the "shortcut detective" models, which are extensively discussed in Chapter 4. Second, the dataset was used to train the opacity filter network, which screened CXR images based on lung opacity. Chapter 5 presents more details on this.

## 3.3 COVID-19 Chest X-ray Datasets

#### 3.3.1 Henry Ford COVID-19 Dataset

To train a deep-learning COVID-19 classification network, an institutional review board (IRB)approved study compliant with HIPAA was conducted at the Henry Ford Health System (HF) in the greater Detroit area and the University of Wisconsin-Madison. The study involved acquiring chest radiographs from patients with and without COVID-19 from HF, comprising five major hospitals and over 30 surrounding clinics.

Only patients who underwent frontal view CXR with confirmed COVID-19 diagnosis via RT-PCR testing and performed imaging between March 1, 2020, and September 30, 2020, were included in the study. The data collection included both COVID-19 positive and COVID-19 negative cases during the same time frame to ensure that the data distribution represents the patient cohorts in which the algorithm would be deployed. This approach also eliminates potential shortcut learning risks when using pre-pandemic data as the control group (see Chapter 4).

Chest radiography was performed using imaging systems from various vendors, including Agfa (3543EZE, CR25, CR30, CR30-X, DX-G, DXD30-Wireless, DXD40-1000C), Carestream Health (Classic CR, CR0850A, CR0975, DRX-1, DRX-Revolution), Fujifilm (5000D), GE Healthcare (Definium 5000, Geode Platform, Thunder Platform, Discovery XR656, Optima XR220, Optima XR240, WDR1), Kodak (Classic CR, CR850A), Konica Minolta (AeroDR, CS-7), Philips (DigitalDiagnost, MobileDiagnost), and Siemens Healthineers (Fluorospot Compact FD).



Figure 3.3: The Henry Ford Health System covers the greater Detroit region in Michigan, USA.

The total HF dataset includes 10,079 COVID-19 positive CXRs from 4,545 patients and 19,292 COVID-19 negative CXRs from 9,272 patients. However, for the purposes of model training and evaluation, additional exclusion criteria were used. First, patients under the age of 18 were removed to avoid cases where anatomical differences from the typical COVID-19 patient could significantly impact the results. To ensure the imaging data were consistent with the RT-PCR test, CXRs were excluded if performed more than seven days before or after the RT-PCR test, i.e., a delta window of  $\Delta = [-7, 7]$  days. After these additional exclusion criteria were applied, the resulting dataset included 8,335 COVID-19 positive CXRs from 4,383 patients and 16,584 COVID-19 negative CXRs from 8,733 patients. Demographic information for the patients and the dataset is provided in Figure 3.4 and a summary in Table 3.1.

In order to further explore the imaging characteristics of the HF dataset, presented in Figure 3.5 is the averaged radiograph of 5,000 randomly selected COVID-19 positive and COVID-19 negative cases, along with 25 randomly chosen CXR examples. Upon qualitative inspection, one can observe no significant difference between the two classes.

Furthermore, the pixel intensity distributions of each class are depicted in the left panel of Figure 3.6. Also compared are the pixel intensity distributions of the four major vendors in the HF dataset (Agfa, Carestream, GE, and Konica Minolta) to the overall distribution (right panel of Figure 3.6). There is no significant difference in pixel intensity distributions between the two classes or among vendors. Each vendor, however, displays a characteristic pixel intensity distribution.



Figure 3.4: Dataset population characteristics of the Henry Ford COVID-19 dataset.



**Figure 3.5:** Average image (with 25 random examples) from 5,000 randomly sampled images from the HF dataset for COVID-19 positive (left) and COVID-19 negative (right).

This work uses the HF dataset as the primary COVID-19 dataset due to several key features that make it advantageous for developing a deep-learning model. First, it is a relatively large dataset compared to other curated COVID-19 datasets, and it includes a diverse cohort of patients, which can enhance the model's generalizability. Second, the dataset contains both COVID-19 positive and negative cases collected from the same sources, which reduces label bias. This concept is explored further in Chapter 4. Additionally, unlike most public datasets, the original DICOM file and metadata are available for each image in the HF dataset. This enables better control of window/leveling and overall image quality, which can improve the model's accuracy.



**Figure 3.6:** Average image pixel intensity PDF for the HF dataset (left) and the four most contributing vendors Agfa, Carestream, GE Healthcare, and Konica Minolta (right).

	Total	<b>COVID-19</b> Positive	COVID-19 Negative
No. Images	24,919	8,335	16,584
No. Patients	13,116	4,383	8,733
Patient Sex	48% / 52%	17% / 53%	18% / 52%
(M/F)	40707 3270	47 /0 / 00 /0	10/0 / 02/0
Patient Age	$63.7\pm16.8$	$59.9 \pm 17.9$	$65.6 \pm 15.9$
<b>View Position</b>	73% / 7% / 20%	78% / 3% / 19%	70% / 9% / 21%
(AP/PA/Unknown)	7370777072070	70/07 5/07 19/0	70/07 9/07 21/0
Modality	51% / 16%	18% / 52%	57% / 12%
(CR/DX)	31/0 / 10/0	40/07 52/0	57 /0 / 42 /0
Imaging Vendor	7% / 56% /	7% / 54% /	7% / 57% /
(Agfa/Carestream/	20% / 16% /	19% / 19% /	20% / 14% /
Konica Minolta/GE/	20707 10707	<b>1</b> 70/ <b>1</b> 70/	20/0 / 14/0 /
Others)	2/0	2 /0	۷/۵

Table 3.1: A summary of the demographic information for the HF COVID-19 dataset used in this work.

## 3.3.2 Henry Ford Temporal COVID-19 Dataset

To ensure an accurate assessment of model performance on prospective data, it is essential to evaluate a dataset collected in a period after the training data. This type of dataset is known as an internal temporal test set. This study used a temporal test dataset from the same Henry Ford Health System hospitals as the previous dataset, consisting of patient cases received in October 2020 (note that the HF dataset comprised of cases from March 2020 to September 2020). Only CXRs performed close to the RT-PCR test within a narrow delta window of  $\Delta = [-3, 3]$  days, and patient age  $\geq 18$  were included to maintain testing integrity. This approach ensured that the test set accurately reflected the time frame and conditions of the training data, enabling a comprehensive evaluation of the trained model's performance. The resulting dataset comprised 695 COVID-19 positive CXRs from 526 patients and 8,878 COVID-19 negative CXRs from 6,081 patients. For more details, see Figure 3.7 for demographic information on the patients and a summary in Table 3.2.

The averaged radiograph of 600 randomly selected COVID-19 positive and COVID-19 negative cases and 25 randomly chosen CXR examples from the HF temporal dataset are presented in Figure 3.8. In addition, the pixel intensity distributions of COVID-19 positive and negative classes are depicted in Figure 3.9, as well as the distributions for the major vendors (Agfa, Carestream, GE, and Konica Minolta) used in the dataset.



Figure 3.7: Dataset population characteristics of the HF temporal COVID-19 dataset.



**Figure 3.8:** Average image (with 25 random examples) from 600 randomly sampled images from the HF temporal dataset for COVID-19 positive (left) and COVID-19 negative (right).



**Figure 3.9:** Average image pixel intensity distribution for the HF temporal dataset and the four most contributing vendors Agfa, Carestream, GE Healthcare, and Konica Minolta.

	Total	COVID-19 Positive	COVID-19 Negative
No. Images	9,573	695	8,878
No. Patients	1,134	526	6,081
Patient Sex (M/F)	48% / 52%	53% / 47%	47% / 53%
Patient Age	$61.7 \pm 18.9$	$60.9 \pm 18.9$	$61.7 \pm 18.9$
View Position (AP/PA/Unknown)	69% / 9% / 22%	75% / 9% / 15%	68% / 9% / 22%
Modality (CR/DX)	62% / 38%	49% / 51%	62% / 38%
Imaging Vendor (Agfa/Carestream/ Konica Minolta/GE/ Others)	15% / 56% / 18% / 7% / 4%	17% / 64% / 14% / 4% / 2%	15% / 54% / 18% / 7% / 4%

**Table 3.2:** A summary of the demographic information for the HF temporal COVID-19 dataset used in this work.

## 3.3.3 Valencian Region Medical ImageBank COVID-19+ Dataset

The BIMCV-COVID19+ dataset contains chest X-ray images CXR (CR, DX) and CT imaging of COVID-19 patients along with their radiographic findings, pathologies, polymerase chain reaction, immunoglobulin G (IgG) and immunoglobulin M (IgM) diagnostic antibody tests and radiographic reports from the Medical Imaging Databank in Valencian Region Medical Image Bank (BIMCV).<sup>102</sup>

The BIMCV COVID-19+ dataset is a large open multi-institutional databank collected from 11 hospitals in the Valencian region, Spain, between February and April 2020. Raw pixel data were extracted from the DICOM images and stored in files in nii.gz format. When available, the images were processed by rescaling the dynamic range using the DICOM window width and center and stored as 16-bit PNG images. The images were not rescaled to avoid loss of resolution. The information on image projection was estimated using a neural network. The total available data collected contains 1,311 subjects, 2,429 image studies, and 5,530 image series with 1,380 CR, and 885 DX studies. A total of 2,425 PCR tests have been performed on the patients, of which 1,773 were positive for COVID-19, 622 negative for COVID-19, and 30 indeterminate.

Chest radiography was performed using imaging systems from various vendors, including 3DISC (QuantorMed), Agfa (3543EZE, ADC Compact, ADC Compact Plus, CR 30, CR 75, CR 85, CR15-X, CR30-X, DR 14e C, DX-G, DX-M, DXD30-Wireless, DXD40-1000C, DXD40-1000G,



Figure 3.10: The BIMCV dataset was collected from 11 hospitals in the Valencian Region, Spain. From https://maigva.github.io/maps/HealthDepartCOVID19.html

Pixium-4343E-CSI), Canon Inc. (CXDI Control Software NE), Carestream Health (DRX-1, DRX-Evolution, DRX-Revolution, Image Suite Vita CR System), Digitec (DigiRad, MobilRad, RX-ELX04), Fujifilm Corporation, GE Healthcare (Thunder Platform, Discovery XR656, Optima XR220), GMM (ACCORD DR), Kodak (CR850A, CR975, DRX-EVOLUTION, ELITE CR), Konica Minolta (110, 5601, 862, CS-7), Philips Medical Systems (DigitalDiagnost, Essenta DR, PCR Eleva), Radiologia S.A., Sedecal, Siemens (Fluorospot Compact FD, FD-X), and Varian (4343R, 4343R-DRZ).

Similar to the HF temporal dataset, patient age  $\geq$  18 and  $\Delta$  = [-3, 3] inclusion criteria were applied to the BIMCV dataset. The resulting dataset included 4,169 COVID-19 positive CXRs from 2,663 patients and 5,050 COVID-19 negative CXRs from 3,710 patients. Demographic information for the patients and the dataset is provided in Figure 3.11 and a summary in Table 3.3.

The averaged radiograph of 5,000 randomly selected COVID-19 positive and COVID-19 negative cases and 25 randomly chosen CXR examples from the BIMCV dataset are presented in Figure 3.12. Note that markers are present in the averaged image (indicated by the white arrows in Figure 3.12), indicating that a significant portion of the images contains these markers.

The pixel intensity distributions of each class are depicted in Figure 3.13. Also compared are the pixel intensity distributions of the four major vendors in the BIMCV dataset (Agfa, Carestream, Konica Minolta, and Philips) to the overall distribution (right panel of Figure 3.13). There is a slight difference in pixel intensity distributions between the two classes compared to the HF datasets.


Figure 3.11: Dataset population characteristics of the BIMCV COVID-19 dataset.



**Figure 3.12:** Average image (with 25 random examples) from 600 randomly sampled images from the BIMCV dataset for COVID-19 positive (left) and COVID-19 negative (right). Note the residual markers in the average image highlighted by the boxes.

However, each vendor displays a characteristic pixel intensity distribution, as expected.

The BIMCV dataset played a multifaceted role in this work, effectively complementing the HF dataset across various applications. Originating from diverse regions throughout Spain, the BIMCV dataset boasts a distinctly different patient population and vendor distribution compared to the HF dataset, which was collected in Detroit, Michigan. Consequently, the BIMCV dataset serves as an authentic external test set, enabling a thorough evaluation of model generalization. The issues of bias and shortcuts between the BIMCV and HF datasets are extensively discussed in Chapters 4 and 6.



**Figure 3.13:** Average image pixel intensity distribution for the BIMCV dataset and the four most contributing vendors Agfa, Carestream, Konica Minolta, and Philips.

	Total	<b>COVID-19</b> Positive	COVID-19 Negative
No. Images	9,219	4,169	5,050
No. Patients	6,373	2,663	3,710
Patient Sex	51% / 49%	54% / 46%	19% / 51%
(M/F)	51/0 / 49/0	01/0/10/0	49707 5170
Patient Age	$64.6 \pm 17.7$	$64.3\pm16.9$	$64.8 \pm 18.4$
<b>View Position</b>	10% / 11% / 7%	56% / 30% / 5%	110/ 1 180/ 1 80/-
(AP/PA/Unknown)	1)/0/11/0/7/0	30787 37787 378	<b>H</b> /0/ <b>H</b> /0/ 0/0
Modality	50% / 50%	52% / 18%	18% / 52%
(CR/DX)	30787 3078	32/07 40/0	40707 5270
Imaging Vendor	37% / 15% /	16% / 13% /	70% / 16% /
(Agfa/Carestream/	0% / 10% /	110/ / 70/ /	29/0 / 10/0 /
Konica Minolta/Philips/	9% / 10% /	11 /o / / /o /	0 /0 / 12 /0 /
Siemens/Others)	0/0/21/0	0 /0 / 17 /0	10 /0 / 24 /0

Table 3.3: A summary of the demographic information for the BIMCV COVID-19 dataset used in this work.

## 3.3.4 University of Wisconsin-Madison COVID-19 Dataset

The University of Wisconsin-Madison (UW) COVID-19 dataset contains patient CXR cases from the UW hospitals and clinics collected from March 2020 to September 2021. Due to the smaller area and corresponding population coverage, the UW dataset was primarily used as a test dataset to evaluate the performance and generalizability of deep learning networks trained on the larger COVID-19 datasets. To maintain testing integrity, only CXRs performed close to the RT-PCR test within a narrow delta window of  $\Delta$  = [-3, 3] days, and patient age  $\geq$  18 were included. The resulting dataset included 1,025 COVID-19 positive CXRs from 658 patients and 8,774 COVID-19 negative



Figure 3.14: UW Health Hospital in Madison, Wisconsin.



**Figure 3.15:** Average image (with 25 random examples) from 600 randomly sampled images from the UW dataset for COVID-19 positive (left) and COVID-19 negative (right).

CXRs from 5,953 patients. Demographic information for the patients and the dataset is provided in Figure 3.17 and a summary in Table 3.4.

The averaged radiograph of 1,000 randomly selected COVID-19 positive and COVID-19 negative cases and 25 randomly chosen CXR examples from the UW dataset are presented in Figure 3.15. In addition, the pixel intensity distributions of COVID-19 positive and negative classes are depicted in Figure 3.16, as well as the distributions for the four major vendors used in the dataset (Philips, Fujifilm, Siemens, and Samsung).



**Figure 3.16:** Average image pixel intensity distribution for the UW dataset and the four most contributing vendors Philips, Fujifilm, Siemens, and Samsung.



Figure 3.17: Dataset population characteristics of the UW COVID-19 dataset.

	Total	COVID-19 Positive	COVID-19 Negative
No. Images	9,799	1,025	8,774
No. Patients	6,611	658	5,953
Patient Sex	52% / 48%	56% / 44%	52% / 48%
Patient Age	$58.9 \pm 19.2$	$55.1 \pm 19.2$	$59.3 \pm 19.2$
View Position (AP/PA/Unknown)	80% / 8% / 12%	86% / 4% / 10%	80% / 8% / 12%
Modality (CR/DX)	20% / 80%	15% / 85%	21% / 79%
Imaging Vendor (Philips/Fujifilm/ Samsung/Siemens/ GE)	78% / 12% / 5% / 4% / 2%	84% / 10% / 3% / 2% / 1%	77% / 12% / 5% / 4% / 2%

Table 3.4: A summary of the demographic information for the UW COVID-19 dataset used in this work.

#### 3.3.5 Medical Imaging and Data Resource Center COVID-19 Dataset

The Medical Imaging and Data Resource Center (MIDRC) is a multi-institutional collaborative initiative funded by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) and hosted at the University of Chicago, and is co-led by the American College of Radiology (ACR), the Radiological Society of North America (RSNA), and the American Association of Physicists in Medicine (AAPM).

Leveraging the existing and developing infrastructure provided by the participating organizations, MIDRC serves as a linked-data commons that coordinates access to data and harmonizes data management activities at three critical stages: (1) intake, including curation, de-identification, abstraction, and quality assessment (2) annotation and labeling of images and other data using



**Figure 3.18:** MIDRC is a multi-institutional collaborative initiative hosted at the University of Chicago with images from hundreds of institutions across the United States. From https://www.midrc.org/donate.



**Figure 3.19:** Average image (with 25 random examples) from 6,000 randomly sampled images from the MIDRC dataset for COVID-19 positive (left) and COVID-19 negative (right).

semi-automated approaches and (3) distributed access and query methods.<sup>103</sup>

The imaging data is collected from many sources, including academic medical centers, community hospitals, and others across the United States. At the time of writing, the MIDRC dataset contains over 117,258 imaging studies released to the public which focuses on imaging and data of COVID-19 patients. To date, the imaging studies made available have mainly been chest radiography.

MIDRC has performed extensive equity, diversity, and inclusion studies on their dataset, seeking to provide an unbiased, representative health data resource for all, lowering the possibility of statistical fallacies and representational errors.





Figure 3.20: Average image pixel intensity distribution for the MIDRC dataset.

stated by the organization, "MIDRC possesses a unique level of expertise related to medical imaging and is therefore pursuing an approach which will look to optimize the value of combining medical images with the clinical data typically being analyzed by other COVID-19 groups."<sup>103</sup> The dataset is unique in that many cases contain information about patient race, a population characteristic not present in any of the other datasets. However, MIDRC does not release any information on the imaging vendor. Applying the same filters of patient age  $\geq$  18 and  $\Delta$  = [-3, 3], the resulting included 6,453 COVID-19-positive CXRs from 5,199 patients and 20,072 COVID-19-negative CXRs from 9,947 patients. Demographic information for the patients and the dataset is provided in Figure 3.21 and a summary in Table 3.5.

The averaged radiograph of 6,000 randomly selected COVID-19 positive and COVID-19 negative cases, along with 25 randomly chosen CXR examples, from the MIDRC dataset are presented in Figure 3.19. The pixel intensity distributions of COVID-19 positive and negative classes are depicted in Figure 3.20, however, there is no vendor information to compare.

The MIDRC dataset was used in various aspects of this thesis work. Like the other datasets outside of HF, MIDRC was used to test the generalizability of a baseline deep learning network (see Chapter 5) as well as study race as a potential bias in datasets (see Chapter 4). MIDRC also hosted a COVID-19 classification challenge, COVIDx Challenge, in which participants competed to develop the best-performing model. The methods developed and studied in this thesis helped our team to win this challenge and are elaborated further in Chapter 5.

	Total	COVID-19 Positive	COVID-19 Negative
No. Images	26,525	6,453	20,072
No. Patients	15,146	5,199	9,947
Patient Sex	11% / 13% / 13%	18% / 15% / 7%	12% / 13% / 15%
(M/F/Other or Unknown)	<b>H</b> /0 / <b>H</b> /0 / <b>H</b> /0 / <b>H</b> /0	10/0 / 10/0 / 7/0	42/0/40/0/10/0
Patient Age	$57.4 \pm 17.7$	$56.0\pm17.9$	$58.2 \pm 17.5$
Patient Race	0.20/ / 100/ /	0.20/ / 00/ /	0.00//100//
(American Indian/Asian/	0.3% / 10% /	0.3%/9%/	0.2%/ 10% /
Black/Pacific Islander/	22% / 0.4% /	31% / 0.3% /	18% / 0.4% /
White/Other/Unknown)	48% / 6% / 14%	40% / 8% / 10%	51% / 4% / 16%
<b>View Position</b>	860/ / 60/ / 80/	Q10/ / Q0/ / Q0/	86% / 6% / 8%
(AP/PA/Unknown)	00/0 / 0/0 / 0/0	04/0/0/0/0/0/0	00/0 / 0/0 / 0/0
Modality		20% / 80%	21% / 60%
(CR/DX)	27/0 / / 1 /0	20 /0 / 00 /0	51/0 / 09/0

Table 3.5: A summary of the demographic information for the MIDRC COVID-19 dataset used in this work.



Figure 3.21: Dataset population characteristics of the MIDRC COVID-19 dataset.

#### 3.3.6 COVIDx CXR Dataset

The final dataset examined in detail in this thesis is the COVIDx dataset, <sup>104</sup> a result of collaboration between researchers at the University of Waterloo, DarwinAI, and other contributors. The dataset is continually updated with new data from additional sources. This large, open-source CXR dataset is designed to facilitate the development of machine-learning models to detect COVID-19. The dataset was created by combining multiple sources of CXR images, including images of patients with COVID-19, pneumonia, and healthy controls. The primary sources of the COVID-19 negative images include the NIH dataset previously discussed and the RSNA Pneumonia Detection Challenge dataset. The positive images come from various public sources such as research articles, websites, and tweets.<sup>105</sup>

The COVIDx dataset employs a unique curation approach compared to other COVID-19 datasets previously discussed, and it has been subject to several critiques from the community.<sup>85,106–108</sup> The majority of the images are sourced from public repositories, resulting in limited metadata availability, most notably the confirmation of COVID-19 infection through verified PCR tests. Moreover, the images, which do not originate from DICOMs, display a wider range of quality compared to other COVID-19 datasets. Notably, the negative images are derived from various sources, from periods before the pandemic. This data collection method may render models trained on the COVIDx dataset more susceptible to shortcut learning.

A total of 29,986 CXRs from 16,648 patients are included in the dataset, including 16,490 positive COVID-19 images from over 2,800 patients. There is no accompanying metadata available, so only image comparison will be presented. The averaged radiograph of 5,000 randomly selected COVID-19 positive and COVID-19 negative cases and 25 randomly chosen CXR examples from



**Figure 3.22:** Average image (with 25 random examples) from 5,000 randomly sampled images from the COVIDx dataset for COVID-19 positive (left) and COVID-19 negative (right).



Figure 3.23: Average image pixel intensity distribution for the COVIDx dataset.

the dataset are presented in Figure 3.22. In addition, the pixel intensity distributions of COVID-19 positive and negative classes are depicted in Figure 3.23.

Compared to the professionally curated COVID-19 datasets discussed previously, the COVIDx dataset has much more image heterogeneity, as evidenced by the average images and pixel-intensity PDFs. The COVIDx dataset was used in this work to highlight the dangers of shortcut learning and the importance of dataset analysis and curation. This is further examined in Chapter 4.

# 3.4 Analysis and Comparison of COVID-19 Datasets

This section provides a concise analysis and comparison of the various COVID-19 datasets. As briefly mentioned in Section 2.2.7 and further supported by the research presented in this thesis, deep learning models are heavily influenced by the quality and characteristics of the data they are trained on. If the data is of low quality or contains biases, these models may inadvertently learn undesired features that fail to generalize to new data. Consequently, conducting a comprehensive analysis of the datasets used for training and validating models is crucial. While the notion of "big data" carries some validity, blindly combining data solely to increase dataset size can be counterproductive. Instead, several methods can be employed to analyze datasets, including examining metadata and

		Ĺ		-	ŀ	-									
	C	IF Datas	lei	Ē	- Iempo	rai		BIMCV			<b>у</b> меа	th		MIDKC	
Inclusion Criteria	Age	Frontal view ≥ 18; Δ = [-	; 7, 7]						Frontal ∖ Age ≥ 18; Δ	view: = [-3, 3]					
	Total	Positive	Negative	Total	Positive	Negative	Total	Positive	Negative	Total	Positive	Negative	Total	Positive	Negative
No. Images	24,919	8,335	16,584	9,573	695	8,878	9,219	4,169	5,050	9,799	1,025	8,774	26,525	6,453	20,072
No. Patients	13,116	4,383	8,733	1,134	526	6,081	6,373	2,663	3,710	6,611	658	5,953	15,146	5,199	9,947
Patient Age	64 ± 17	60 ± 18	66 ± 16	62 ± 19	61 ± 19	62 ± 19	65 ± 18	64 ± 17	65 ± 18	59 ± 19	55 ± 19	59 ± 19	57 ± 18	56 ± 18	58 ± 18
Patient Sex % (M/F/Unk)	48/52	47/53	48/52	48/52	53/47	47/53	51/49	54/46	49/51	52/48	56/44	52/48	44/43/13	48/45/7	42/43/15
View Position % (AP/PA/Unk)	73/7/20	78/3/19	70/9/21	69/9/22	75/9/15	68/9/22	49/44/7	56/39/5	44/48/8	80/8/12	86/4/10	80/8/12	86/6/8	84/8/8	86/68
Modality % (CR/DX)	54/46	48/52	57/42	62/38	49/51	62/38	50/50	52/48	48/52	20/80	15/85	21/79	29/71	20/80	31/69
Imaging Vendor %	Carestream (56) Konica (20) GE (16) Agfa(7) Other (2)	) Carestream (54) Konica (19) GE (19) Agfa(7) Other (2)	Carestream (57) Konica (20) GE (14) Agfa(7) Other (2)	Carestream (56) Konica (18) Agfa (15) GE (7) Other (4)	Carestream (64) Konica (14) Agfa (17) GE (4) Other (2)	Carestream (54) Konica (18) Agfa (15) GE (7) Other (4)	Agfa (37) Carestream (15) Philips (10) Konica (9) Siemens (8) Other (21)	Agfa (46) Carestream (13) Philips (7) Konica (11) Siemens (6) Other (17)	Agfa (29) Carestream (16) Philips (12) Konica (8) Siemens (10) Other (24)	Philips (78) Fujifilm (12) Samsung (5) Siemens (4) GE (2)	Philips (84) Fujfilm (10) Samsung (3) Siemens (2) GE (1)	Philips (77) Fujifilm (12) Samsung (5) Siemens (4) GE (2)	N/A	A/N	N/A
Patient Race %	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	A/N	White (48) Black (22) Unknown (14) Asian (10) Other (6) Pacific Is. (0.4) Am. Indian (0.3)	White (40) Black (31) Unknown (10) Asian (9) Other (8) Pacific Is. (0.3) Am. Indian (0.3)	White (51) Black (18) Unknown (16) Asian (10) Other (4) Pacific Is. (0.4) Am. Indian (0.2)

-

Figure 3.24: Table of the available metadata for the COVID-19 datasets used in this work.



**Figure 3.25:** Average pixel intensity distribution for COVID-19 positive and negative images based on data source.

the images themselves.

Figure 3.24 compares patient demographics and other available metadata from the COVID-19 datasets utilized in this study. Significant disparities in patient demographics or dataset attributes identified through metadata analysis should be considered when examining a network for biases and shortcut learning.

#### 3.4.1 Comparison of Image Contrast

In this section, some of the image characteristics of the COVID-19 datasets are analyzed. Images are fundamentally characterized by their contrast and resolution, making these features essential for comparing different image datasets. Figure 3.25 presents the average pixel intensity PDF for COVID-19 positive and negative images based on the data source. As anticipated, there is some variability in average image intensity due to varying population demographics, imaging vendors, and protocols. Notably, significant differences between the datasets can be observed at the distributions' low (dark), and high (bright) ends. Some of these variances can be attributed to specific image characteristics related to their dataset origin, such as image background color, image text, markers covering text, and window and leveling protocols, as shown in Figure 3.26.



**Figure 3.26:** Some examples of image differences based on dataset source, including image background color, image text, or markers covering text, as well as window and leveling protocols.

A more complete understanding can be obtained by examining the spatial differences in contrast for images from various datasets. Figures 3.27 - 3.30 underscore the differences in the average (average pixel intensity across all images in a dataset) and standard deviation (pixel standard deviation across all images in a dataset) image for each dataset. From these figures, several key features emerge. First, it is evident that each dataset has a different mean intensity. In the difference images in Figure 3.27, patient de-identification boxes are visible in the shoulder regions for the HF and HF temporal datasets, along with overall background mean intensity differences. The standard deviation images reveal more significant pixel variation, particularly in the BIMCV dataset. Generally, heterogeneity in contrast between the different datasets is observed, which is to be expected. Nonetheless, it is crucial to recognize these differences, as they can be readily exploited by neural networks as shortcuts, as demonstrated in Chapter 4.

### 3.4.2 Comparison of Edges and High-Frequency Content

Spatial resolution is another vital aspect of images. Although there are numerous spatial resolution metrics, the modulation transfer function (MTF) is considered the gold standard for evaluating an imaging system. However, since MTF cannot be extracted solely from a chest X-ray image, edge detection is employed as an alternative method for providing a very rough comparison of relative sharpness between datasets. Higher spatial resolution leads to more accurate edge detection and improved overall image quality. In contrast, low spatial resolution may result in blurred or lost edges, causing the image to appear pixelated or lacking in detail. It is important to note that while edges do not directly define spatial resolution, they are influenced by it. Other factors such as image markers or text, noise, medical equipment, jewelry, and different anatomical structure can also influence edge detection. With this in mind, edges are extracted using the Canny algorithm<sup>109</sup>



Average COVID-19 Positive Image Comparison (row: left half, column: right half)

Average COVID-19 Negative Image Comparison (row: left half, column: right half)

**Figure 3.27:** A comparison of the average images (obtained by averaging pixel intensities over all images) for each dataset. The left half of each individual image corresponds to the row label and the right to the column label.



**Figure 3.28:** A comparison of the average image difference images (obtained by averaging pixel intensities over all images) for each dataset. Each individual image is obtained by taking the row label and subtracting the column label.



**Figure 3.29:** A comparison of the standard deviation images (obtained by taking the standard deviation of pixel intensities over all images) for each dataset. The left half of each individual image corresponds to the row label and the right to the column label.



**Figure 3.30:** A comparison of the standard deviation difference images (obtained by taking the standard deviation of pixel intensities over all images) for each dataset. Each individual image is obtained by taking the row label and subtracting the column label.



Figure 3.31: Example CXR and the extracted edges using the Canny method.

and compared across datasets.

The Canny algorithm begins by applying a Gaussian filter to smooth the image and remove excess noise. Subsequently, the intensity gradients of the image are calculated. Next, a gradient magnitude threshold is applied to minimize spurious responses to edge detection. A double threshold is then applied to filter out edge pixels with weak gradient values and preserve those with high gradient values. This is achieved by selecting high and low threshold values. An edge pixel with a gradient value higher than the high threshold value is marked as a strong edge pixel. If an edge pixel's gradient value falls between the high and low threshold values, it is marked as a weak edge pixel. Edge pixels with gradient values below the low threshold value are suppressed. Finally, the remaining edges are tracked by hysteresis, suppressing all other weak edges that are not connected to strong edges.

There are certain limitations when applying this method to CXR images, as inconsistent windowleveling between images can impact edge detection. Moreover, while this method is somewhat robust to noise, CXR images with significant noise will lead to more extracted edges. To address the window-leveling issue, histogram equalization was applied to each image before edge detection. After equalization, the Canny algorithm was applied with empirically selected  $\sigma = \sqrt{2}$ , an upper threshold of 0.060, and a lower threshold of 0.024. The resulting binary edge image was averaged



**Figure 3.32:** The mean edge intensity distribution, obtained by averaging the binary edge image, for the COVID-19 datasets used in this work.

across all dimensions to obtain the mean edge intensity. Figure 3.31 displays an example CXR and its extracted edges. A histogram of the mean edge intensity for each dataset is presented in Figure 3.32 and reveals that, except for UW and COVIDx, the datasets have comparable edge content.

One explanation for the secondary peak in the UW dataset is the qualitative observation that many images in this peak exhibit higher noise levels compared to other datasets. In addition, the presence of a relatively large number of high-contrast images in the COVID-19 positive class (see Figure 3.22) in the COVIDx dataset contributes to the secondary peak present in the positive class.

It has been shown that CNNs are sensitive to high-frequency content in images, <sup>110–113</sup> and specific targeted high frequencies can actually completely change the prediction of the network without visibly changing the image; this is known as an adversarial attack. <sup>113,114</sup> In this regard, a simple comparison of the high-frequency components of the different datasets is presented. The 2-D Fourier transform of each image using the fast Fourier transform (FFT) was taken, and the zero-frequency component was shifted to the middle of the spectrum. An averaged magnitude image was calculated for each dataset, and to aid in visualization due to the large dynamic range, the log transform of each image is presented in Figure 3.33 and the differences in Figure 3.34.

These figures show noticeable differences in the high-frequency components of the MIDRC and COVIDx datasets compared to the others. Note that high-frequency components in an image's



**Figure 3.33:** Comparison of the average Fourier transform for each dataset (log transform taken to compress dynamic range). The left half of each individual image corresponds to the row label and the right to the column label.



**Figure 3.34:** Comparison of the average Fourier transform difference for each dataset. Each individual image is obtained by taking the row label and subtracting the column label.

Fourier transform result from rapid changes in intensity or pixel values across the image. These rapid changes often correspond to noise, edges, textures, and sharp features (e.g., text or boxes) in the image. However, they can also be attributed to intricate details in the patient anatomy or vendor image processing techniques, such as sharpening or edge enhancement.

### 3.4.3 Deep Learning Feature Extraction

Another approach to comparing the COVID-19 datasets involves leveraging a CNN for feature extraction and subsequently analyzing the resulting features. As reviewed in Section 2.2.4, CNNs consist of several layers, including convolutional, pooling, and fully connected layers. When a CNN is trained on a large dataset of images, it learns to automatically extract hierarchical features by passing the images through these layers. The early layers of the network typically capture low-level features like edges, textures, and simple shapes. In contrast, the deeper layers capture more complex, high-level features like object parts or even whole objects.

Once the CNN is trained or fine-tuned, it can be used to extract features from new images. This is typically done by removing the last fully connected layer(s) responsible for the classification task and using the output from the remaining layers as the feature representation of the input image (see Figure 3.35). These extracted features can be used as input for other machine learning algorithms, such as support vector machines or dimensionality reduction, to perform various image-related tasks. The advantage of deep learning for feature extraction is that the CNN can automatically



**Figure 3.35:** An example of how a CNN can be used to extract high-level features from images. The convolutional layers extract image features which are then classified using fully connected layers. By removing the fully connected layers, the high-dimensional features extracted by the network can be obtained to be further analyzed.

learn the most relevant and discriminative features for the task at hand, often resulting in better performance than traditional hand-engineered features.

To facilitate the comprehension of high-dimensional extracted features, several dimensionality reduction techniques, such as Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP), can be employed to visualize and analyze high-dimensional data effectively. In the following, to aid in visualizing the extracted features, dimensionality reduction was accomplished using the UMAP<sup>115</sup> algorithm, which is particularly effective for visualizing.

UMAP is based on manifold learning and topological data analysis, making it suitable for preserving local and global data structures. UMAP starts by creating a high-dimensional graph representation of the data. For each data point, it identifies its nearest neighbors based on a chosen distance metric (in this work, the Euclidean distance was used). It then computes the probabilities of connections between each data point and its neighbors, resulting in a weighted graph. This graph represents the local structure of the data in a fuzzy topological way, which means that it preserves the relationships between neighbors while allowing some uncertainty.

Next, UMAP aims to find a low-dimensional representation of the data that is as close as possible to the original high-dimensional graph. To achieve this, it constructs a low-dimensional graph (2-D in this work) and optimizes its layout by minimizing the cross-entropy between the two graphs (the high-dimensional and low-dimensional graphs). Cross-entropy is a measure of the dissimilarity between two probability distributions, and minimizing it ensures that the low-dimensional graph is a good approximation of the original graph (see Section 2.2.5.1). Finally, UMAP uses gradient descent (see Section 2.2.5.3) to optimize the positions of the data points in the low-dimensional space, resulting in a visualization that maintains both local and global structures.

Supervised UMAP is a modification of the original unsupervised UMAP algorithm, which incorporates label information into the learning process, aiming to improve the separation of different classes in the lower-dimensional representation. In supervised UMAP, the class labels are used to create additional constraints on the optimization process, which can lead to better separation of classes and more meaningful visualizations. It can be particularly useful in cases where there is prior knowledge about the data structure or the relationships between the data points, which can be incorporated to enhance the quality of the lower-dimensional representation.



**Figure 3.36:** UMAP clustering of the ImageNet pretrained ResNet-101 extracted features of the COVID-19 datasets labeled according to disease label, dataset source, and K-means cluster index.

In addition to UMAP, K-means clustering <sup>116,117</sup> was performed to analyze the extracted features. K-means clustering is an unsupervised machine learning algorithm that aims to partition data into K distinct, non-overlapping clusters based on the similarity between data points. Generally, it is used to capture meaningful structure, underlying processes, and grouping inherent in a dataset. The algorithm identifies cluster centroids (the centers of clusters) and assigns each data point to the cluster with the closest centroid. Each data point's label (cluster number K) was recorded and used as a label in the UMAP 2-D graph.

To accomplish deep learning feature extraction, a CNN (ResNet-101<sup>118</sup>) was trained on the ImageNet dataset (over 14 million images belonging to 20,000 classes). The final fully connected layer was removed and the activations of the final pooling layer, consisting of 2,048 features (i.e., values), were obtained for each image in the COVID-19 datasets highlighted previously. Supervised and unsupervised UMAP dimensionality reduction was performed as well as K-means clustering (K = 7) and are shown in Figures 3.36 and 3.37.

Numerous insights can be drawn from the feature analysis performed using deep learning. In a UMAP visualization, clusters represent groups of similar data points. Points situated close to one another in the UMAP plot exhibit similarities in the original high-dimensional space, while distant points are more dissimilar. Ideally, for a network trained on a binary classification dataset, the two classes would be perfectly distinguishable in a binary classification dataset, forming two separate and distinct clusters.

However, this model was not trained to distinguish between COVID-19 CXRs and so the unsupervised UMAP clustering of the total data in Figure 3.36 reveals a primary cluster with smaller subclusters and satellite clusters. The same pattern is observed for both COVID-19 positive and





**Figure 3.37:** Unsupervised (top) and supervised (bottom) UMAP dimensionality reduction of the ImageNet pretrained ResNet-101 extracted features of the COVID-19 datasets labeled according to dataset source and K-means cluster centroids (K = 7).

negative clustering. In addition, K-means clustering showcases a relatively distinct grouping of data within the UMAP clustering. To explore what features are prominent in these clusters, Figure 3.38 shows 25 randomly sampled images for six different regions of interest (ROIs) in the total data main cluster, six satellite clusters, as well as the seven clusters grouped by K-means clustering.

The main cluster's samples display comparable image characteristics, exhibiting minor contrast variations across the six distinct regions of interest (ROIs). In contrast, the satellite clusters reveal unique features such as cropped background colors, varying image contrasts, and enhanced textures. Therefore, these images can be regarded as outliers compared to the main cluster samples, and the visual differences are evident. Moreover, samples within the same centroid neighborhood in the K-means clustering demonstrate that images sharing similar qualities are grouped together.

When labeled by dataset source, the primary cluster comprises of a mixture of the sources, while the satellite clusters predominantly consist of images from specific datasets (UW-green, MIDRC-blue, and COVIDx-purple). This implies that the outlier images possess features that significantly differentiate them from the majority of the images. The supervised UMAP clustering further supports this observation (see Figure 3.37), where even when labels are provided, UW, MIDRC, and COVIDx form multiple clusters, suggesting that there is significant heterogeneity in the images within these datasets. To enhance the visualization of the distinct dataset substructure within the overall cluster, Figure 3.40 displays each cluster according to the dataset source.

The data heterogeneity, as evidenced by the satellite clusters and subclusters, can play a crucial role in the context of shortcut learning. Characteristics that cause these images to form separate clusters could potentially become shortcuts that a model learns. For instance, the largest satellite cluster consists of COVIDx images with a notably distinct contrast compared to all other images in the dataset (see Figure 3.38). It is worth noting that all of these images are COVID-19 positive (see Figure 3.37). Instead of learning disease-related features, a model may simply learn to classify all cases with this specific contrast scheme as positive. This correlation between image contrast and label is not generalizable and hinders the model's utility for the clinical task of identifying COVID-19 infections.

In the worst-case scenario, each dataset would have unique, discriminating features, resulting in a separate cluster for each dataset. These dataset-specific features could become shortcuts, and the model could entirely avoid learning any disease features. Fortunately, as shown in Figure 3.40,



**Figure 3.38:** To explore what features are prominent in the UMAP clustering, 25 randomly sampled images are shown for six different areas in the total data main cluster (top), six satellite clusters (middle), as well as the seven clusters grouped by K-means clustering (bottom).



**Figure 3.39:** UMAP clustering of the NIH ChestX-ray14 pretrained EfficientNet extracted features of the COVID-19 datasets labeled according to disease label, dataset source, and K-means cluster index.

images in the main cluster overlap, suggesting that most images in the datasets share similar features. One notable exception is the COVIDx dataset, which exhibits evident heterogeneity in both positive and negative classes, resulting in a dataset with a clear data bias. This biased data severely limits the performance of a model trained on it and supports the conclusions drawn in the critiques of the COVIDx dataset.<sup>85,106–108</sup> This dataset will be revisited in Chapter 4.

It is important to consider that the features extracted using a model trained on natural images, such as animals, plants, vehicles, and scenes, may differ significantly from medical chest X-rays. The features crucial for ImageNet classification tasks might not directly apply to medical imaging. Another CNN, EfficientNet,<sup>119</sup> was trained on the NIH ChestX-ray14 dataset to investigate this further. The model was designed to classify over 100,000 images into 14 different thoracic pathologies, a more closely related task detecting COVID-19 infection. Using the same strategies as in the ImageNet training presented earlier, the activations of the final pooling layer, comprising 1,280 features, were obtained for each image in the COVID-19 datasets. Supervised and unsupervised UMAP dimensionality reduction was performed, as well as K-means clustering (K = 7). The results can be seen in Figure 3.39 and Figure 3.42.

The clustering of features extracted from the NIH-trained EfficientNet differs significantly from those obtained using the ImageNet ResNet. Figure 3.42 displays a smoother and more homogeneous spherical cluster with a tapering point and only one outlier cluster. Moreover, the K-means clustering is well-organized and arranged vertically along the main cluster. Interestingly, the same shape and K-means clustering trend are also observed in the supervised clustering. To better understand the shared features in the K-means clustering and the outlier cluster, 25 randomly sampled images are shown in Figure 3.41 for the outlier cluster and each K-means cluster centroid.



**Figure 3.40:** UMAP reduction of features extracted from the COVID-19 datasets using the ImageNet pretrained ResNet-101 model, with labels according to dataset source. Each dataset source is displayed individually to enhance the visualization of their distinct structures.



**Figure 3.41:** To explore what features are prominent in the UMAP reduction, 25 randomly sampled images (framed by the label color) are shown for the one outlier cluster as well as the seven clusters grouped by K-means clustering.

Upon qualitative inspection, the K-means clusters' grouping and the overall UMAP cluster shape seem to depend on image contrast. Notably, features extracted by the ImageNet-trained ResNet, such as image cropping (see Figure 3.38), are less strongly grouped by the NIH-trained EfficientNet. The clusters' extremes show variations in image contrast (overexposed vs. low contrast), with the COVIDx-specific images located near the periphery. As expected, a network trained to extract features from CXR images appears to extract different features compared to a network trained on natural images. Image contrast seems to play a critical role in the classification task for the NIH ChestX-ray14 dataset, while dataset-specific characteristics such as image markers, background color, or cropping are not as strongly discriminating.

It is apparent that which features are extracted using a deep neural network are dependent on many factors (e.g., architecture, loss function, classification task, etc.). This is important to consider in many stages of developing a deep learning algorithm. For example, it is common to use ImageNet weights for pretraining a network, and for CXR applications it is also very common to use NIH pretraining. However, as shown in this section, the differences in the datasets as well as the overall classification task, cause a network to select very different features. This is also entangled with the differences in network architecture, as it is not guaranteed that different models extract the same features.<sup>55</sup> How different pretraining datasets affect performance in COVID-19 classification is explored further in Chapter 5 and further analysis of extracted features for different network architectures is studied in Chapter 6.

Deep learning feature extraction plays a vital role in comparing datasets by uncovering mean-





**Figure 3.42:** Unsupervised (top) and supervised (bottom) UMAP clustering of the ChestX-ray14 pretrained EfficientNet extracted features of the COVID-19 datasets labeled according to dataset source and K-means cluster centroids (K = 7).

ingful patterns and relationships within the data. Automatically identifying relevant features allows for a comprehensive comparison of different datasets, revealing similarities, differences, and biases that might not be apparent through manual inspection. Analyzing the datasets used for network training is crucial because it enables a better understanding of the model's strengths and weaknesses, ensuring that dataset-specific shortcuts or biases do not artificially inflate the model's performance. Additionally, such analysis can help identify potential challenges in generalization and guide the development of more robust models. By leveraging deep learning feature extraction and thorough dataset analysis, more accurate, reliable, and generalizable machine learning models can be developed, ultimately leading to improved performance in real-world applications.

#### 3.4.4 Outlier Detection

Deep learning feature extraction has also proven to be a powerful approach for outlier detection. Outliers, or anomalous data points, can offer valuable insights into the underlying patterns and characteristics of data, often uncovering potential issues or novel phenomena. As demonstrated in the previous section, CNNs are capable of automatically learning and extracting pertinent features from complex data, making them particularly well-suited for outlier detection tasks. Deep learning feature extraction methods are especially useful for identifying outliers in large, highdimensional datasets. In addition, by using embedding vectors as a lower-cost representation instead of full-resolution images, outlier detection methods can achieve greater computational efficiency.

In this section, the features extracted from the ImageNet-trained ResNet-121 and NIH-trained EfficientNet were used to analyze outlying images in each COVID-19 dataset. Identifying these outliers not only improves the dataset's quality but also contributes to a better understanding of the underlying processes and relationships within the data. To detect outlier images, an unsupervised machine learning algorithm called Isolation Forest<sup>120</sup> was employed. The algorithm is computationally efficient, making it suitable for large datasets and high-dimensional data, and it is less sensitive to the underlying data distribution compared to distance-based and density-based outlier detection methods, which rely on distance or density measures.

Isolation Forest is based on the concept of isolating anomalies or outliers rather than building profiles for normal data points. The algorithm can efficiently identify outliers in large, high-





**Figure 3.43:** An example of random partitioning in a 2D dataset of normally distributed points for a nonanomalous point (left) and for a point that's more likely to be an anomaly (right). It is apparent from the plots how anomalies require fewer random partitions to be isolated, compared to normal points. Image from https://en.wikipedia.org/wiki/Isolation\_forest.

dimensional datasets, and it is relatively robust to noise and different data distributions. The algorithm starts by randomly selecting a feature and a split value between the minimum and maximum values of that feature. It then splits the dataset based on the chosen feature and split value. This process is recursively applied to the partitions created by the split until all data points are isolated or a specified tree depth is reached. Multiple isolation trees are created as an ensemble, and each tree is constructed using a random subsample of the dataset. Next, the average path length of a data point in the ensemble of isolation trees is calculated. The path length is the number of edges from the root node to the terminal node containing the data point. Anomalies or outliers are expected to have shorter path lengths, as they can be isolated more quickly than normal data points. Last, an anomaly score is calculated for each data point based on its average path length across the ensemble of isolation trees. The score is a value between 0 and 1, where higher scores indicate a higher likelihood of the data point being an outlier. An example of this process is shown in Figure 3.43.

The subsequent figures present 180 identified outlier images for each COVID-19 dataset, based on features extracted from both the ImageNet-trained ResNet-121 and the NIH-trained EfficientNet models. A prevalent trend observed is that the ImageNet model highlights images with more distinct features, such as medical devices, spinal implants, and varying orientations/views, as outliers, whereas the NIH model primarily focuses on contrast. This observation aligns with the UMAP grouping from the previous section, where contrast was the principal grouping feature. One plausible explanation is that the model trained on the NIH dataset, due to its extensive patient cohort, has likely learned that implants are a relatively rare but acceptable feature in CXR images. Conversely, in natural images from ImageNet, sharp contrast is more critical for object definition. As a result, the outlier images differ slightly across the datasets; for example, the COVIDx dataset contains numerous unique outlier images, including arrows, markers, and pediatric cases.

Examining outlying cases is essential for comparing datasets, evaluating overall quality, and uncovering hidden patterns and inconsistencies that can significantly impact data interpretation and understanding. Outliers, or anomalous data points, often result from measurement errors, data entry errors, or genuine variations in the underlying process. Identifying and addressing these anomalies can lead to improved data quality, more accurate models, and better decision-making. Outlier detection is a crucial aspect of dataset analysis, empowering researchers and practitioners to enhance data quality, gain valuable insights, and develop more effective models to tackle real-world challenges.



**HF ImageNet-Trained Isolation Forest Outliers** 



**HF NIH-Trained Isolation Forest Outliers** 



HF Temp ImageNet-Trained Isolation Forest Outliers



**HF Temp NIH-Trained Isolation Forest Outliers**


**BIMCV ImageNet-Trained Isolation Forest Outliers** 



**BIMCV NIH-Trained Isolation Forest Outliers** 



**UW ImageNet-Trained Isolation Forest Outliers** 



**UW NIH-Trained Isolation Forest Outliers** 



MIDRC ImageNet-Trained Isolation Forest Outliers



**MIDRC NIH-Trained Isolation Forest Outliers** 



**COVIDx ImageNet-Trained Isolation Forest Outliers** 



**COVIDx NIH-Trained Isolation Forest Outliers** 

## 3.5 Discussion

The significance of high-quality data in the field of artificial intelligence cannot be overstated. The success of AI systems and models is fundamentally dependent on not only the quantity and diversity of the datasets used for training and validation, but also the quality. Dataset curation and analysis are vital processes that ensure the reliability and representativeness of the data, enabling the development of more accurate and robust AI models.

One essential aspect of dataset curation is the identification and mitigation of biases present in the data. Biases can stem from various sources, such as patient population characteristics or image acquisition characteristics, and can adversely impact the performance and fairness of AI systems. Rigorous dataset analysis enables the detection of potential biases, allowing researchers and practitioners to address these issues, ideally before deploying AI models in real-world applications. By carefully curating datasets and thoroughly analyzing them for biases, the AI community can develop models that better generalize to diverse scenarios and provide clinically useful AI solutions. Moreover, this attention to data quality and bias reduction fosters trust in AI systems and encourages their adoption across various domains.

The importance of data in AI highlights the need for meticulous dataset curation and analysis to ensure data quality, diversity, and fairness. High-quality data is just as crucial, if not more so, than model architecture and training strategies. As demonstrated in the following chapters, "good data" takes precedence over "big data" when it comes to achieving optimal outcomes. By addressing potential biases and fostering the development of more accurate and robust AI models, AI can unlock its full potential to address complex real-world challenges and create more equitable and inclusive solutions. However, even in well-curated datasets, there may be subtle biases present. The upcoming chapter emphasizes how these hidden biases can act as shortcuts, leading to inflated model performance while ultimately undermining their clinical viability.

## Chapter 4

# Uncovering Hidden Patterns: Analyzing Shortcut Learning in COVID-19 Chest X-ray Datasets

Wilhelm von Osten was a math teacher who lived in Germany in the late 1800s. Caught up in some of the pseudoscience of those days, Osten believed that animals were far more intelligent than what humans gave them credit. To prove this, he began to tutor his Arab stallion, named Hans, in arithmetic. First, Hans learned to tap his foot out to numbers written on a blackboard. Soon Hans could tap out the correct answers to simple arithmetic problems. Indeed Hans was the most clever animal Osten had ever encountered.

Osten began showcasing Hans to crowds all over Germany, proclaiming that "Clever Hans" was a horse with true mathematical proficiency. Crowds would gather as Hans would correctly answer: "What is the square root of sixteen?" with four hoof stomps. Hans could stomp out the time of day from looking at a clock. Hans had even learned how to spell people's names where one tap was "A," two "B," and so on. While Hans would make some mistakes, it has been estimated that his reasoning skills were equivalent to an adolescent human.

Naturally, there were many skeptics who believed that Osten was somehow signaling the horse because no animal, let alone a horse, could be so clever. As a result, the German board of education requested an independent test of Hans' abilities without Osten present, to which Osten promptly agreed. The testing panel gathered to test Hans consisted of two zoologists, a psychologist, a horse trainer, school teachers, and even a circus animal handler. Amazingly, this panel concluded that there was no trickery to Hans' abilities; Hans apparently was that clever.



**Figure 4.1:** Clever Hans, a horse who demonstrated mathematical genius which was later discovered to be giving the right answers by watching the reactions of the people who were watching him.

Many could not accept this result, claiming there must be some explanation, some even more clever way in which Osten was manipulating the horse. The investigation was taken up by Oskar Pfungst, a psychologist who had some theories on how to address this mystery. Pfungst conducted his experiments much more rigorously than the panel before, housing the horse in a large tent to remove outside factors or visual stimuli. He developed a large list of questions that differed in complexity and reasoning ability. Pfungst observed that Hans responded correctly when Osten asked questions, but Hans *also* performed well with other controlled questioners. So Pfungst started changing testing conditions and observed that Hans was more error-prone when the questioners stood further away. Hans also could not answer the question if the questioner did not know the answer themselves. Finally, Hans completely failed when he could not see the questioner. These observations together lead Pfungst to believe that Hans' abilities depended on whether Hans had an up-close view of the questioner who knew the correct answer.

Pfungst then focused on the interactions between the questioner and Hans. He noticed that as Hans approached the correct answer, the questioner involuntarily demonstrated anticipation in their body language through breathing, posture, and facial expressions. Once Hans reached the correct answer, the questioner would change their body language cues slightly as the build of anticipation passed. Pfungst also noticed that the questioner would not exhibit these cues when they did not know the correct answer. It became obvious that Hans was not a math genius, but he still was very clever. Hans had learned to recognize the subtle cues given by the questioners through body language and facial expressions.

Today, the term "Clever Hans Effect"<sup>121,122</sup> is used to describe the influence of a questioner's subtle and unintentional cues upon their subjects, in both humans and in animals. This led to the modern standard of the double-blind method to prevent inherent biases from influencing the outcomes of experiments.

What does a horse that notices a slight raise of the eyebrows have to do with convolutional neural networks and COVID-19 classification? More than it might appear at first. Hans performed a task: answering arithmetic questions correctly, with astonishing performance. However, Hans did not understand mathematics, so when the situations changed, he failed to reproduce his amazing results. Hans had learned a shortcut, a way to get the correct answer, but using different logical rules than what was needed to generally perform the task. In fact, AI is a lot like Hans in some situations. For example, a trained CNN was able to diagnose pneumonia in CXR with near-perfect accuracy, far better than even the best human radiologists; however, it was discovered that the model was exploiting the fact that all the positive images contained a hospital-specific token in the image.<sup>83</sup> In this case, "Clever AI" was just being perceptive, much like Hans, and failed to learn the true characteristics of pneumonia in CXR. So when tested on new data which did not have these markers, the model was left without a clue (and was about as useful to a radiologist as a horse).

## 4.1 The COVID-19 Pandemic: A Perfect Storm for Shortcut Learning

Shortcut learning<sup>80</sup> in deep learning refers to a phenomenon where a model learns to exploit certain patterns or biases in the training data that allow it to perform well on the training set and validation set, but without truly understanding or generalizing the underlying concepts. In other words, the model learns to take "shortcuts" by focusing on superficial or spurious correlations, which can lead to poor performance when encountering novel situations or testing data that deviate from the biases present in the training data. Shortcut learning can occur for a variety of reasons, such



**Figure 4.2:** Much like Hans, "Clever AI" has learned to detect a metal token that radiology technicians place on the patient in the corner of the image instead of the true imaging features of pneumonia. Image from Zech et al., "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study." *PLoS medicine*, 2018.

as insufficiently diverse training data, poorly designed networks or training, or overfitting due to excessive model capacity. This severely limits the generalizability, and thus the practical usefulness, of the network.

Shortcut learning is not a new concept and is related to many well-studied concepts such as covariate shift, <sup>123</sup> anti-causal learning, <sup>124</sup> and dataset bias. <sup>125</sup> However, the COVID-19 pandemic brought shortcut learning to the forefront of the AI community's attention. Due to the rapid onset of the COVID-19 pandemic, it was believed that chest CT scans and radiography could potentially serve as a primary method for diagnosing COVID-19, particularly during the early stages of the pandemic when PCR testing was limited. With such an ideal scenario for AI solutions, the community jumped on this opportunity leading to an explosion in the number of publications on AI-related topics concerning COVID-19. Search results on Scopus using TITLE-ABS-KEY-AUTH("Deep learning" OR "Artificial intelligence" OR "AI" AND "COVID" AND "Chest X-Ray") show 185 papers in 2020, 697 papers in 2021, 774 papers in 2022, and 135 papers to date in 2023. However, despite the efforts of thousands of researchers to utilize AI to address the COVID-19 crisis, for most deep learning models, "...the predictive performance was weak in real-world clinical settings,"<sup>84</sup> "...diagnosing COVID-19...entirely unusable,"<sup>85</sup> and "...to be of little practical use."<sup>126</sup>

In many ways, the COVID-19 pandemic created the perfect storm for AI to fail, with several factors contributing to its shortcomings. In the years leading up to the pandemic, AI (particularly deep learning) experienced a meteoric ascent, marked by numerous impressive achievements

that fueled excitement and bolstered confidence in the technology. Additionally, the barriers to entry in AI have significantly reduced, thanks to modern libraries that simplified neural network implementation, online cloud GPU computing, and public dataset services; and with the urgent goal to save lives, many individuals were inspired to contribute to this important cause. This democratization of AI enabled anyone interested to delve into the world of deep learning. However, while increased accessibility is generally a positive development, it can create problems when individuals attempt to solve complex problems without adequate domain knowledge. This lack of expertise can lead to numerous oversights or incorrect implementations, highlighting the importance of domain knowledge in tackling real-world challenges.

However, perhaps the biggest contributing factor lay not in the deep neural networks themselves but rather in the data used to train them. While there were numerous publicly available chest X-ray datasets (see Chapter 3), there was a scarcity of COVID-19 CXR images, particularly at the beginning of the pandemic. This issue of data imbalance, stemming from disease prevalence, is a common challenge in medical imaging as the limited data exacerbate overfitting problems in DNNs. To address this, numerous initiatives were quickly launched to gather as many COVID-19 images as possible, with the goal of creating a comprehensive public dataset for developing more effective DNNs. A notable example is the COVIDx dataset,<sup>104</sup> where COVID-19 CXRs were collected from various public sources such as research articles, websites, and tweets.<sup>105</sup> These images, paired with negative images from pre-pandemic datasets, unfortunately, provided the ideal conditions for shortcut learning to occur.

Non-relevant features can become shortcuts when they are correlated with image labels. For instance, if every COVID-19 positive image contains text in the upper right corner, while COVID-19 negative images do not, the presence or absence of text becomes entangled with the COVID-19 positive/negative distinction. Deep neural networks may choose these shortcuts because, unlike humans, who can employ abstract reasoning to focus on disease features and ignore irrelevant text, DNNs lack such awareness. As discussed in Chapter 2, CNNs extract features from images and use them to predict the image labels. The extraction and utilization of features are guided by the loss function and backpropagation of gradients.

By employing gradient descent, users enforce network minimization of the loss function, aiming to find the lowest minimum possible. Given these requirements, it becomes obvious why shortcuts

are used: the network will use *the most discriminating features* as these features provide the lowest loss value. Determining if a patient has COVID-19 pneumonia versus another lung abnormality in a CXR is a highly non-trivial task, but checking the right corner for text is something one could train a horse to do (albeit a clever one). Therefore, it is hard to fault the network for exploiting this shortcut, as it is only fulfilling its role in minimizing the loss.

For deep learning solutions to be clinically viable, they must learn generalizable features of the data. However, shortcut learning hampers this process, contributing to the struggle of AI in medical imaging to achieve the success observed in other fields. While standard image classification tasks in computer vision require models to learn general object appearances, medical imaging often necessitates the detection of minor structural differences, such as local changes in image contrast and intensity, which are critical for disease classification. Consequently, DNNs may become highly sensitive to these image details. It is thus crucial to understand and detect shortcut learning, and in this chapter, potential shortcut features in COVID-19 CXR datasets are explored, along with a general framework for detecting these features in a dataset. Identifying shortcut learning helps improve dataset quality and fosters the development of more generalizable networks.

## 4.2 Shortcut Features in COVID-19 CXR Datasets

Much like in the story of Hans, shortcuts in AI can be subtle and difficult to pick out beforehand. While features such as image markers in a CXR are obvious once pointed out, humans suffer from attentional and expectation bias, focusing on results that intuitively make sense or conform to expected outcomes. This, in conjunction with the staggering size of most datasets, makes the identification of bias in datasets nearly impossible without thorough analysis.

#### 4.2.1 Dataset Metadata Shortcuts

To demonstrate that deep learning models can use various unexpected features for shortcut learning in COVID-19 CXR classification, deliberately biased training datasets containing various shortcuts were created. The shortcuts were determined based on the metadata available for the COVID-19 datasets (see Chapter 3) and included patient age (>70/<50 years), sex (male/female), and race (white/black), as well as CXR view position (AP/PA), modality (DX/CR), and vendor



**Figure 4.3:** To identify potential shortcut features in COVID-19 CXR classification, biased training datasets containing various shortcuts were created. For example, 1,500 COVID-19 **positive male patients** and 1,500 COVID-19 **negative female patients** were randomly selected to form a shortcut training dataset where patient sex is completely correlated with the COVID-19 label. To evaluate if a model trained on the shortcut dataset learned the shortcut features instead of the COVID-19 features, (1) high accuracy on an internal dataset with the same shortcut correlation as the training dataset and (2) low accuracy on on external dataset where patient sex is not correlated with the COVID-19 label is expected.

(Carestream/Konica Minolta). A shortcut dataset was created by selecting 1,500 images with COVID-19 positive and one shortcut variable (e.g., COVID-19 positive male patients) and 1,500 images with COVID-19 negative and another grouping (e.g., COVID-19 negative female patients). The resulting training datasets contained 3,000 images, where the shortcut feature was entirely correlated with the COVID-19 label.

To evaluate if a model trained on the shortcut dataset learned the shortcut features instead of the COVID-19 features, it was tested on two datasets:

- The first dataset is a hold-out test set where the shortcut features correlate with the COVID-19
  labels in the same manner as the training dataset. If the model learned the shortcut, a high
  accuracy (i.e., the area under the receiver operating characteristic curve (AUC) close to 1) is
  expected.
- 2. The second test dataset assesses the generalizability of the shortcut model. This dataset consists of COVID-19 positive/negative images from an external dataset (i.e., a dataset not used to create the training dataset) and does not contain the shortcut introduced in the training dataset. If the model learned the shortcut instead of the COVID-19 features, a low accuracy

**Table 4.1:** A comparison of internal test performance (AUC and 95% confidence interval) and external test performance was conducted for models trained on biased training datasets. These biased datasets featured a complete correlation between the COVID-19 label and the shortcut variable. The baseline model had an internal AUC of 0.781, roughly equivalent to a human radiologist's performance, and exhibited comparable performance on the external dataset. For the shortcut-trained models, a high internal dataset AUC greater than 0.93 indicates that the model used the shortcut to enhance performance. This conclusion is further supported by the significant drop in performance on a test set where the shortcut is absent.

	Training Dataset	External Test Dataset	Internal Test AUC	External Test AUC
Baseline Model	HF	BIMCV	0.781	0.765
			[0.752, 0.809]	[0.735, 0.794]
Patient Age	ЦЕ	MIDRC	0.983	0.499
(>70/<50)	111		[0.977, 0.989]	[0.464, 0.535]
Patient Sex	HE	MIDRC	0.992	0.557
(Male/Female)	111		[0.989, 0.996]	[0.521, 0.593]
Patient Race	MIDPC	BIMCV	0.933	0.641
(White/Black)	WIIDIKC		[0.918, 0.947]	[0.607, 0.675]
View Position	SP	MIDRC	0.991	0.532
(AP/PA)	51	WIIDKC	[0.985, 0.998]	[0.496, 0.568]
Imaging Modality	PIMCV	MIDRC	0.951	0.673
(DX/CR)	DIVICV		[0.939, 0.964]	[0.640, 0.706]
Imaging Vendor	НЕ	MIDRC	0.987	0.536
(Carestream/Konica Minolta)	111		[0.982, 0.993]	[0.500, 0.572]

(i.e., an AUC close to 0.5, equivalent to random guessing) is expected.

This process is outlined in Figure 4.3 for the shortcut of patient sex. To establish baseline performance and evaluation of generalizability, 1,500 COVID-19 positive and 1,5000 COVID-19 negative images were randomly selected from the HF dataset to train a shortcut-free model. This training dataset was confirmed to have no metadata variable correlation with the COVID-19 label, and since the data originate from the same location, it is considered to be comparatively bias-free. A DenseNet-121<sup>127</sup> network architecture was used with NIH pretraining. Details of specific model training parameters are provided in Appendix A.

A comparison of internal test performance (AUC) and external test performance was conducted for models trained on biased training datasets and is shown in Table 4.1. The baseline model had an internal AUC of 0.78, roughly equivalent to a human radiologist's performance, and exhibited comparable performance on the external dataset.

For the shortcut-trained models, a high internal dataset AUC greater than 0.93 in every case suggests that the model leveraged the shortcut to improve performance. This conclusion is further



**Figure 4.4:** Randomly selected images corresponding to the training datasets described in Table 4.1. Apart from patient sex, primarily due to differences in breast tissue, none of the shortcut variables exhibit obvious discriminating features upon inspection.

supported by the significant drop in performance on a test set where the shortcut is absent. Figure 4.4 displays 16 randomly selected images corresponding to the training datasets described in Table 4.1. Apart from patient sex (primarily due to differences in breast tissue), none of the shortcut variables exhibit obvious discriminating features upon inspection. However, somewhat unexpectedly, the results indicate that all variables available from the dataset metadata can become powerful shortcuts when they are correlated with the COVID-19 label.

Neural networks do not "see" the same way a human observer does, so it is not unreasonable for a network to learn AP vs. PA based on thoracic and cardiac width comparisons, for example. Some of the other variables have less straightforward explanations. For instance, there is no biological explanation for why a network could detect differences in race, a result observed in other studies as well.<sup>128</sup>

#### 4.2.2 Dataset Source Shortcuts

The datasets constructed in the previous experiment represent the worst-case scenario and may not accurately reflect real-life datasets. For example, it is unlikely that an entire patient cohort would consist solely of males, patients above 70 years of age, or individuals of Caucasian descent. Ensuring adequate data diversity and conscientious dataset curation and analysis can eliminate



**Figure 4.5:** When using data from different sources as the classification label (left), a network will simply learn the differences between the datasets and fail to generalize to outside data. By removing the correlation between the dataset source and classification label (right), the network learned more generalizable features and relied less on dataset-specific shortcuts.

these biases if they exist, provided that metadata is available for analysis.

However, a more realistic scenario in deep learning, particularly in medical imaging where data is relatively scarce, involves combining distinct but related datasets. For instance, many studies combined COVID-19-positive CXR images with pre-pandemic CXRs, reasoning that pre-pandemic images would have no features of COVID-19 and therefore no mislabeling issues, as well as being readily available. While there might be diversity within each individual dataset, the dataset itself becomes correlated with the label, allowing the network to exploit spurious differences between the datasets. Essentially, the network memorizes the dataset source instead of disease features.

To demonstrate that a network can use the dataset source as a shortcut, a training dataset was created using 1,500 COVID-19 positive cases from the BIMCV dataset and 1,500 COVID-19 negative cases from the HF dataset. Both BIMCV and HF datasets comprise a diverse range of patient ages, races, and imaging vendors and maintain balance in terms of patient sex, imaging modality, and view position. In this case, the only evident correlation between the images and the label is the dataset source.

A DenseNet model was trained on this source-correlated dataset and tested on a hold-out test set



**Figure 4.6:** Even with 70% of the image masked out, when using data from different sources as the classification label, a network will primarily learn the differences between the datasets and fail to generalize to outside data.

(BIMCV+/HF-) and an external test set (MIDRC) to evaluate if the network learned generalizable COVID-19 features or if it learned dataset-specific differences. The network achieved an AUC of 0.990 (95% CI: [0.985, 0.994]) on the hold-out test set and an AUC of 0.508 (95% CI: [0.483, 0.534]) on the MIDRC test set (see Figure 4.5). In fact, this shortcut is so strong that only a small percentage of the image is needed to completely discriminate the dataset sources. To demonstrate this, a DenseNet model was trained on the BIMCV+/HF- dataset where 70% of the image area was masked, leaving only the border remaining (see Figure 4.6). The network achieved an AUC of 0.983 (95% CI: [0.976, 0.990]) on the hold-out test set and an AUC of 0.520 (95% CI: [0.484, 0.556]) on the MIDRC test set. This confirms that when using data from different sources as the classification label, a network will primarily learn the differences between the datasets and fail to generalize to outside data. Note that even if the data is diverse within each label, shortcut learning occurs because the dataset source is correlated with the classification label.

To further demonstrate this point, a new model with 1,000 BIMCV and 1,000 HF COVID-19 positive cases and 1,000 BIMCV and 1,000 HF COVID-19 negative cases was trained. In this scenario,



**Figure 4.7:** Efforts to mitigate shortcut learning include methods to harmonize data between different datasets to remove biases. Histogram equalization addresses image contrast and window/leveling differences, and segmentation can isolate relevant features.

the classification labels are no longer correlated with the dataset source but are evenly balanced between the two different datasets. The network achieved an AUC of 0.794 (95% CI: [0.766, 0.821]) on the hold-out test set and an AUC of 0.688 (95% CI: [0.665, 0.711]) on the MIDRC test set (see Figure 4.5). By removing the correlation between the dataset source and classification label, the internal AUC dropped to the baseline model performance, and the external AUC increased by nearly 0.2, suggesting that the network learned more generalizable features and relied less on dataset-specific shortcuts. Note that there still exists a substantial generalization gap, and methods to reduce this gap even further are explored in Chapter 5; the focus here is to highlight shortcut learning.

These findings indicate that the effects of shortcut learning can be somewhat mitigated by creating balanced datasets that prevent correlations with classification labels. However, this approach is not ideal, particularly in the field of medical imaging where data is often limited and combining datasets is frequently the only viable option. Some research studies have proposed techniques such as lung tissue segmentation and histogram equalization as ways to harmonize data across different datasets.<sup>107,129</sup> The rationale behind this is that COVID-19 infections predominantly impact the lungs, making lung tissue the primary source of relevant classification features. Segmentation can effectively remove extraneous features, thereby reducing biases across datasets. Furthermore, histogram equalization can standardize image contrast and window/leveling differences.

To evaluate the efficacy of histogram equalization and segmentation in mitigating shortcut learning, the BIMCV+/HF- dataset underwent preprocessing, applying histogram equalization to



**Figure 4.8:** In comparison to the model without segmentation and histogram equalization (left), the model trained on equalized and segmented images still exhibits considerable shortcut learning (right).

each image. Subsequently, a CNN was trained for lung segmentation and employed to generate lung masks for all images. Figure 4.7 illustrates an example of this process. Finally, a new DenseNet model was trained on the equalized and segmented images and tested on both the hold-out test set (BIMCV+/HF-) and an external test set (MIDRC) after applying the same histogram equalization and segmentation processing. The network achieved an AUC of 0.942 (95% CI: [0.928, 0.955]) on the hold-out test set and an AUC of 0.549 (95% CI: [0.513, 0.584]) on the MIDRC test set (see Figure 4.8). Compared to the model without segmentation and histogram equalization, this model still exhibits considerable shortcut learning. Thus, even with these preprocessing techniques, dataset-specific features within the lungs continue to serve as potent shortcuts.

The outcome of this study is rather unexpected. With extraneous features eliminated by segmentation and contrast equalized, one might wonder what the shortcut features could be in this instance. Reflecting on the story of Hans, the most apparent explanation—Wilhelm von Osten signaling the horse—was investigated by a testing panel, only to find that the actual shortcut was far more subtle. In the case of the BIMCV+/HF- dataset, the most evident shortcuts were addressed through segmentation and histogram equalization; however, the underlying shortcut appears to be less conspicuous. Given that the input image is a segmented lung, segmentation itself may serve

as a shortcut. To investigate whether lung shape correlates with the dataset source, the binary masks of the BIMCV+/HF- dataset were used as training data in place of the CXR images. Another DenseNet model was trained on the binary lung masks and tested on both the hold-out test set (BIMCV+/HF-) and external test set (MIDRC) masks. The network achieved an AUC of 0.703 (95% CI: [0.671, 0.735]) on the hold-out test set and an AUC of 0.505 (95% CI: [0.469, 0.541]) on the MIDRC test set. Although the AUC of 0.7 suggests a correlation with lung shape/size, this feature alone is insufficient to fully account for the near-perfect accuracy observed in the BIMCV+/HF- trainings shown previously (see Figure 4.8).

Since the segmented training dataset contains only lung tissues and lung shape is not the most significant shortcut, the remaining shortcuts might arise from the intrinsic characteristics of the CXRs, such as image contrast and sharpness. In COVID-19 classification tasks involving chest CXRs, image contrast and sharpness might differ among hospitals due to the use of various imaging systems, different generations of X-ray imaging equipment (e.g., DX vs. CR), a range of hardware components (including flat-panel detectors, X-ray tubes, and anti-scatter grids), supplemental image post-processing methods employed by vendors to produce the final digital images for radiologists' interpretation, and distinct imaging protocols (such as tube potential and exposure levels) utilized by technologists in clinics for creating X-ray radiographs. All these factors affect the digital representation of the acquired image data, leading to variations in image contrast and sharpness, as summarized in Figure 4.9.

## 4.3 Demonstration of Intrinsic Shortcut Features in CXR Imaging

Addressing extrinsic shortcut features of a dataset, such as patient age, sex, race, image markers, and so on, can be achieved through proper dataset curation, balancing, and image preprocessing. However, intrinsic shortcut features like image sharpness or contrast are considerably more difficult to identify and mitigate. The reason for this lies in the fact that the desired image features also manifest as image contrast (pixel intensity variations) and spatial correlations, just like the contrast-and sharpness-related shortcuts. This entanglement of the desired image and shortcut features makes studying contrast and sharpness-related shortcuts exceedingly challenging. In this work, these image contrast and sharpness-related shortcuts are referred to as intrinsic shortcuts due to



**Figure 4.9:** For chest CXRs employed in COVID-19 classification tasks, image contrast and sharpness may vary across hospitals due to a variety of reasons. These inherent characteristics of the CXRs can serve as shortcuts.

their entanglement with desired image features.

In contrast to many conspicuous extrinsic features (e.g., patient sex determined by breast tissue or hospital-specific markers) that can be removed through cropping or segmentation, contrast and resolution are global features not easily eliminated or harmonized. While processing techniques such as histogram equalization or blurring/sharpening can help reduce variations between datasets, some differences will persist. This is because these techniques work on individual images, adjusting their pixel intensities based on unique histograms. Consequently, processing alone does not guarantee that transformed images will have the same contrast or resolution characteristics. This is particularly true for datasets containing images from multiple sites, vendors, modalities, and so on.

To demonstrate that contrast and sharpness can serve as shortcut features, a reference training dataset consisting of CXRs from 3,000 COVID-19 positive and 3,000 COVID-19 negative patients randomly selected from the HF dataset was created. This training dataset was confirmed to have no metadata variable correlation with the COVID-19 label, and since the data originates from the same general location, it is considered relatively bias-free. Intrinsic shortcut datasets were generated by adjusting the image sharpness and contrast in the COVID-19 positive images. To compare how a network learns global intrinsic features versus local extrinsic features, a third dataset with an extrinsic shortcut feature, a simulated lead marker in the COVID-19 positive images, was created. An overview of applying these shortcuts to the images is shown in Figure 4.10. Shortcut training



**Figure 4.10:** Methods for adjusting the image sharpness and contrast to the COVID-19 positive images. Contrast was adjusted using parameter c = 0.02 and sharpness was adjusted using parameter  $\alpha = 1.1$ .

datasets were created as follows (using the same 3,000+/3,000- cases as the reference dataset to ensure the same COVID-19 features are present in the shortcut datasets):

- 1. **Image Sharpness Shortcut**: All positive cases have image sharpness adjusted by a factor  $\alpha$  = 1.1, and negative cases remain unchanged.
- Image Contrast Shortcut: All positive cases have image contrast adjusted by a factor *c* = 0.02, negative cases remain unchanged.
- 3. **Image Marker Shortcut**: All positive cases have a randomly inserted marker (pixels assigned value 255) in a random location in the shoulder region (1 px), and negative cases remain unchanged.

It is important to introduce realistic shortcuts that mimic the variations in images across different datasets, ensuring no drastic changes are observed in the overall image appearance. As shown in Figure 4.11, the changes described above are nearly imperceptible.



Figure 4.11: Methods for adjusting the image sharpness and contrast to the COVID-19 positive images.

To evaluate if the model learned COVID-19 relevant features or used the simulated shortcut features, multiple test sets were created:

- 1. **COVID-19 Test Set**: To evaluate the model's baseline ability to learn COVID-19 features, the BIMCV dataset, which consists of 4,169 COVID-19 positive and 5,050 COVID-19 negative cases, was used as an external test set.
- 2. Sharpness Test Set: To assess whether the model learned the sharpness shortcut, 10,000 "normal findings" cases from the MIMIC dataset (released pre-pandemic and thus without COVID-19 cases) were randomly selected. The image sharpness of 50% of the cases was adjusted by a factor of 1.1 and assigned a class label "1," corresponding to COVID-19 positive. The remaining cases were left unchanged and assigned a class label of "0," corresponding to COVID-19 negative.
- 3. **Contrast Test Set**: To determine if the model learned the contrast shortcut, the same 10,000 "normal findings" cases from the MIMIC dataset were used, with 50% of the cases having image contrast adjusted by a factor of 0.02 and assigned a class label of "1." The remaining cases were left unchanged and assigned a class label of "0."

4. Marker Test Set: To assess whether the model learned the simulated marker shortcut, the same 10,000 "normal findings" cases from the MIMIC dataset were used, with 50% of the cases having a marker inserted (pixels assigned a value of 255) in a random location in the shoulder region and assigned a class label of "1." The remaining cases were left unchanged and assigned a class label of "0."

Finally, to explore the features learned when multiple shortcuts are present, various combinations of the sharpness, contrast, and marker datasets were created. For example, the COVID-19 positive cases in the training and test datasets could have both sharpness and contrast adjustments applied, or sharpness and contrast adjustments combined with the insertion of a marker, and so on. This allows for a more comprehensive understanding of how the model learns and responds to the presence of multiple shortcut features in the dataset.

A comparison of model performance (AUC and 95% confidence interval) for the reference model without shortcuts, simulated sharpness, contrast, and marker shortcut models, as well as their combinations, is presented in Table 4.2. Deep neural network models can effectively learn these nearly imperceptible shortcut features, as evidenced by their near-perfect AUC on the corresponding shortcut test set. When these features correlate with COVID-19 labels, they can become shortcuts for deep neural network models, causing the true disease features to be bypassed during the training process. This is illustrated by the significant drop in AUC for the COVID-19 test set when shortcut features are present.

Another observation is that global features, such as image sharpness and contrast, are more potent shortcuts than local features like the simulated marker for the COVID-19 classification task. This is demonstrated by the marker model, which, despite learning to detect the marker perfectly (AUC = 1), only experienced a moderate drop in COVID-19 test performance. In other words, this shortcut feature can coexist with the COVID-19 features. In contrast, both the contrast and sharpness shortcuts bypass COVID-19 features almost entirely when present (AUC close to 0.5). However, when both contrast and sharpness shortcuts are present, the marker shortcut is disregarded (AUC = 0.6 on the marker test).

These findings reveal a concerning conclusion. In the case of CXR, different imaging system vendors utilize distinct hardware, such as detectors, X-ray filtration, and anti-scatter grids, as well as

**Table 4.2:** A comparison of model performance (AUC and 95% confidence interval) for the reference shortcutfree model, simulated sharpness, contrast, and marker models as well as their combinations. These nearly imperceptible shortcut features can be effectively learned by deep neural network models demonstrated by their nearly perfect AUC on the corresponding shortcut test set. When these features are correlated with the COVID-19 labels, they can become shortcuts for deep neural network models, such that true disease features are bypassed in the training process, shown by the significant drop in AUC for the COVID-19 test set when shortcut features are present. Global features, such as image sharpness and contrast, are stronger shortcuts than local features, such as the simulated marker for the COVID-19 classification task.

	COVID-19 Test AUC	Sharpness Test AUC	Contrast Test AUC	Marker Test AUC
Potoronco Model	0.780	0.450	0.568	0.498
Reference would	[0.770, 0.790]	[0.438, 0.461]	[0.557, 0.579]	[0.487, 0.509]
Sharpness Model	0.527	0.997	0.582	0.492
	[0.514, 0.540]	[0.996, 0.999]	[0.571, 0.593]	[0.481, 0.503]
Contrast Model	0.599	0.517	0.996	0.499
	[0.586, 0.612]	[0.506, 0.529]	[0.995, 0.997]	[0.488, 0.511]
Marker Model	0.721	0.472	0.527	1.000
	[0.710, 0.732]	[0.460, 0.483]	[0.516, 0.539]	[0.999, 1.000]
Sharpness & Contrast	0.495	0.969	0.941	0.486
Model	[0.482, 0.508]	[0.966, 0.972]	[0.937, 0.946]	[0.475, 0.498]
Sharpness & Marker	0.536	0.995	0.995 0.564	
Model	[0.523, 0.549]	[0.993, 0.996]	[0.553, 0.576]	[0.890, 0.903]
Contrast & Marker	0.602	0.509	0.986	0.870
Model	[0.589, 0.615]	[0.497, 0.520]	[0.984, 0.988]	[0.863, 0.877]
Sharpness & Contrast	0.488	0.976	0.949	0.604
& Marker Model	[0.475, 0.501]	[0.973, 0.979]	[0.945, 0.953]	[0.593, 0.615]

proprietary image processing methods. Even within the same vendor and system model, different hospitals might have slightly varying preferences for data acquisition settings, lead marker size and shape, or image compression methods. Any of these factors can lead to subtle changes in image contrast and sharpness. While these variations may be imperceptible to the human eye and irrelevant to radiologists, deep neural network models can detect them with astonishing sensitivity.

For CXR classification problems, if positive and negative cases are collected from two different sources (such as vendors, hospitals, or even different departments within the same hospital), these image features become naturally correlated with the disease labels, creating shortcuts in model training. As a result, the common practice of collecting data from diverse sources in AI model training, which is intended to improve generalization, may, in fact, achieve the opposite effect, as demonstrated in these findings.

Developing an effective strategy to mitigate contrast- and sharpness-related shortcuts requires



**Figure 4.12:** Grad-CAM activation maps were generated for several examples from the reference, sharpness, contrast, and marker trained models. The reference model highlights features in the lung, which aligns with the intuition that COVID-19 features appear in this area. The marker model is also easily understood, as the model focuses on the simulated marker for classification. However, the sharpness and contrast shortcuts do not offer any clear conclusions that can be drawn from these visual maps.

the establishment of reliable methods for detecting their existence and severity in a curated dataset. This task is particularly challenging when these shortcuts are not visible to the naked eye. Although interesting post hoc model interpretability methods, such as class activation maps<sup>77</sup> and expected gradients, <sup>130</sup> have been developed to identify relevant image features primarily used by trained deep learning models for prediction, as shown in Chapter 6, these methods are unable to detect the intrinsic shortcuts in a curated training dataset before model training is conducted. Moreover, research suggests that these methods may not be helpful in diagnosing the poor generalization performance of a model.<sup>131</sup>

To illustrate this point, Grad-CAM activation maps were generated for several examples from the reference, sharpness, contrast, and marker-trained models, as shown in Figure 4.12. The reference model highlights features in the lung, which aligns with the intuition that COVID-19 features appear in this area. The marker model is also easily understood, as the model focuses on the simulated marker for classification. However, the sharpness and contrast shortcuts do not offer any clear conclusions that can be drawn from these visual maps, demonstrating the limitations of such methods for detecting intrinsic shortcuts. Model visualization and interpretability are extensively discussed in Chapter 6.

It might be argued that rather than identifying shortcuts, a more comprehensive solution, such as segmentation, histogram equalization, or other preprocessing procedures, should be employed. However, as demonstrated earlier (refer to Figure 4.8) and supported by other studies,<sup>88</sup> these strategies do not fully succeed in mitigating shortcut learning. This emphasizes the importance of understanding and addressing the underlying causes of shortcut learning to develop more effective solutions.

## 4.4 Detection of Intrinsic Shortcut Features in CXR Imaging

The previous sections have demonstrated that shortcut learning is easily accomplished when features become correlated with the classification label, making it easy for models to exploit these relationships. While careful dataset collection and curation procedures can help avoid obvious biases, combining data from different sources, which is often unavoidable in medical imaging, can introduce subtle or even imperceptible differences between sources that serve as shortcuts. To address these shortcut features, they must first be identified. Therefore, this section presents a method for detecting contrast- and sharpness-related shortcuts in CXR COVID-19 datasets.

The detection framework can be summarized in the following steps:

- 1. Establish qualification standards for a given suspected intrinsic shortcut.
- Design a training curriculum (training data sets and training strategies) to train the intrinsic shortcut detection deep neural network models ("shortcut detectives") to detect the intrinsic shortcut.
- 3. Perform certification tests for the trained shortcut detectives to ensure they can pass the qualification standards.
- 4. Deploy the certified shortcut detective to the curated datasets to examine the suspected shortcuts.

This framework was implemented to train certified contrast- and sharpness-related "shortcut detectives" and applied to the COVID-19 datasets highlighted in Chapter 3 to assess the quality of these available datasets.

#### 4.4.1 Training and Certification of Shortcut Detective Models

To create the datasets used for training the shortcut detective models, 50,000 "normal" or "no findings" CXRs from the MIMIC dataset were randomly selected and divided into two equal groups: 25,000 of the CXRs were arbitrarily assigned as the positive class ("1"), while the remaining 25,000 were assigned as the negative class ("0").

To construct the training dataset for the shortcut detective, the image contrast or sharpness of the positive class was adjusted using the approach shown in Figure 4.10. In this study, for each CXR image, the image sharpness factor, *s*, was randomly sampled in the range [-1.0, 1.0], and the contrast factor, *c*, was sampled in the range of [0.015, 0.020]. An example of the subtle changes of these transformations on a CXR image is illustrated in Figure 4.13.

Since only normal CXRs are included, there are no disease-specific features present that can be used to distinguish the two classes. Therefore, if a model is able to differentiate the two classes, *it can only be because the model has learned the corresponding global image contrast or sharpness characteristics,* rather than any disease features. This ensures that there is no confounding of the shortcut features with disease-related features.

Next, shortcut detectives (deep neural network models for binary classification) were trained using the constructed training datasets with the added contrast or sharpness shortcut (for detailed training parameters, see Appendix A). To assess the effectiveness of shortcut detectives on detecting shortcuts in COVID-19 CXR datasets, two types of tests are needed. First, when applied to a COVID-19 CXR dataset without any corresponding shortcuts, the shortcut detective should be unable to distinguish between COVID-19 positive and COVID-19 negative cases, resulting in an AUC close to 0.5 (equivalent to random guessing). This test is crucial to ensure that the image features utilized by the shortcut detectives are not entwined with those of the original imaging task, as the imaging features of COVID-19 positive and COVID-19 negative CXRs are also influenced by image contrast and sharpness.

Second, when applied to a COVID-19 CXR dataset containing a known shortcut, the shortcut detective should demonstrate high classification accuracy, ideally achieving an AUC close to 1. For the first test, the HF dataset was utilized. As the positive and negative cohorts are gathered within the same time frame and from the same hospitals, no contrast and sharpness shortcuts are assumed



**Figure 4.13:** Examples of contrast (*c*) and sharpness (*s*) adjusted images according to the process shown in Figure 4.10. These adjustments only introduce very subtle changes to the original image, as demonstrated in these images.

to be present. This is also demonstrated in Chapter 5, where a model trained using this dataset exhibits consistent testing performance across multiple external COVID-19 clinical test datasets. For the second exam, known shortcuts are added to the COVID-19 positive class or COVID-19 negative class of the HF dataset using the same procedures as the MIMIC training dataset outlined in Figure 4.10.

To provide a clearer understanding, a simple analogy for this process is depicted in Figure 4.14. Imagine a dataset consisting of circles, with half of them colored blue and labeled "0" while the other half is green and labeled "1." A neural network is trained to classify these circles, and naturally, it learns to differentiate colors as this is the only distinguishing feature available.

Now, the trained network is tested on a dataset containing yellow triangles and squares. This



**Figure 4.14:** In this analogy, the shapes represent CXR disease features, while the color symbolizes contrast or sharpness shortcuts. A shortcut detective trained on added sharpness or contrast features (analogous to the color) should not be able to differentiate between COVID-19 positive/negative features (analogous to the shape).

test set is an external one, as the shapes and colors are entirely different from the training dataset. The network, having only learned to differentiate between blue and green, can only make random guesses in this case, corresponding to an AUC of 0.5.

Next, the trained network is tested on the same test set, but the colors have been altered to match the training conditions, with triangles being blue and squares green. In this scenario, the network performs its task perfectly with an AUC close to 1, as it was trained to discriminate color, regardless of the shapes.

In this analogy, the shapes represent CXR disease features, while the color symbolizes contrast or sharpness shortcuts. A shortcut detective trained on added sharpness or contrast features (analogous to the color) should not be able to differentiate between COVID-19 positive/negative features (analogous to the shape). The first test aims to ensure that the trained shortcut detective

**Table 4.3:** Five distinct model architectures that are widely employed for image classification and demonstrate state-of-the-art performance on ImageNet classification tasks were used as shortcut detectives in this work. Varying levels of architectural design and complexity, as characterized by the number of model parameters and floating-point operations (FLOPs), are exhibited by these models. Further technical information on model architecture can be found in Appendix A. ImageNet accuracy reported by: https://pytorch.org/vision/stable/models.html#classification.

Model	# of Parameters	FLOPs	ImageNet Accuracy	Year Developed
<b>VGG-16</b> <sup>132</sup>	138 M	31 B	73.4%	2014
DenseNet-121 <sup>127</sup>	8 M	5.7 B	74.4%	2017
EfficientNet <sup>119</sup>	54 M	24 B	85.1%	2020
Swin Transformer <sup>133</sup>	88 M	15 B	83.6%	2021
<b>ConvNeXt</b> <sup>134</sup>	89 M	15 B	84.1%	2022

only recognizes the specific shortcut on which it was trained, and when it is absent, its accuracy is expectedly low.

In the second exam, the sharpness or contrast features are added to the test set, such that the shortcut features (analogous to the color) are perfectly correlated with the disease features (analogous to the shape). In this case, the network can classify with high accuracy, as it has learned the shortcut. The second exam's purpose is to ensure that, in the presence of other features, the shortcut detective can identify the shortcut with high accuracy.

Finally, if the trained shortcut detectives successfully complete the two tests—achieving an AUC close to 0.5 on the dataset without shortcuts and an AUC close to 1.0 on the dataset with shortcuts—they are deemed certified shortcut detectives. An overview of the training and certification process is shown in Figure 4.15.

In this study, five distinct model architectures (see Table 4.3) that are widely employed for image classification and demonstrate state-of-the-art performance on ImageNet classification tasks have been explored. Although every potential model architecture cannot be examined, the investigation encompasses both classic and modern convolutional networks, as well as the recently developed Swin Transformer (originally designed for natural language processing applications). Varying levels of architectural design and complexity, as characterized by the number of model parameters and floating-point operations (FLOPs), are exhibited by these models. Further technical information on model architecture and training can be found in Appendix A.

The certification exam results for the models listed in Table 4.3 are displayed in Table 4.4.



**Figure 4.15:** Overview of the training and certification of shortcut detective deep neural network models. 50,000 "normal" CXR images were randomly selected from the MIMIC dataset and split evenly into positive ("1") and negative ("0") classes. Next, the contrast or sharpness shortcut was introduced to the positive class using the approach shown in Figure 4.10, and a neural network was trained. To assess the effectiveness of shortcut detectives on the COVID-19 CXR dataset, two types of tests are needed. First, when applied to a COVID-19 CXR dataset without any corresponding shortcuts, the shortcut detective should be unable to distinguish between COVID-19 positive and COVID-19 negative cases, resulting in an AUC close to 0.5. Second, when applied to a COVID-19 CXR dataset containing a known shortcut, the shortcut detective should demonstrate high classification accuracy, ideally achieving an AUC close to 1. Finally, if the trained shortcut detectives successfully complete the two tests, they are deemed certified shortcut detectives.

**Table 4.4:** Results of the certification exams for the models listed in Table 4.3. An AUC of 0.0 is simply due to the assignment of class labels, which is equivalent to an AUC of 1.0. Both indicate perfect classification performance. Note: (+) means the shortcut is added to the COVID-19 positive CXRs; (-) means the shortcut is added to the COVID-19 negative CXRs.

	VGG-16	Densenet-121	EfficientNet	Swin Transformer	ConvNeXt	
Sharpness Shortcut Detective						
Exam 1:	0.49	0.55	0.56	0.55	0.51	
HF	[0.48, 0.50]	[0.54, 0.56]	[0.55, 0.57]	[0.54, 0.56]	[0.50, 0.51]	
Exam 2:	1.00	1.00	1.00	1.00	1.00	
HF (+)	[1.00, 1.00]	[1.00, 1.00]	[1.00, 1.00]	[1.00, 1.00]	[1.00, 1.00]	
Exam 2:	0.00	0.00	0.00	0.00	0.00	
HF (-)	[0.00, 0.00]	[0.00, 0.00]	[0.00, 0.00]	[0.00, 0.00]	[0.00, 0.00]	
Contrast Shortcut Detective						
Exam 1:	0.47	0.50	0.49	0.50	0.50	
HF	[0.46, 0.48]	[0.50, 0.51]	[0.48, 0.49]	[0.49, 0.50]	[0.50, 0.51]	
Exam 2:	1.00	1.00	1.00	1.00	1.00	
HF (+)	[1.00, 1.00]	[1.00, 1.00]	[1.00, 1.00]	[1.00, 1.00]	[1.00, 1.00]	
Exam 2:	0.00	0.00	0.00	0.00	0.00	
HF (-)	[0.00, 0.00]	[0.00, 0.00]	[0.00, 0.00]	[0.00, 0.00]	[0.00, 0.00]	

It is important to note that shortcuts were introduced to both COVID-19 positive CXRs in one set and to COVID-19 negative CXRs in another separate set. The notation (+) signifies that the shortcut is added to the COVID-positive CXRs, while (-) indicates that the shortcut is added to the COVID-positive CXRs, while (-) indicates that the shortcut is added to the COVID-negative CXRs. Table 4.4 also demonstrates that all five neural network architectures exhibit comparable performance levels, which is consistent with the understanding that contrast and sharpness shortcuts are inherent to the dataset and the overall classification task.

#### 4.4.2 Certified Shortcut Detective Investigations on COVID-19 CXR Datasets

After the shortcut detective models were trained and validated, they were employed to examine intrinsic shortcuts in real-world COVID-19 CXR datasets, including the BIMCV, UW, MIDRC, and COVIDx datasets. A detailed overview of each dataset is provided in Chapter 3. Briefly, UW Health is a single-site, privately curated dataset; BIMCV and MIDRC are both multi-institutional public datasets; COVIDx is the first open-access COVID-19 CXR dataset released by experts in the


**Figure 4.16:** Real CXR images from the MIMIC dataset compared to synthetic CXRs generated by the RoentGen model, a GAN that produces highly realistic images that are visually challenging to differentiate from authentic CXR images.

computer science community.

One method for augmenting datasets involves using generative adversarial networks<sup>135</sup> (GANs) to create artificial images, <sup>136,137</sup> generating additional training samples. GANs are a type of deep neural network designed for unsupervised learning tasks, particularly aimed at generating new data samples resembling a given dataset. To investigate whether these generative techniques can introduce shortcut features, a dataset containing 1,000 synthetic CXRs created by the RoentGen model <sup>138</sup> and 1,000 real "normal" CXRs from the MIMIC dataset was assembled. The RoentGen model, trained using the MIMIC dataset, can generate visually convincing synthetic CXRs with various pathologies. For this work, a "no finding" text prompt was used as input to generate synthetic CXRs, including only frontal view images. Examples CXR images are shown in Figure 4.16.

Analogous to the detective certification process, for a COVID-19 CXR dataset where COVID-19 positive cases receive a label of "1" and negative cases a label of "0" if the shortcut detective can distinguish the two classes (i.e., AUC significantly deviates from 0.50), it indicates the presence of the corresponding shortcut. For the RoentGen-MIMIC dataset, real CXRs are labeled as "1" and

synthetic CXRs as "0." The results of the trained shortcut detectives for the COVID-19 and RoentGen datasets are displayed in Table 4.5. The findings reveal that shortcuts related to image sharpness and contrast indeed exist in the COVIDx dataset. Models trained using such datasets inevitably exploit these shortcuts and consequently cannot be generalized in real clinical settings. In contrast, the other three datasets curated by medical professionals do not exhibit such shortcuts.

Additionally, as shown in Table 4.5, major contrast and sharpness differences were also detected in the RoentGen dataset. While the generated synthetic CXRs appear visually realistic, caution must be exercised when using them for AI model development due to the potential learning of shortcuts caused by the inherent contrast and sharpness differences between real and synthetic data.

#### 4.4.3 Using Shortcut Detectives To Explain Generalizability

To substantiate the assertion that datasets with pronounced shortcuts, as identified by the shortcut detectives, suffer from poor generalizability, a comparative analysis of performance was carried out between two models: one trained on the COVIDx dataset and the other on the HF dataset. Both datasets are of a similar size, but as demonstrated in Table 4.5, the COVIDx dataset is affected by sharpness and contrast shortcuts (among others) as determined by the shortcut detectives. Utilizing a DenseNet-121 architecture, models were trained on the COVIDx and HF datasets, and their performance was then assessed on hold-out test sets as well as external datasets, namely UW, BIMCV, and MIDRIC. The results, displayed in Table 4.6, reveal that the COVIDx model exhibits subpar generalization performance, as indicated by the considerable AUC gap between internal and external tests. In contrast, the HF model demonstrates consistent performance across both internal and external tests.

The abundance of shortcut features in the COVIDx dataset, which arises from the combination of data from diverse sources, accounts for the near-perfect accuracy achieved on hold-out test sets by models developed using this dataset. However, these models ultimately lacked clinical utility as they didn't learn any generalizable features of COVID-19 infection. The presence of contrast and sharpness shortcuts by the shortcut detectives further clarifies why attempts at lung segmentation or implementing alternative model architectures were unsuccessful in enhancing generalizability, as these intrinsic shortcuts are embedded within the images themselves. **Table 4.5:** Results of the trained shortcut detectives for the COVID-19 and RoentGen datasets. These findings reveal that shortcuts related to image sharpness and contrast indeed exist, particularly in the COVIDx and RoentGen datasets. Models trained using such datasets inevitably exploit these shortcuts and consequently cannot be generalized in real clinical settings. In contrast, the other three datasets curated by medical professionals do not exhibit such shortcuts.

	COVIDx	UW Health	BIMCV	MIDRC	RoentGen		
VGG-16							
Sharpness	0.71	0.49	0.49	0.44	0.98		
Detective	[0.70, 0.71]	[0.47, 0.51]	[0.48, 0.50]	[0.43, 0.45]	[0.97, 0.98]		
Contrast	0.78	0.49	0.57	0.48	0.94		
Detective	[0.78, 0.79]	[0.47, 0.50]	[0.56, 0.58]	[0.48, 0.49]	[0.94, 0.95]		
		DenseNe	et-121				
Sharpness	0.83	0.51	0.47	0.46	0.70		
Detective	[0.83, 0.84]	[0.49, 0.52]	[0.46, 0.48]	[0.45, 0.46]	[0.68, 0.72]		
Contrast	0.78	0.55	0.50	0.42	0.88		
Detective	[0.70, 0.71]	[0.54, 0.57]	[0.49, 0.52]	[0.42, 0.43]	[0.86, 0.89]		
EfficientNet							
Sharpness	0.84	0.50	0.46	0.40	1.00		
Detective	[0.83, 0.84]	[0.49, 0.52]	[0.45, 0.47]	[0.39, 0.41]	[0.99, 1.00]		
Contrast	0.79	0.51	0.55	0.48	0.94		
Detective	[0.78, 0.79]	[0.50, 0.53]	[0.53, 0.56]	[0.48, 0.49]	[0.93, 0.95]		
		Swin Trans	sformer				
Sharpness	0.77	0.52	0.48	0.46	0.73		
Detective	[0.77, 0.78]	[0.50, 0.53]	[0.47, 0.49]	[0.45, 0.47]	[0.70, 0.75]		
Contrast	0.78	0.52	0.58	0.48	0.91		
Detective	[0.77, 0.78]	[0.50, 0.53]	[0.57, 0.59]	[0.47, 0.49]	[0.90, 0.93]		
ConvNeXT							
Sharpness	0.86	0.53	0.48	0.36	0.98		
Detective	[0.85, 0.86]	[0.51, 0.55]	[0.47, 0.49]	[0.35, 0.37]	[0.97, 0.98]		
Contrast	0.60	0.46	0.49	0.42	0.75		
Detective	[0.59, 0.61]	[0.44, 0.48]	[0.48, 0.50]	[0.42, 0.43]	[0.72, 0.77]		

**Table 4.6:** To demonstrate that datasets with pronounced shortcuts, as identified by the shortcut detectives, suffer from poor generalizability, a comparative analysis of performance was carried out between two models: one trained on the COVIDx dataset and the other on the HF dataset. Their performance was then assessed on hold-out test sets as well as external datasets, namely UW, BIMCV, and MIDRIC. The COVIDx model exhibits subpar generalization performance, as indicated by the considerable AUC gap between internal and external tests. In contrast, the HF model demonstrates consistent performance across both internal and external tests.

	Internal Test	UW Test	BIMCV Test	MIDRC Test
COVIDx	0.99	0.60	0.62	0.57
	[0.99, 1.00]	[0.58, 0.61]	[0.61, 0.63]	[0.56, 0.58]
HF	0.78	0.77	0.78	0.75
	[0.76, 0.80]	[0.75, 0.79]	[0.77, 0.79]	[0.74, 0.75]

## 4.5 Discussion

Undoubtedly, deep learning has achieved remarkable success in object detection, image classification, and natural language processing over the past decade, primarily due to its capacity for learning complex features from data. However, despite its impressive performance on benchmark datasets, it is becoming increasingly evident that there are limitations and practical challenges when applying these models in real-world situations. A key issue is poor generalizability, where performance significantly declines when models are used on external datasets. This hampers the adoption and deployment of deep learning models in high-stakes domains, such as healthcare applications. The COVID-19 pandemic painfully highlighted this issue, as numerous machine learning models were developed, but very few demonstrated strong performance in real-world clinical tests or made any tangible clinical impact.

The phenomenon of shortcut learning has emerged as a subject of recent deep learning research. Studies have shown that poor model generalizability can result from shortcut learning when training datasets contain overlooked shortcuts, which are spurious correlations between irrelevant image features and their associated training labels. Consequently, models quickly latch onto these spurious correlations instead of the intended image features, forming inaccurate associations between input image data and output labels. Medical imaging is particularly susceptible to shortcut learning due to the limited availability of data. To address this issue, smaller datasets are often combined with other sources to maximize the training size for deep learning. However, varying characteristics between data sources can lead networks to learn differences between datasets rather than the



**Figure 4.17:** The detective framework effectively serves as a litmus test for detecting shortcut features in a dataset. Analogous to the color change in litmus paper that helps determine whether a solution is acidic or basic (alkaline), the shortcut detectives function like a litmus sheet when tested on a dataset (solution), and the AUC measurement (color change) indicates whether the dataset is biased toward a particular shortcut feature.

desired features for classification tasks.

This chapter highlights that virtually any population or imaging characteristic can become a potent shortcut if correlated with the classification label. Worryingly, these shortcuts may lack any visual explanation, rendering them undetectable through qualitative inspection or model saliency methods. As a response, a systematic approach for training and validating shortcut detectives for CXR classification has been developed, focusing on image contrast and sharpness—two essential intrinsic properties of CXR images. Nonetheless, contrast and sharpness are merely a subset of the countless potential features in a dataset that can be leveraged for shortcut learning. If other intrinsic shortcuts are suspected within a dataset, the general framework proposed in this work can be utilized to develop similar shortcut detectives and identify the alleged intrinsic shortcuts.

This framework effectively serves as a litmus test for detecting shortcut features in a dataset, as shown in Figure 4.17. Analogous to the color change in litmus paper that helps determine whether a solution is acidic or basic (alkaline), the shortcut detectives function like a litmus sheet when tested on a dataset (solution), and the AUC measurement (color change) indicates whether the dataset is biased toward a particular shortcut feature. This provides a quantitative metric for data quality and enables the testing of features that might be challenging to discern using alternative methods such as saliency maps. However, one limitation of this method is the requirement to know and train for the specific shortcut beforehand. This is where domain expertise becomes crucial in both the dataset collection and analysis processes.

After identifying shortcut features, it is crucial to devise strategies to mitigate their influence on the learned models. Although this may seem straightforward in theory, eliminating shortcuts can prove to be quite challenging in practice. To illustrate this, consider the BIMCV+/HF- trained model depicted in Figure 4.8, which exhibits strong shortcut learning. Despite applying histogram equalization and segmentation, significant discriminating features persist between the two dataset sources. To determine if intrinsic contrast and sharpness shortcuts exist in this model, the developed contrast and sharpness shortcut detectives were evaluated on the BIMCV+/HF- dataset, yielding AUC values of 0.50 (95% CI: [0.462, 0.534]) and 0.67 (95% CI: [0.641, 0.708]), respectively. While histogram equalization succeeded in mitigating contrast differences, a weak sharpness difference remained between the datasets, along with lung shape differences (refer to Section 4.2.2). These features, in conjunction with other unidentified shortcut features, contribute to the strong shortcut learning exhibited by this model. However, addressing these shortcuts becomes problematic, as it is not immediately clear how to homogenize lung shape or image sharpness, for example.

One potential solution is to establish standardization and normalization techniques for image contrast and sharpness, allowing adjustments to these attributes without impacting disease features. Alternatively, examining proven intrinsic shortcut-free datasets, such as the baseline dataset (Henry Ford Health) and the three additional datasets (UW Health, BIMCV, and MIDRC), demonstrated to be devoid of intrinsic shortcuts in this study, may offer insights into avoiding these shortcuts during the data curation process. Nonetheless, it is important to acknowledge that this is a limitation of the current work, and future research should explore the development of mitigation strategies for the identified shortcuts.

Currently, the most prudent approach might be to minimize the introduction of shortcut learning by refraining from combining datasets whenever possible. As demonstrated in this chapter, the notion that "more is better" may not always hold true when developing deep learning models. Chapter 5 delves into the topic of data size, illustrating how high-performing models can still be developed without relying on conventionally large datasets.

## Chapter 5

# Mastering Generalizability: Training and Optimizing a Robust Deep Learning Network for COVID-19 Classification

The potential of AI to revolutionize various aspects of healthcare, enabling more accurate, efficient, and personalized patient care, has been widely recognized. AI-powered tools and techniques, especially deep learning-based approaches, have demonstrated remarkable performance in tasks such as medical image analysis, diagnostics, and treatment planning. In disciplines like radiology, pathology, and ophthalmology, AI models have demonstrated the ability to analyze medical images <sup>69–71,139</sup> and identify diseases with precision comparable to or even surpassing that of human experts. Additionally, AI has been instrumental in drug discovery<sup>140</sup> and repurposing, significantly reducing the time and cost associated with traditional methods. Natural language processing techniques<sup>141</sup> have facilitated the extraction of valuable information from electronic health records and scientific literature, further enhancing decision-making and clinical research. Moreover, AI-driven predictive analytics<sup>142</sup> are being used to identify patterns in patient data, enabling early intervention and personalized treatment plans. However, questions remain about the clinical utility of these AI advancements.

The U.S. Food and Drug Administration (FDA) approved its first AI algorithm in 1995, and fewer than 50 algorithms were approved over the next 18 years. Nonetheless, the numbers have surged in the past decade. The number of AI and machine learning-enabled devices reviewed by the FDA more than doubled from 2017 to 2018, and the growth has persisted. In 2021, the FDA authorized a record 115 submissions,<sup>143</sup> an 83% increase from 2018. However, most devices that



**Figure 5.1:** The growth of AI-approved medical devices and algorithms by the FDA has surged in the past decade. Of the 521 submissions the FDA has authorized to date, greater than 75% have been in radiology. Data from https://www.fda.gov/medical-devices/software-medical-device-samd/artificial-int elligence-and-machine-learning-aiml-enabled-medical-devices.

have undergone FDA review have received 510(k) clearance, which does not necessitate clinical trials if developers can demonstrate that their device is "substantially equivalent" to an existing one. To date, 96% of authorized AI and machine learning-enabled medical devices have 510(k) clearance, while only three devices have undergone the FDA's more rigorous premarket approval process. Any major changes to a medical device must be cleared by the FDA, causing most algorithms to remain static after market introduction. Although the discussion of proper regulations, policies, and ethics regarding AI adoption in medicine goes beyond the scope of this work, significant conversations around these issues are ongoing.<sup>144–148</sup>

Despite AI's promise, numerous obstacles impede its implementation in daily clinical practice. These challenges include issues with transparency surrounding software programs, inherent biases in the data used, and security concerns. The most pressing concern with AI in medicine stems from the lack of generalizability in developed models. The failure of models to generalize when applied across institutions with heterogeneous populations and imaging protocols has been cited as a primary obstacle in incorporating AI-based decision-making tools in medicine.<sup>149</sup> Moreover, most machine learning-based healthcare studies are unable to test on external patient cohorts, resulting in a gap between locally reported model performance and cross-site generalizability.<sup>150</sup> Demonstrating a deep learning model's generalizability efficiently and sufficiently before implementing it in clinical practice remains challenging.<sup>151</sup> Therefore, addressing generalizability is critical for the success of

clinical AI solutions.

## 5.1 The Enigma of Deep Neural Network Generalizability

Although simple to state and understand in general, the problem of deep learning generalizability is incredibly complex and poorly understood. Generalizability in deep learning refers to the ability of a trained model to perform well on new, unseen data that was not part of the training dataset. In other words, it measures how well a model can extrapolate its learned knowledge from the training data to make accurate predictions or classifications on previously unencountered data.

A deep learning model with good generalization can effectively handle the variability and complexities of real-world data and thus be more useful and reliable in practical applications. On the other hand, a model with poor generalization might perform well on the training data but struggle with new data, leading to suboptimal or incorrect predictions. Generalizability is a critical aspect of deep learning models, particularly in scenarios where it is crucial to ensure that the model performs consistently and accurately across different datasets, such as in medical diagnostics, autonomous vehicles, and financial predictions.

Various factors can impact the generalizability of a deep learning model, including the quality and diversity of the training data, the model's architecture, the loss function, and the training process (including hyperparameters and regularization techniques). To improve generalizability, researchers often employ techniques such as data augmentation, regularization (e.g., dropout, weight decay), and cross-validation to reduce overfitting and ensure that the model learns meaningful patterns instead of memorizing the training data.

However, fundamentally, the generalization of DNNs remains a somewhat unexpected and enigmatic phenomenon. While it is known that deep networks are universal approximators, capable of representing arbitrarily complex functions given sufficient capacity, <sup>152–154</sup> the traditional view of generalization holds that over-parameterized models (e.g., more parameters than training examples) should "memorize" each example, overfitting the training set and yielding poor generalization to validation and test sets.<sup>92</sup> Yet, DNNs often achieve excellent generalization performance with massively over-parameterized models, a phenomenon that is not well-understood.<sup>155–158</sup>

One theory behind the generalization of over-parameterized models involves hierarchical feature



**Figure 5.2:** For reasons not well understood, DNNs generally do not memorize real data (left) but instead learn simple patterns (right) before resorting to memorization. Implicit regularization in deep learning may also contribute to the generalization ability of over-parameterized models. During training, deep networks may undergo implicit regularization, encouraging simpler solutions and reducing the risk of overfitting. This can result from optimization algorithms such as stochastic gradient descent or its variants, which favor flatter minima and simpler models.

learning.<sup>7</sup> Deep networks can learn hierarchical representations of data, with lower layers learning basic features and higher layers learning increasingly complex and abstract patterns by combining lower-layer features. This allows deep networks to efficiently capture and represent the underlying structure and patterns in the data, enabling better generalization.

Implicit regularization in deep learning may also contribute to the generalization ability of over-parameterized models. During training, deep networks may undergo implicit regularization, encouraging simpler solutions and reducing the risk of overfitting. This can result from optimization algorithms such as stochastic gradient descent or its variants, which favor flatter minima and simpler

models. Regularization techniques like dropout, weight decay, and batch normalization can also help improve generalization. However, some experiments suggest that this might not be the case, demonstrating that DNNs can fit pure noise without substantially longer training time.<sup>157</sup>

Despite these contradictory results and intuitions, DNNs generally do not memorize real data but instead, learn simple patterns before resorting to memorization.<sup>155</sup> This contributes to the widespread adoption and success of modern deep networks, although it also leads to the "black-box" nature and general lack of trust in these networks. Understanding generalizability at a fundamental level is beyond the scope of this work, which aims to create a model that retains performance across multiple datasets. It is important to remain conscious that this generalizability is not rigorously proven or guaranteed. Bearing this in mind, the focus of this chapter is on the technical development of a generalizable COVID-19 classification model.

## 5.2 Generalizability of a Single-Source Dataset

The challenge of poor generalizability in AI models is often linked to the size and quality of training data. As illustrated in Chapter 4, improper data collection and curation strategies can lead to spurious confounding factors, or "shortcuts," in the training data. Consequently, models learn these shortcuts instead of the desired disease features, resulting in high performance on internal test sets but poor performance on real-world clinical datasets.

Although it is commonly believed that increasing dataset size and gathering data from multiple institutions can lead to more generalizable models,<sup>89</sup> regulatory challenges, and the urgency of pandemic response make this approach difficult and potentially suboptimal. Moreover, as the previous chapter demonstrated, combining data from various sources can inadvertently introduce shortcut learning, as models may be more likely to memorize differences between sources rather than general disease features.

The challenge, therefore, lies in balancing the benefits of large datasets (such as increased data diversity and reduced overfitting) with the associated costs (such as data sharing negotiations, time required for data collection, and the risk of shortcut learning). In this context, using a single-source dataset can be advantageous. The primary question is whether it is possible to develop a model from a relatively small dataset that can generalize to external sites. To investigate this, a study was

conducted in which a deep learning model was trained using a single COVID-19 CXR dataset and evaluated on external datasets to assess its generalization performance. The key to success lies in ensuring the chosen dataset's data quality is sufficient. As demonstrated in Chapter 3, the HF dataset is diverse in terms of patient population characteristics and homogeneous in data quality, as all images were collected within the same hospitals and timeframe. Chapter 4 further demonstrated that this dataset is free from intrinsic shortcuts. Consequently, the HF dataset was selected as the training dataset with the aim of creating a generalizable network.

#### 5.2.1 Preparing High-Quality Training and External Evaluation Datasets

Extensively discussed in Chapter 3 and briefly reviewed here, the model development process utilized CXRs from patients with confirmed COVID-19 diagnoses at Henry Ford Health, which includes five hospitals. Patients who underwent frontal view CXRs, had COVID-19 diagnoses confirmed by RT-PCR tests, and received imaging between March 1, 2020, and September 30, 2020, were included. Both COVID-19 positive and negative data were collected within the same timeframe. This ensures the data distribution reflects the patient cohorts in which the algorithm is deployed and mitigates potential risks of shortcut learning when using pre-pandemic data as the control group. Patients under the age of 18 were excluded. To maintain consistency between imaging data and RT-PCR test results, CXRs performed more than seven days before or after the RT-PCR test were excluded, resulting in a delta window of [-7, 7] days. After applying these criteria, the training dataset for model development comprised of 8,335 COVID-19 positive CXRs from 4,383 patients and 16,584 COVID-19 negative CXRs from 8,733 patients.

To thoroughly evaluate the generalizability of the trained network, multiple test datasets were employed, which are summarized as follows:

 HF Temporal Internal Test Set: Evaluating model performance on prospectively collected datasets is crucial in determining if a model learns relevant and generalizable disease features. This type of dataset is curated during a time frame following the training data collection period and is referred to as an internal temporal test set. For this purpose, a test set from Henry Ford Health was used to evaluate the trained model over time. This test set consists of consecutive patient cases received during October 2020. A narrow delta window of [-3, 3] days ensured testing integrity by only including CXRs performed close to the RT-PCR test. The resulting dataset (HF-temporal) consisted of 695 COVID-19 positive CXRs from 526 patients and 8,878 COVID-19 negative CXRs from 6,081 patients.

- 2. BIMCV External Test Set: BIMCV<sup>102</sup> is a public COVID-19 chest X-ray dataset collected from 11 hospitals in the Valencian Region, Spain, between February and April 2020. A narrow delta window of [-3, 3] was used to include CXRs performed close to the RT-PCR test. The resulting dataset consisted of 4,169 COVID-19 positive CXRs from 2,663 patients and 5,050 COVID-19 negative CXRs from 3,710 patients.
- 3. UW Health External Test Set: This dataset contains consecutive patient cases from the University of Wisconsin Hospitals and Clinics, collected between March 2020 and September 2021, with the same inclusion criteria used above. The dataset comprised 1,019 COVID-19 positive CXRs from 654 patients and 8,597 COVID-19 negative CXRs from 5,908 patients.
- 4. MIDRC External Test Set: MIDRC<sup>103</sup> provides an open, diverse, and multi-institutional COVID-19 imaging dataset. Using similar inclusion criteria, a total of 6,453 COVID-19 positive CXRs from 5199 patients and 20,072 COVID-19 negative CXRs from 9,947 patients were included in this study.

As shown in Chapter 3, the last three evaluation datasets constitute as external datasets with different patient cohorts, population statistics, and imaging characteristics. Chapter 4 has also shown these datasets to be relatively shortcut-free, ensuring sufficient data quality.

An EfficientNet architecture was used, and the specifics of model training and parameters are given in Appendix A.

#### 5.2.2 Model Evaluation and Generalizability

After training on the HF training dataset, the model was evaluated on an internal test temporal test set and three external test sets. As shown in Table 5.1, the performance of the trained model on the external test sets is consistent with internal temporal validation, even though the test population and imaging system vendor distributions are entirely different between the training and the external test sets. For sensitivity and specificity calculations, no dataset-specific threshold tuning was applied; a

**Table 5.1:** Comparison of the HF trained model performance on internal and external test sets. The performance of the trained model on the external test sets is consistent with internal temporal validation, even though the test population and imaging system vendor distributions are entirely different between the training and the external test sets. For sensitivity and specificity calculations, no dataset-specific threshold tuning was applied; a fixed threshold value of 0.5 was used for all tests, further demonstrating the generalizability of the trained model.

	HF Temp Test	<b>BIMCV</b> Test	UW Health Test	MIDRC Test
AUC	0.80	0.80	0.78	0.76
AUC	[0.78, 0.82]	[0.79, 0.81]	[0.77, 0.80]	[0.76, 0.77]
Soncitivity (%)	47.6	57.4	39.5	45.3
Sensitivity (78)	[43.9, 51.4]	[55.9, 58.9]	[36.4, 42.6]	[44.1, 46.5]
Specificity (%)	93.2	87.3	94.5	91.7
	[92.6, 93.7]	[86.6, 88.2]	[94.0, 95.0]	[91.3, 92.0]
P-Value (AUC)	_	0.65	0.23	0.002

fixed threshold value of 0.5 was used for all tests, further demonstrating the generalizability of the trained model.

## 5.3 Influential Factors in Model Generalization

In order to delve deeper into the generalizability of the HF-trained model, a series of comprehensive experiments were conducted to examine the influence of different factors on the model's generalizability. These factors encompassed various aspects, such as network architecture and input image size, training dataset size and pretraining approach, severity of disease manifestation in CXRs, as well as patient demographics like age, sex, and ethnicity. By scrutinizing the interplay between these variables and the model's generalizability, these studies aim to shed light on the extent to which a single-source model can adapt and maintain its performance across diverse settings and conditions.

#### 5.3.1 Network Architecture and Image Size

Convolutional neural networks are designed to efficiently extract features from images. However, there are hundreds of distinct CNN architectures, each with unique features and strengths. It is crucial to investigate the performance of various CNN architectures because each one offers distinct

advantages and characteristics that can impact their suitability for specific tasks or problems. By examining multiple architectures, decisions can be made to select a model that not only delivers the highest performance but also balances computational efficiency, transfer learning potential, customizability, interpretability, and ultimately robustness and generalizability.

For this study, DenseNet-121, EfficientNet, and Swin Transformer architectures were analyzed for generalization performance. These models were chosen because they are widely used and represent state-of-the-art CNN classification models (see Table 4.3). Each model was trained using the HF dataset described earlier and evaluated on the same hold-out and external test sets mentioned previously. Further information on specific architecture details and training for each model can be found in Appendix A.

CNNs are often trained on downsized images for several reasons, primarily related to computational efficiency and resource constraints. Downsizing images significantly reduces the number of pixels, which in turn decreases the number of parameters and calculations the CNN has to process. This enables faster training and inference while maintaining a manageable memory footprint. However, downsizing images may also lead to a loss of fine-grained details and high-frequency information that could be essential for certain tasks. To study how image size affects performance and generalization, each model architecture was trained not only on the standard  $224 \times 244$  resolution but also on  $480 \times 480$  resolution images.

The results for each architecture and input image size on the test sets described previously are shown in Table 5.2. The results vary slightly with different architectures and image sizes, with the EfficientNet using 480×480 resolution images proving to be the highest-performing combination. The results demonstrate that increasing image size does not guarantee an increase in performance, consistent with observations in other studies.<sup>159</sup> Additionally, the average inference time for each 480×480 resolution image was 1.30 seconds compared to 0.43 seconds for 224×224 resolution images. However, training time on the higher resolution images increases by a much larger factor due to smaller batch sizes and processing operations. In general, performance and generalization remain consistent across different architectures and image sizes for the HF-trained models.

**Table 5.2:** Results for each architecture and input image size on the test sets. The results vary slightly with different architectures and image sizes, with the EfficientNet using 480×480 resolution images proving to be the highest-performing combination. In general, performance and generalization remain consistent across different architectures and image sizes for the HF-trained models.

	DenseNet		Efficie	entNet	Swin Transformer	
	224×224	<b>480</b> × <b>480</b>	224×224	<b>480</b> × <b>480</b>	224×224	480×480
IIE Tomm	0.779	0.787	0.777	0.797	0.788	0.773
III Iemp	[0.757, 0.797]	[0.767, 0.805]	[0.757, 0.799]	[0.777, 0.816]	[0.768, 0.806]	[0.749, 0.793]
DIMCV	0.788	0.773	0.790	0.802	0.781	0.791
DINCV	[0.778, 0.797]	[0.763, 0.782]	[0.781, 0.800]	[0.793, 0.811]	[0.772, 0.790]	[0.782, 0.801]
UW	0.772	0.781	0.774	0.781	0.763	0.769
	[0.754, 0.788]	[0.765, 0.798]	[0.756, 0.788]	[0.764, 0.797]	[0.747, 0.780]	[0.751, 0.785]
MIDRC	0.745	0.743	0.738	0.764	0.734	0.733
	[0.738, 0.753]	[0.736, 0.751]	[0.730, 0.745]	[0.757, 0.770]	[0.723, 0.741]	[0.723, 0.741]

#### 5.3.2 Disease Severity Measured by Lung Opacity Score

It is widely known that some COVID-19 patients may be asymptomatic or minimally symptomatic without exhibiting pulmonary infection on CXRs.<sup>160</sup> This presents a unique challenge for AI development, as the network is tasked with identifying COVID-19 features, but these cases may offer no learnable information. This scenario can potentially encourage the model to learn spurious features to minimize the loss. To address this issue, an independently trained abnormality detection model was utilized in this study to remove such cases.

Enhancing the efficacy and reliability of AI models for disease classification can be achieved by removing asymptomatic cases from the training dataset. Asymptomatic or minimally symptomatic individuals may not display noticeable pathological findings in their medical imaging, complicating AI-driven classification. Including these cases can lead to AI models learning spurious features or noise rather than true disease characteristics. Excluding asymptomatic cases allows the training dataset to focus on relevant and informative cases, ultimately leading to more accurate and robust disease classification models. This may improve the model's generalizability and applicability in real-world clinical settings, where accurate disease feature identification is vital for effective diagnosis and treatment planning.

To eliminate asymptomatic cases, a deep neural network classifier was trained to identify CXRs



**Figure 5.3:** Excluding asymptomatic cases allows the training dataset to focus on relevant and informative cases, ultimately leading to more accurate and robust disease classification models. To eliminate asymptomatic cases, a deep neural network classifier was trained to identify CXRs without pneumonia, lung opacities, or other lung abnormalities. Example CXRs with abnormality scores from the trained opacity detection model.

without pneumonia, lung opacities, or other lung abnormalities. A total of 94,967 CXRs from the MIMIC dataset were selected for training the classifier. Images with positive lung opacity, lung lesion, or pneumonia labels (41,115 CXRs) were assigned to the positive class ("1"), while those with "No Finding" labels (48,458 CXRs) were assigned to the negative class ("0"). The images were randomly shuffled and partitioned into an 80% training and 20% validation split, using a per-patient strategy to prevent data leakage.

The classifier model was built on a DenseNet-121 architecture with weights pre-trained from the ImageNet dataset. Additional details and training specifics can be found in Appendix A. The trained model achieved an AUC of 0.86 (95% CI: [0.85, 0.87]) on the validation set. By applying a threshold of 0.2 to the output probability scores, the model's sensitivity and specificity were determined to be 0.90 and 0.67, respectively. Example CXRs with abnormality scores are shown in Figure 5.3. For each CXR, the model outputs a score between 0 and 1, where "0" indicates no lung opacities present and "1" indicates lung opacities present.

Figure 5.4 displays the opacity score distribution for the HF training dataset, as well as the test sets. Using these opacity scores, the AUC from the HF-trained COVID-19 classifier model can be



Figure 5.4: The opacity score distribution for the HF training dataset, as well as the test sets.

analyzed in terms of opacity score, as presented in Figure 5.5. The AUC was calculated images in the corresponding opacity score bin using bootstrapping resampling. These results demonstrate that the trained classifier performs better on CXRs with higher opacity scores. This finding aligns with the intuition that higher opacity scores imply more disease features being encoded in the CXR, which in turn enhances the performance of the network.

#### 5.3.3 Patient and Imaging Characteristics

Model performance in classification tasks, such as COVID-19 detection in CXR images, can vary across different subgroups within the test set. This variability in performance may arise from factors such as differences in image acquisition, patient demographics, or disease presentation. For instance, the classification accuracy might differ between male and female patients, X-ray views, or modality techniques. As shown in Chapter 4, a significantly higher AUC across a specific characteristic could indicate bias in the trained model. Investigating the model's performance across various groupings can help identify potential areas for improvement and gain insights into the factors that contribute



**Figure 5.5:** Using the opacity scores from the trained opacity model, the AUC from the HF-trained COVID-19 classifier model can be analyzed in terms of opacity score. These results demonstrate that the trained classifier performs better on CXRs with higher opacity scores. This finding aligns with the intuition that higher opacity scores imply more disease features being encoded in the CXR, which in turn enhances the performance of the network.

to the observed discrepancies.

The performance of the HF-trained COVID-19 detection model was evaluated across different patient characteristics, including sex, view position, and modality, as shown in Table 5.3. These results indicate that the model's performance does vary across subgroups, but the variation is generally minor (within 0.1 AUC). Drawing concrete conclusions from these differences can be

**Table 5.3:** The performance of the HF-trained COVID-19 detection model evaluated across different patient characteristics, including sex, view position, and modality. While the model's performance does vary across subgroups, the variation is generally minor (within 0.1 AUC), and drawing concrete conclusions from these differences can be challenging.

	Male	Female	CR	DX	AP	PA
HF Tomp	0.822	0.769	0.766	0.824	0.803	0.767
in imp	[0.797, 0.845]	[0.736, 0.799]	[0.734, 0.793]	[0.796, 0.849]	[0.777, 0.826]	[0.749, 0.793]
BIMCV	0.821	0.782	0.801	0.800	0.809	0.800
DINCV	[0.809, 0.832]	[0.769, 0.796]	[0.788, 0.813]	[0.788, 0.814]	[0.796, 0.821]	[0.785, 0.814]
UW	0.785	0.783	0.689	0.794	0.791	0.742
	[0.763, 0.806]	[0.758, 0.807]	[0.639, 0.738]	[0.777, 0.810]	[0.773, 0.807]	[0.638, 0.818]
MIDRC	0.754	0.774	0.782	0.753	0.763	0.767
	[0.744, 0.764]	[0.764, 0.786]	[0.768, 0.797]	[0.745, 0.762]	[0.755, 0.771]	[0.738, 0.793]

challenging. For example, the UW dataset has an AUC of 0.689 for CR and 0.794 for DX, which might initially suggest that the model is biased toward DX images. However, these characteristics are often entangled with other characteristics, making it difficult to isolate specific causes. Moreover, comparisons of CR and DX on other datasets do not show significant deviations, suggesting that the discrepancy in generalization performance may be unique to the UW test data distribution.

The MIDRC dataset includes patient race information, and the model performance across this subgroup is presented in Table 5.4. An anomaly is observed with the Asian subgroup, even though the training data predominantly consists of white and black patients. This finding may be a spurious correlation, and further investigations are needed to determine whether race is genuinely correlated with model performance.

The analysis of model generalizability across patient subgroups has revealed some variability in performance. This natural fluctuation is expected due to the statistical nature of the data, as well as the complex relationships between various factors such as image acquisition, patient

**Table 5.4:** The performance of the HF-trained COVID-19 detection model evaluated across patient race. Further studies remain to determine whether race is truly correlated with model performance.

	White	Black	Asian	Other	Not Reported	Total
	(# Pts. 11,993)	(# Pts. 4,831)	(# Pts. 3,317)	(# Pts. 1,387)	(# Pts. 4,760)	(# Pts. 26,525)
AUC	0.742	0.762	0.848	0.816	0.712	0.764

demographics, and disease presentation. While minor variations in performance are anticipated, extreme differences could indicate potential bias in the trained model. Identifying and addressing such biases is crucial for developing more robust and generalizable models, ultimately leading to better performance across a wider range of patient populations and clinical settings.

#### 5.3.4 Dataset Size and Pretraining

The availability of large and diverse datasets is often assumed to be a critical factor in the development of robust and high-performing machine-learning models. This is particularly true for medical imaging applications, where data scarcity can significantly limit the potential of deep learning models. Acquiring medical image data can be challenging due to a variety of reasons, such as patient privacy concerns, data sharing restrictions, and the requirement for expert annotations. Additionally, medical imaging datasets often exhibit class imbalance and may not adequately represent the demographic diversity and range of disease presentations. As a result, the performance of models trained on limited datasets may suffer from overfitting or lack generalizability when applied to new and unseen data.

To address this limitation, studies were conducted to investigate the impact of dataset size on model performance and generalizability in the task of COVID-19 classification using CXR images. Models were developed using training datasets with CXRs from 100, 200, 400, 800, 1,200, 1,600, 2,000, and 6,000 patients (some patients may have more than one CXR) sampled from the entire training dataset with a 1:1 class ratio. For each size, ten random samplings were performed, and a corresponding model was trained. The mean and standard deviation of the AUC were evaluated. The models were identical in parameters and training, with only the training datasets being different. Model and training details are outlined in Appendix A.

Figure 5.6 displays the AUC of the models on the four test sets as a function of the training data size. The AUC values corresponding to the data sizes of 100, 200, 400, 800, 1,200, 1,600, and 2,000 were used to fit the parameters of a learning curve in the form of a power law  $y = aN^k + b$ , where y is the AUC of the model, N is the training data size (number of patients), and a, k, and b are parameters. Remarkably, the AUC predicted by this fitted function for the data size of 6,000 closely matches the actual measured AUC, demonstrating the excellent predictive power of the fitted function for larger sample sizes. The performance of models with different training data

	HF Temp	BIMCV	UW Health	MIDRC
100 Pts.	$0.732\pm0.016$	$0.725\pm0.023$	$0.729\pm0.024$	$0.700\pm0.025$
200 Pts.	$0.766\pm0.026$	$0.749\pm0.028$	$0.764\pm0.018$	$0.731\pm0.019$
400 Pts.	$0.786\pm0.009$	$0.772\pm0.015$	$0.779\pm0.009$	$0.747\pm0.010$
800 Pts.	$0.794 \pm 0.007$	$0.781\pm0.009$	$0.785\pm0.006$	$0.757\pm0.009$
1,200 Pts.	$0.799\pm0.008$	$0.787\pm0.007$	$0.791\pm0.005$	$0.764\pm0.009$
1,600 Pts.	$0.802\pm0.003$	$0.792\pm0.005$	$0.797\pm0.006$	$0.766\pm0.005$
2,000 Pts.	$0.808\pm0.004$	$0.796\pm0.005$	$0.800\pm0.005$	$0.771\pm0.006$
6,000 Pts.	0.819	0.811	0.813	0.786
(Predicted)	[0.805, 0.834]	[0.802, 0.820]	[0.797, 0.829]	[0.772, 0.799]
6,000 Pts.	0.818	0.809	0.813	0.786
(Measured)				

**Table 5.5:** The generalization performance of models with different training dataset sizes as shown in Figure 5.6. Note, values after  $\pm$  represent the standard deviation of the performance of the ten trained models per dataset size. Values in the square brackets are 95% confidence intervals of the prediction.

sizes is also reported in Table 5.5. As seen from these results, a significant performance gain can be observed when the training data size increases from 100 patients to 800 patients. Although adding more training data continually improves the model's performance, the performance gain rapidly becomes marginal. This weak power-law relationship is consistent with other studies<sup>161</sup> for various computer vision applications, where the authors demonstrate typical k values ranging from -0.07 to -0.35.

These results highlight that even a small dataset size, such as 100 patients, can produce a reliable baseline model, provided that the data is of sufficient quality. Such a baseline model can offer initial clinical value when integrated into the clinical workflow. Furthermore, the model can be continuously updated to enhance its performance in real-world clinical settings. The quality of the training data is crucial in the development of deep learning models, particularly when working with limited data. This underlines the importance of meticulous data curation and annotation accuracy when constructing medical imaging datasets.



**Figure 5.6:** Model performance in terms of AUC on the four test sets as a function of the training data size. The AUC values corresponding to the data sizes of 100, 200, 400, 800, 1,200, 1,600, and 2,000 were used to fit the parameters of a learning curve in the form of a power law  $y = aN^k + b$ , where y is the AUC of the model, N is the training data size (number of patients), and a, k, and b are parameters. Remarkably, the AUC predicted by this fitted function for the data size of 6,000 closely matches the actual measured AUC, demonstrating the excellent predictive power of the fitted function for larger sample sizes. A significant performance gain can be observed when the training data size increases from 100 patients to 800 patients. Although adding more training data continually improves the model's performance, the performance gain rapidly becomes marginal.

#### 5.3.4.1 The Value of Pretraining for Small Datasets

Data dependence is one of the most challenging problems in deep learning. Deep learning has a very strong dependence on massive training data compared to traditional machine learning methods because it needs a large amount of data to understand the latent patterns of data. Pretraining or transfer learning are powerful techniques employed in deep learning that enable the development of robust and high-performing models, even when dealing with limited labeled data.<sup>162–165</sup> In this approach, a neural network is initially trained on a large and diverse dataset, often unrelated to the specific task of interest. This pretraining phase allows the model to learn general features and representations that can be transferable across various tasks. Once the pretraining is complete, the model can be fine-tuned on the specific task using a smaller, task-specific dataset. This process significantly reduces the required amount of labeled data and training time for the target task while improving model performance and generalizability. For example, Figure 5.7 illustrates the loss curves for a DenseNet model trained on the HF dataset under three conditions: no pretraining, pretraining with ImageNet, and pretraining with NIH. The figure clearly shows that pretraining significantly accelerates the training process (i.e., reducing the number of epochs needed for convergence), enhances stability, and boosts classification performance. The benefits of pretraining and transfer learning have been widely demonstrated in various applications, including computer vision, natural language processing, and medical imaging, where high-quality labeled data is often scarce and difficult to obtain.

Reviewed in Chapter 2, in the training process of a CNN, the input training data plays a crucial role in learning filters that can effectively extract relevant information from the data. During training, the network adjusts its parameters (i.e., weights and biases) through a process called backpropagation, which involves minimizing a predefined loss function that measures the difference between the model's predictions and the ground truth labels. As the training progresses, the CNN learns a hierarchical set of filters, where the initial layers capture low-level features such as edges, textures, and simple patterns, while the deeper layers capture increasingly complex and abstract representations that are more specific to the task at hand. These learned filters enable the CNN to efficiently analyze and extract meaningful information from the input data. However, this can be an arduous and unstable process, especially for small training datasets. Standard practice in



**Figure 5.7:** Loss curves for a DenseNet model trained on the HF dataset under three conditions: no pretraining, pretraining with ImageNet, and pretraining with NIH. Pretraining significantly accelerates the training process (i.e., reducing the number of epochs needed for convergence), enhances stability, and boosts classification performance.

many modern CNN implementations is to initialize the network weights with those trained on the ImageNet dataset.<sup>166</sup> As shown in Chapter 3 for the deep learning feature extraction, models trained on ImageNet can already extract meaningful information from CXR images, despite not being trained to do so specifically.

The concept of pretraining can be extended beyond natural images, such as those in the ImageNet dataset, to encompass more specialized imaging tasks, including medical imaging. By further pretraining on specific features of the anatomy in question, the learned filters can be refined to better suit the task at hand. In the case of CXR imaging, various datasets with different classification tasks are available to facilitate this process.

It is important to note that the classification task in a CNN significantly influences the learning of kernels or filters throughout the network. The network's primary goal is to recognize and classify patterns in the input data that pertain to the specific task. To achieve this, the CNN refines its kernels during training to capture and emphasize the most discriminative features in the data.

To illustrate the impact of the classification task on the features learned, pretrained DenseNet models from the TorchXRayVision library<sup>167</sup> were used to extract features from the combined COVID-19 datasets, following a similar process to that described in Section 3.4.3 of Chapter 3. The extracted features were then visualized using UMAP, as shown in Figure 5.8. The CXR datasets used for training are summarized here:

1. NIH ChestX-ray14: Comprises 112,120 frontal-view X-ray images of 30,805 unique patients

with the text-mined fourteen disease image labels (where each image can have multi-labels) including atelectasis, consolidation, infiltration, pneumothorax, edema, emphysema, fibrosis, effusion, pneumonia, pleural thickening, cardiomegaly, nodule, mass, and hernia.

- RSNA Pneumonia Detection Challenge: A joint effort from RSNA, the Society for Thoracic Radiology, and MD.ai to label pneumonia cases found in the NIH ChestX-ray14.<sup>168</sup> The task was to identify and localize pneumonia in the selected cases.
- PadChest: Includes more than 160,000 images from 67,000 patients that were interpreted and reported by radiologists at Hospital San Juan (Spain) from 2009 to 2017, covering six different position views.<sup>169</sup> The reports were labeled with 174 different radiographic findings, 19 differential diagnoses, and 104 anatomic locations.
- 4. **CheXpert**: Consists of 224,316 chest radiographs of 65,240 patients, where each report was labeled for the presence of 14 observations as positive, negative, or uncertain.<sup>170</sup>
- 5. **MIMIC**: A large CXR dataset of 227,835 imaging studies for 65,379 patients with text-mined fourteen disease image labels.

As depicted in Figure 5.8, the classification task influences the features extracted and their relationships in high-dimensional space, even though each network was trained on CXR images. A detailed exploration and comparison of the extracted features from these models can be found in Appendix B. The main takeaway from this analysis is that the pretraining scheme can impact the learned kernels, even within similar datasets.

However, this effect may not be as drastic as it first appears because when fine-tuning the pretrained network to the new target task, the kernels or filters within the network undergo modifications to optimize their performance for the specific task at hand. While the pretraining scheme may initially impact the kernels, the fine-tuning process ensures that the network becomes better aligned with the desired task. Consequently, the differences in kernels resulting from the pretraining stage may not be as significant once the fine-tuning process is completed. This highlights the power of transfer learning and its ability to adapt and optimize a network for a new task, even when the pretraining and target tasks differ significantly. The success of fine-tuning stems from the



**Figure 5.8:** UMAP dimensionality reduction of extracted features from the combined COVID-19 datasets. The models were trained on different CXR datasets with different classification tasks, which influences what features are extracted and their relationships in high-dimensional space.

CheXpert

MIMIC

idea that the learned low-level features are transferable across different tasks and can serve as a useful foundation for learning more specialized features in the target domain.

The impact of the pretraining scheme on the kernels may not be as drastic as it first appears due to the fine-tuning process applied when adapting the pretrained network to the new target task. During fine-tuning, the kernels or filters within the network are modified to optimize their performance for the specific task at hand. Although the pretraining scheme might influence the kernels initially, the fine-tuning process ensures that the network becomes more aligned with the desired task. As a result, the differences in kernels arising from the pretraining stage may not be as substantial after the fine-tuning process is completed. This underscores the effectiveness of transfer learning and its capacity to adapt and optimize a network for a new task, even when the pretraining and target tasks differ significantly. The success of fine-tuning is rooted in the concept that the learned low-level features are transferable across various tasks and can provide a valuable foundation for learning more specialized features within the target domain.

As detailed in Appendix A, the results of the generalization in relation to dataset size (see Figure 5.6) were obtained using models that employed a two-stage pretraining scheme, initially on ImageNet and then on the NIH CXR dataset. The generalizability of the model is intrinsically linked to this pretraining, as it establishes the foundation for the learned kernels. To investigate the impact of pretraining on model generalizability, the previous study was repeated, comparing ImageNet pretraining with NIH pretraining. The results, shown in Figure 5.9, reveal that different pretraining schemes can significantly influence model performance, particularly when using smaller training datasets. As dataset size increases, this effect somewhat diminishes; however, pretraining on data and classification tasks more closely related to the final target data clearly enhances model performance.

These findings illustrate the crucial role pretraining plays in developing a generalizable deep learning network. It serves as more than just an advantageous starting point; in addition to stabilizing and accelerating training, pretrained weights can offer generalizable low-level feature extraction without the need for extensive training datasets. Furthermore, using pretraining with data similar to the target data can further improve performance and generalization. Additional studies are required to compare the effects of pretraining on performance and generalization between similar yet distinct data, such as pretraining using NIH versus RSNA datasets. Optimizing



**Figure 5.9:** Model performance in terms of AUC on the four test sets as a function of the training data size and ImageNet and NIH pretraining. Different pretraining schemes can significantly influence model performance, particularly when using smaller training datasets. As dataset size increases, this effect somewhat diminishes; however, pretraining on data and classification tasks more closely related to the final target data clearly enhances model performance. Optimizing pretraining can contribute to enhanced performance and generalization, which is particularly beneficial in medical applications where data availability is often a limiting factor.

pretraining can contribute to enhanced performance and generalization, which is particularly beneficial in medical applications where data availability is often a limiting factor.

### 5.4 Discussion

Generalization is perhaps the most important aspect of a deep learning solution, yet it continues to be a challenge to address fully, especially in the realm of medicine. The capacity of a model to generalize beyond its training data is vital for its practicality and applicability in real-world situations, such as medical diagnostics and treatment planning. Although the field has made remarkable progress, the current limitations of deep learning models in generalizing and adapting to unfamiliar data restrict their extensive use in the medical sector. Nonetheless, deep learning remains a potent tool for problem-solving, with numerous successes to its name.

In this study, generalizability was explored in relation to dataset size and quality. To tackle the common issue of limited data in medical imaging and the risk of shortcut learning when combining datasets, a model was developed using a single data source, demonstrating its ability to generalize to real-world external data. Contrary to the popular belief that diverse data sources are necessary for highly generalizable AI models, this research showed that, with meticulous data curation, it is feasible to train AI models with strong generalizability using a single data source. The model's generalizability was assessed across various patient and imaging demographics, proving its consistency in diverse contexts.

Furthermore, generalizability was examined for different training dataset sizes, revealing that even a small dataset of a few hundred images can achieve a solid baseline performance. This outcome was found to rely on pretraining schemes, which were studied and shown to alleviate the challenges associated with small datasets. Moreover, pretraining with data related to the target data resulted in enhanced performance and generalization, underscoring the potential of carefully curated data and pretraining strategies in addressing the generalization challenge in deep learning. This is a particularly relevant result for medical imaging applications where data can be difficult to obtain.

Employing meticulous curation strategies detailed in Chapter 3 to compile a high-quality COVID-19 dataset and ensuring it is free from shortcuts as demonstrated in Chapter 4, this chapter

illustrates that a high-performing model can be developed to generalize effectively to external data. To provide further evidence and proof-of-concept, these strategies were applied in the development and training of a model for the MIDRC COVIDx Challenge. In this contest, participating teams aimed to train an AI model to differentiate between COVID-19 negative and positive patients using frontal-view portable chest radiographs.

Utilizing the strategies described in this thesis, Ran Zhang, Ph.D., and the author submitted a model trained with the training strategies outlined in Appendix A. Although MIDRC supplied its own chest radiograph dataset, our model primarily relied on the HF dataset, with only a small selection of MIDRC cases. Remarkably, this submission achieved the highest performance in the competition, proving that a model developed using a high-quality, single-source dataset can generalize even better than models designed with the test data. This outcome further validates the effectiveness of the strategies presented in this work.





We want to thank SPIE greatly for sponsoring MIDRC's first Grand Challenge, COVIDx Challenge

This successful event kicked off in August and wrapped up in November. The winners were acknowledged in Chicago during the RSNA Annual Meeting 2022. SPIE's generous support allowed us to offer prizes to our top three winners, and their code will be made publically available to further COVID research through the MIDRC GitHub site.

MIDRC





MEDICAL IMAGING AND DATA RESOURCE CENTER.

MIDRC's first Grand Challenge, the COVIDx Challenge, a COVID classification Challenge performed using MIDRC data, concluded in November. We'd like to recognize the following top-ranked finishers:

#### **FIRST PLACE**

RAN ZHANG, PHD

**DALTON GRINER, MS** 

**GUANG-HONG CHEN, PHD** 

(University of Wisconsin) Score - Area under the ROC curve, or AUC, of 0.703

#### **SECOND PLACE**

#### **MATHIEU GOULET, PHD**

(Centre régional intégré de cancérologie (CRIC), LéViS, QC Canada) Score - AUC of 0.701

#### THIRD PLACE

FINN BEHRENDT

Institute of Medical Technology and Intelligent Systems,Hamburg, Germany Score - AUC of 0.678

Acknowledgement - the Computational Biomarker Imaging Group (CBIG) at the University of Pennsylvania: Chunrui Zou, PhD, and Despina Kontos, PhD

**Figure 5.10:** Utilizing the strategies described in this thesis, Ran Zhang, Ph.D., and the author submitted a model to the MIDRC COVIDx Challenge. Remarkably, this submission achieved the highest performance in the competition, proving that a model developed using a high-quality, single-source dataset can generalize even better than models designed with the test data. This outcome further validates the effectiveness of the strategies presented in this work.



## Chapter 6

# Decoding AI Decisions: Employing Saliency Methods to Illuminate Model Predictions

The ancient board game of Go, invented in China over 3,000 years ago, continues to captivate players to this day. In this strategic game, opponents take turns placing stones on a board with the objective of surrounding and capturing their rival's stones or skillfully creating territorial spaces. The player with the highest combined score of stones and territories wins the game. Despite its seemingly simple rules, Go boasts an astounding complexity, with a standard board presenting  $10^{170}$  possible configurations—far more than the grains of sand on every beach or even the number of atoms in the entire universe. This complexity puts Go in a league of its own, surpassing even chess by over a googol times.

Given its intricacy, Go presented a formidable challenge for AI, and for decades, even the most advanced computers struggled to compete with amateur players. However, in March 2016, the deep learning algorithm AlphaGo<sup>47</sup> astonished the world by defeating the legendary Go player Lee Sedol 4-1 in a match watched by 200 million people. Within less than a decade, AI progressed from an amateur dan level to outperforming the world's greatest player. Similarly, AI-based algorithms in the ImageNet challenge witnessed a drop in the average error rate from 25% in 2010 to surpassing human ability by 2015, thanks to advances in CNNs. Deep learning is rapidly approaching human-level performance in numerous tasks once deemed impossible or at least decades away.

However, behind these amazing achievements is an uglier side to deep learning. When Microsoft

launched its AI chatbot Tay in 2016, it could not have anticipated the ensuing disaster. Within hours of its release, Tay began posting offensive and inflammatory tweets, leading to its shutdown just 16 hours after launch. Another example is Amazon's Rekognition, a cloud-based computer vision platform released in 2016. By 2017, the Washington County, Oregon Sheriff's Office was using Rekognition for the facial identification of suspects. Unfortunately, the American Civil Liberties Union soon revealed that Rekognition had falsely matched several members of Congress, particularly those of color, with mugshot photos. Subsequent studies<sup>171</sup> highlighted Rekognition's biased performance, demonstrating its proficiency with male faces but significantly poorer results for darker-skinned female faces.

Deep learning models, with their inherent complexity and opaque decision-making processes, are often viewed as "black boxes." The limited understanding of their inner workings can lead to unexpected and, at times, disastrous outcomes. This has resulted in a growing sense of caution and a lack of trust in deep learning models, particularly in fields like medicine. While errors may be tolerable when predicting text or making a move in a game, mistakes involving human lives can have severe consequences.

In response to the rapid advancement of deep learning, there has been a surge of interest in "explainable AI" (XAI).<sup>172</sup> XAI algorithms adhere to three core principles: transparency, interpretability, and explainability. However, despite the growing interest in this field, a rigorous and universally accepted definition for each of these concepts has yet to be defined.<sup>173</sup>

Model interpretability generally refers to the capacity to understand and articulate the reasoning behind an AI system's predictions and decisions. Numerous methods have been developed to improve explainability and provide insights into the complex inner workings of these advanced models. This chapter explores several such methods to gain a deeper understanding of the features utilized by the COVID-19 models developed in this work. This is crucial for identifying and comprehending shortcut features and addressing generalizability concerns. Ultimately, a more profound understanding of the features employed by a model paves the way for a reliable clinical implementation, which is the primary goal of any medical AI algorithm.

## 6.1 An Overview of Common Saliency Methods

Convolutional neural networks are commonly used in the classification of medical images, necessitating the development of methods that explain these models' decisions to establish trust with clinicians. Saliency maps have emerged as a prevalent approach for post hoc interpretability of CNNs, offering visual representations that highlight the most important or relevant regions within input images. As a result, numerous CNN medical imaging studies have employed saliency maps to justify model predictions and provide localization.<sup>69,174,175</sup> However, recent research<sup>174,176–180</sup> reveals that saliency maps may lack robustness, reproducibility, and reliability.

There are many different saliency methods, each with unique strengths and applications. This chapter examines some of the more common methods using models developed in Chapters 4 and 5. The methods fall broadly into two categories: pixel attribution and gradient-only attribution. Pixel attribution techniques assign an importance score or relevance value to each pixel in the input image. These attributions indicate the contribution of individual pixels to the model's prediction. Gradient-only attribution techniques rely on the gradients of the model's output with respect to the input image to determine the importance of each pixel. These methods essentially measure how sensitive the model's prediction is to small changes in the input image.

The following subsections will briefly summarize each algorithm, with visual demonstrations provided by the image marker model from Chapter 4, which was trained on data where a small marker was randomly added to the shoulder region in each COVID-19 positive image (see Figure 4.11). This model was chosen because it effectively demonstrates saliency methods since the key discriminating feature (white marker) is visually evident, and the methods should consistently highlight this feature.

#### 6.1.1 Integrated Gradients

Integrated Gradients (IG)<sup>181</sup> is a technique for attributing a classification model's prediction to its input features. The key idea behind IG is to approximate the integral of the gradients along a straight path from a baseline input to the actual input:

$$IG_i(x) := (x_i - x_i') \times \int_{\alpha=0}^1 \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$
(6.1)



**Figure 6.1:** Integrated Gradients is a technique for attributing a classification model's prediction to its input features. Example heatmaps for the marker-trained model with a positive case (top) with the inserted marker circled and a negative case (bottom) without the added marker. The model correctly predicted each class for these cases, and IG has highlighted the marker in the positive image following the intuition that this is the key feature.

where  $\alpha$  is the scaling coefficient. The baseline input is often an image with no discernible features or content, such as a black image. The method computes the gradients at different points along the path and integrates them to determine the importance of each feature in the input image. This results in an attribution map that highlights the most influential regions for the model's prediction.

### 6.1.2 Gradient SHAP

Gradient SHAP<sup>182</sup> is an interpretability method that combines the principles of IG and Shapley values to provide explanations for deep learning models, especially in image classification tasks. Gradient SHAP adds Gaussian noise to each input sample multiple times, selects a random point along the path between baseline and input, and computes the gradient of outputs with respect to


**Figure 6.2:** Gradient SHAP is an interpretability method that combines the principles of IG and Shapley values to provide explanations for deep learning models. The method highlights the marker as a key feature in the positive case. The baselines used were a random sample of negative images that did not contain the added marker. Using these baselines, Gradient SHAP identifies the absence of a marker as a key feature, which is circled in the negative example (bottom).

those selected random points. The final SHAP values represent the expected value of gradients \* (inputs - baselines).

### 6.1.3 Deep Learning Important FeaTures

Deep Learning Important FeaTures (DeepLIFT)<sup>183</sup> aims to provide meaningful attributions for each input feature by comparing the activations of each neuron to a reference or baseline activation, highlighting the most influential features that contribute to the model's prediction. DeepLIFT computes the contribution scores by backpropagating the differences between the neuron activations and their reference activations through the network. This process considers the non-linearities in the network while calculating the importance scores, resulting in a more accurate representation of



**Figure 6.3:** DeepLIFT is a back-propagation-based approach that attributes a change to inputs based on the differences between the inputs and corresponding baselines for non-linear activations. The method highlights the marker as a key feature in the positive case. The baselines used were a random sample of negative images that did not contain the added marker. No definitive conclusion can be drawn from the negative case, but the shoulder regions are highlighted, suggesting that features in this region are important.

the feature importance.

### 6.1.4 Occlusion

Occlusion<sup>55</sup> is an interpretability method used to understand and visualize the importance of regions within an input image for deep learning models, particularly in image classification tasks. The method is based on the idea of systematically occluding or masking different parts of the image and observing the impact on the model's prediction. By doing so, the occlusion method helps identify the most influential regions that contribute to the model's decision.

The occlusion technique involves using a sliding window or patch to mask different parts of the input image. For each occluded version of the input image, the model's prediction is recorded.



**Figure 6.4:** Occlusion is an interpretability method used to understand and visualize the importance of regions within an input image for deep learning models, particularly in image classification tasks. The method is based on the idea of systematically occluding or masking different parts of the image and observing the impact on the model's prediction. The marker is highlighted as a key feature in the positive case, suggesting that when it is occluded, the prediction changes drastically. For the negative case, it is harder to make a definite conclusion.

The change in the model's prediction compared to the original, non-occluded image indicates the importance of the occluded region. The differences in prediction scores for each occluded version of the image are combined to create a heatmap or importance map. This heatmap highlights the regions that have the most significant impact on the model's prediction, providing insight into the model's decision-making process.

## 6.1.5 Gradient-Weighted Class Activation Mapping

Gradient-weighted Class Activation Mapping (Grad-CAM)<sup>77</sup> is an interpretability method for visualizing the importance of regions within an input image for deep learning models, especially



**Figure 6.5:** Grad-CAM generates a coarse heatmap (corresponding to the output size of the target layer) that highlights the areas in the input image that contribute the most to the model's prediction. The heatmap is generally upsampled to the original image size. The method highlights the marker as a key feature in the positive case; however, note that the original heatmap size is  $7 \times 7$ , so the localization is severely limited. The method also highlights the shoulder region in the negative case suggesting that this region was important to the model's prediction, which follows the intuition that the marker is the discriminating feature between the classes.

CNNs in image classification tasks. GradCAM computes the gradients of the target output with respect to the given layer, averages for each output channel, and multiplies the average gradient for each channel by the layer activations. The results are summed over all channels, and a ReLU is applied to the output, returning only non-negative attributions. Grad-CAM generates a coarse heatmap (usually upsampled to the original image size) that highlights the areas in the input image that contribute the most to the model's prediction, offering insights into the decision-making process of the model.



**Figure 6.6:** Guided Grad-CAM is a combination of two interpretability methods, Grad-CAM and Guided Backpropagation to address the localization issues with Grad-CAM. The method highlights the marker with much more accuracy as a key feature in the positive case and also highlights the shoulder regions in the negative case suggesting that this region was important to the model's prediction, which follows intuition.

### 6.1.6 Guided Grad-CAM

Guided Grad-CAM is a combination of two interpretability methods, Grad-CAM and Guided Backpropagation.<sup>77</sup> This hybrid method aims to provide more visually interpretable and detailed explanations for deep learning models, particularly CNNs, in image classification tasks. Guided Grad-CAM generates a higher-resolution and finer-grained heatmap compared to Grad-CAM alone, offering improved insights into the decision-making process of the model. Guided GradCAM computes the element-wise product of guided backpropagation attributions with upsampled (layer) GradCAM attributions. Attributions are computed with respect to a given layer, and attributions are upsampled to match the input size.



**Figure 6.7:** Even in a best-case scenario with a highly conspicuous class feature, saliency methods can fail to provide any explanation for certain predictions. In the first case (top), the model misclassifies the image despite the apparent marker. Gradient SHAP identifies the marker as a key feature, but it does not support the model's prediction. Furthermore, Grad-CAM disagrees, emphasizing the opposite shoulder region instead. The second case (bottom) also results in misclassification, with both Guided Grad-CAM and DeepLIFT highlighting seemingly irrelevant features.

### 6.1.7 Counterexamples

Before moving forward to analyze additional models used in this work, a couple of points should be addressed. First, the marker model serves as an ideal case for demonstrating saliency. As previously mentioned, the key discriminating feature is easily discernible by human visual inspection, allowing for the establishment of a "ground truth." In these examples, the highlighted feature is already known and well-understood, which is often not the case in real-world datasets. Identifying key features can be challenging without prior knowledge, and even when a plausible explanation is available, it can lead to confirmation bias, where results are interpreted to confirm pre-existing beliefs. To illustrate this, Figure 6.7 presents some counterexamples. In the first case (top), the model misclassifies the image despite the apparent marker (at least from a qualitative perspective). Gradient SHAP identifies the marker as a key feature, but it does not support the model's prediction. Furthermore, Grad-CAM disagrees, emphasizing the opposite shoulder region instead. The second case (bottom) also results in misclassification, with both Guided Grad-CAM and DeepLIFT highlighting seemingly irrelevant features. Even in a best-case scenario with a highly conspicuous class feature, these methods can fail to provide saliency for certain predictions. Recognizing this is crucial to avoid selection and choice-supportive bias when using these techniques.

### 6.2 Analysis of Trained Models Using Saliency

In this section, some of the models described in Chapters 4 and 5 will be analyzed using the saliency methods previously described. The dataset metadata shortcut models described in Chapter 4 (see Table 4.1), including patient age (>70/<50), patient sex (male/female), patient race (white/black), view position (AP/PA), imaging modality (DX/CR), and imaging vendor (Carestream/Konica Minolta) are shown in Appendix C. While it is important to have a sufficient sample size to draw meaningful conclusions, due to the length constraints of this work, 20 randomly sampled cases for each class (disease + shortcut feature) are shown.

### 6.2.1 BIMCV+/HF- Model

To illustrate that a network can exploit the dataset source as a shortcut, a model was trained on the BIMCV+/HF- dataset, as described in Chapter 4. This model exhibited strong shortcut learning, achieving near-perfect accuracy on the hold-out test set. However, its performance dropped to 0.50 on an external test set (refer to Figure 4.5). To further emphasize the potential for shortcuts, another model was trained on the same BIMCV+/HF- dataset, with the central 70% of each image masked. Despite this, the model still reached near-perfect accuracy (see Figure 4.6). Lastly, a separate model was trained on the same dataset, employing segmentation and histogram equalization techniques to counteract potential shortcuts. However, even with these preprocessing measures, the model continued to display strong shortcut learning (see Figure 4.8).

Example heatmaps for the saliency methods described in the previous section are shown in







**Figure 6.8:** Example heatmaps for the saliency methods described in this chapter are shown for each of the three BIMCV+/HF- models outlined in Chapter 4. The saliency maps for the top two panels indicate that shoulder markers may serve as a shortcut in the BIMCV+/HF- training dataset. However, when these features are removed through segmentation, shortcut learning still persists, and the saliency maps provide no immediate explanation for what the shortcut may be. This example demonstrates the dangers of expectation bias, where information is interpreted to confirm prior beliefs.

Figure 6.8 for each of the three BIMCV+/HF- models. In the full images (top panel), the majority of saliency techniques focus on the shoulder regions. For the HF negative case (bottom image in each panel), most methods emphasize the black box concealing patient data. In the BIMCV positive case (top image in each panel), shoulder regions and markers are also accentuated, indicating their significance. This observation may be attributed to distinct features within the BIMCV and HF datasets, such as unique markers in the top corners of the BIMCV dataset (see Figure 3.12) and the darker top corners in the HF dataset due to de-identification boxes (see Figure 3.28). It is plausible that the network leverages these disparities as shortcut features, as suggested by the saliency maps.

The next model under examination is the center-masked model (middle panel), in which the image center is zeroed out, leaving only the edge content. Given the hypothesis that the network relies on specific shoulder markers for shortcut learning, the top corners are expected to be crucial areas highlighted by the saliency maps. Figure 6.8 provides examples that seem to corroborate this hypothesis, as the methods emphasize the shoulder regions. This also accounts for why masking the image did not alter performance; since the shortcut features are located in the corners, the model does not utilize the content in the image's center.

The final model considered involves images preprocessed with histogram equalization and lung tissue segmentation (bottom panel). The shortcut feature hypothesis, based on marker differences in the datasets, is supported by observational dataset discrepancies discussed in Chapter 3 and the saliency methods for the previous models. Intuitively, eliminating these image areas should block shortcuts and compel the model to learn generalizable lung features. However, as demonstrated in Chapter 4, this is not the case. While histogram normalization and segmentation marginally reduced internal performance, the model's generalization remained comparable to the first two shortcut models. Despite the removal of extrinsic markers, this model still exploits a shortcut, and the saliency maps do not readily reveal the nature of this shortcut.

In truth, this analysis was a carefully orchestrated ruse intended to illustrate expectation bias. The example images were deliberately selected based on the dataset differences observed in Chapter 3. Additional example images are presented in Appendix C, and for each example supporting the shoulder marker shortcut observation, another seems to contradict it. This is not to deny the potential role of shoulder markers as shortcuts but rather to emphasize that the evidence provided by the saliency maps is insufficient to substantiate this claim. It is relatively easy to accept that the

given example maps support the shoulder marker theory, given prior knowledge of this potential shortcut. However, in contrast, drawing conclusions from the saliency maps for the segmented model, where no clear explanation exists, proves to be more challenging. A common limitation of these saliency methods is the lack of rigorous metrics to gauge the accuracy or reliability of their maps,<sup>179,184</sup> which often leads observers to make "mountains out of molehills."

### 6.2.2 Contrast and Sharpness Shortcut Models

Besides extrinsic shortcut features like image markers or patient anatomy, Chapter 4 also demonstrated that intrinsic features such as image contrast or sharpness could serve as shortcut features. Unfortunately, these shortcuts are imperceptible to the naked eye, making their detection through visual inspection of the images nearly impossible. In situations like this, can saliency methods offer any insight into shortcut features? To explore this question, the contrast and shortcut models developed in Chapter 4 were assessed using the described saliency methods.

As depicted in Figure 6.9, drawing conclusions from these heatmaps is challenging (additional examples can be found in Appendix C). This aligns with the notion that the added contrast and sharpness are not only nearly invisible but also globally distributed, making it difficult to imagine how they could be visually emphasized. One advantage of employing a method like the shortcut detectives introduced in Chapter 4 is that they do not depend on visual representation but provide quantitative results that are easier to interpret. While saliency methods can be useful for extrinsic features such as image markers or breast tissue, their effectiveness is limited when features are not easily visualized, as demonstrated in these examples.

### 6.2.3 COVIDx Model

This thesis, along with other studies, has demonstrated that models trained on the COVIDx dataset exhibit exceptional performance on internal test sets, but their accuracy plummets to random chance when applied to external datasets.<sup>85,106–108</sup> This finding underscores the crucial role of dataset quality, which, in many instances, is more important than data quantity. The COVIDx dataset is predisposed to shortcut learning due to the fact that most negative cases stem from a pre-pandemic dataset, while positive cases originate from various sources with differing quality levels. As the two



**Figure 6.9:** Example heatmaps for the sharpness and contrast shortcut models developed in Chapter 4. Drawing conclusions from these heatmaps is challenging because the added contrast and sharpness are not only nearly invisible but also globally distributed. One advantage of employing a method like the shortcut detectives introduced in Chapter 4 is that they do not depend on visual representation but provide quantitative results that are easier to interpret.

classes derive from distinct distributions, it becomes relatively easy for a model to discern differences between dataset sources rather than disease features. Consequently, these dataset-specific features act as shortcuts, enabling the model to bypass any disease-related characteristics.

Chapter 4 revealed that sharpness and contrast correlations exist in the disease classes, among other potential shortcut features. To explore which shortcut features are learned from the COVIDx dataset, the trained model discussed in Chapter 4 (refer to Table 4.6) was assessed using the saliency methods outlined.

Figure 6.10 showcases examples where image markers are accentuated by the saliency maps. It is well-documented that these markers can serve as potent shortcut features,<sup>83,87</sup> especially in cases where positive and negative samples originate from distinct dataset sources. Additional examples



**Figure 6.10:** The COVIDx dataset is predisposed to shortcut learning due to the fact that most negative cases stem from a pre-pandemic dataset, while positive cases originate from various sources with differing quality levels. In these examples, image markers are accentuated by the saliency maps suggesting these may serve as potent shortcut features.

can be found in Appendix C. In these instances, saliency methods may prove more valuable for identifying potential shortcut features, as they are more visually discernible.

### 6.2.4 Generalizable HF Model

Finally, the HF-trained models in Chapter 5 will be examined using the saliency methods presented in this chapter. Unlike the models previously discussed, the HF-trained models exhibit extensive generalizability across multiple external datasets (refer to Table 5.1). This can be partly attributed to the high-quality HF training dataset. Since both COVID-19 positive and negative cases were obtained from a single source, the HF dataset avoids the data heterogeneity observed in other datasets, such as COVIDx. Nonetheless, the HF dataset features a wide range of patient and imaging characteristics, enabling a deep-learning network to learn diverse features. The size of the HF dataset also mitigates overfitting issues commonly encountered in smaller medical datasets.

While this model fulfills one objective of this work—creating an accurate and generalizable model for COVID-19 classification—it, like all deep learning models, lacks transparency. Techniques such as ensemble learning (see Appendix A) and varying model architectures help minimize the variability of model predictions, but the precise features extracted or their utilization remains unclear. To address this, the HF-trained models were also studied using the saliency methods introduced in this chapter.

As detailed in Appendix A, an ensemble training scheme<sup>185–187</sup> was used for the HF-trained



**Figure 6.11:** Example heatmaps for the five separate models in the HF-trained ensemble developed in Chapter 5. The saliency maps are consistent across each model, suggesting that the same features are learned by every individual model. Assuming this dataset is free of shortcuts, these results imply that the models are learning genuine disease features, as each model selects identical image features to accurately predict COVID-19 classification.

models. This approach is based on the notion that a diverse group of models can make superior collective decisions compared to a single model. Merging predictions from multiple models can reduce the likelihood of errors arising from individual model limitations or biases, often resulting in enhanced performance. It is, therefore, intriguing to examine the features each model relies on for predictions.

Figure 6.11 displays the saliency maps for the five separate models in the HF-trained ensemble (additional examples in Appendix C). The saliency maps are consistent across each model, suggesting that the same features are learned by every individual model. Assuming this dataset is free of



**Figure 6.12:** Example heatmaps for the HF-trained DenseNet-121, EfficientNet, and Swin Transformer models. The saliency maps remain consistent across each network architecture, implying that the models learn the same features.

shortcuts, these results imply that the models are learning genuine disease features, as each model selects identical image features to predict COVID-19 classification accurately. The stability of saliency maps represents a positive step toward model transparency.

Furthermore, the saliency maps for different network architectures are compared to ascertain if they extract the same features. Figure 6.12 illustrates the saliency maps for the HF-trained DenseNet-121, EfficientNet, and Swin Transformer models (refer to Appendix A for architecture details). The saliency maps remain consistent across each network architecture, once again implying that the models learn the same features. This observation lends additional support to the notion that the models are acquiring generalizable disease features.

Although this saliency analysis does not conclusively establish the model's interpretability, it reinforces the notion that the generalizable strategy developed in this work consistently selects relevant features. A more comprehensive investigation is necessary to confirm that the strategy relies on pertinent features (e.g., lung opacities), and these saliency methods may aid in demonstrating that the influence of shortcut features is largely minimized.

# 6.3 Alternative Saliency Methods: Beyond Heatmaps in Deep Learning Interpretability

While saliency heatmaps provide valuable insights into the importance of input features for model predictions, they are only some of the many interpretability techniques available. In this section, for the sake of completeness, a short analysis of feature map activation and visualization is presented, which offers additional perspectives on the internal workings of deep learning models, particularly CNNs.

### 6.3.1 Feature Map Activation

Feature map activation is a technique used to visualize and understand the internal workings of a deep learning model, particularly CNNs. This method focuses on the activation of neurons within the model's hidden layers when processing an input, providing insights into the features the model learns and uses for decision-making.

In a CNN, each layer comprises multiple filters, also known as kernels, which are responsible for detecting various features such as edges, textures, or patterns within an input image (refer to Chapter 2). When the model processes an input, these filters convolve with the input image, producing a set of feature maps. Each feature map corresponds to a specific filter and represents the activation of that filter across the input image. An example of the kernels from the final convolutional layer in the HF-trained DenseNet-121 model is shown in Figure 6.13 (right).

To visualize the feature map activations, an example image is processed through the CNN up to the selected layer. The intensity of each pixel in the extracted feature map corresponds to the activation strength at that location. Brighter pixels indicate higher activation, suggesting the presence of the corresponding feature. Also shown in Figure 6.13 (left) are feature map activations from the first, second, third, and fourth denseblocks in the DenseNet-121 model (see Appendix A for more details on model architecture). Generally, earlier layers detect lower-level features (e.g., edges, corners), while later layers capture more complex, abstract patterns (e.g., object parts, shapes). As an image proceeds through the network, its spatial dimensions are reduced as more high-level features (expressed as channels) are extracted, explaining why the early feature maps have higher resolution.



**Figure 6.13:** Example feature map activations (left) and convolutional kernels (right) from the HF-trained DenseNet-121 model. The intensity of each pixel in the extracted feature map corresponds to the activation strength at that location. Brighter pixels indicate higher activation, suggesting the presence of the corresponding feature.

In general, studying kernels and feature map activations can be a daunting task, as modern CNNs often consist of hundreds or even thousands of layers. In the example presented in Figure 6.13, the early feature map activations demonstrate high activation for the image marker. As the feature maps become increasingly abstract, they are more challenging to interpret. One approach to understanding low-level feature maps is through the lens of the Grad-CAM algorithm discussed in the previous section. Essentially, this algorithm takes feature map activations and weights them according to the gradient. An in-depth analysis of feature map activations and convolutional kernels has the potential to offer insights into the model's learned features and the intermediate representations it generates when processing input data.

### 6.3.2 Feature Visualization

Related to feature map activation, feature visualization aims to reveal the features and patterns that the model has learned to recognize during the training process. Visualizing these learned features can provide insights into the model's decision-making process, potentially improving the model's design and interpretability. The process typically involves optimizing input images to maximize the activation of specific neurons or layers within the model. By examining the generated images,



**Figure 6.14:** Feature visualization aims to reveal the features and patterns that the model has learned to recognize during the training process. The process typically involves optimizing input images to maximize the activation of specific neurons or layers within the model. These images typically represent the types of patterns that excite the selected neurons or layers the most.

different types of features and structures that the model has learned to detect can be studied.

Depending on the desired level of granularity, visualization can target individual neurons or entire layers. Early layers generally learn simple features (e.g., edges, textures), while later layers capture more complex, higher-level patterns (e.g., object parts, shapes). Synthetic input images are optimized to maximize the activation of the target neurons or layers. These images typically



**Figure 6.15:** While primarily an artistic tool, DeepDream feature visualization also offers insights into the features and patterns learned by CNNs, contributing to the understanding of their internal representations.

represent the types of patterns that excite the selected neurons or layers the most. Example feature visualization from multiple layers in the HF-trained DenseNet-121 model are shown in Figure 6.14. These images showcase the remarkable ability of modern CNNs to extract intricate patterns and features from input images. Although the visualizations are aesthetically captivating, interpreting them may prove challenging due to their abstract nature.

The HF-trained DenseNet feature maps may also suggest an impact of ImageNet pretraining,

as numerous textures and features resemble those found in ImageNet training (see Figure 2.2.7). Despite the transition from natural images to CXR images, the intricate and high-frequency features carried over from ImageNet pretraining remain present. The usefulness of these features is an open research question, as the persistence of features from another domain might lead to the extraction of irrelevant or potential shortcut features. Investigating the influence of ImageNet pretraining on learned features for CXR is a crucial future study to be conducted.

Finally, as more of a creative endeavor rather than an interpretability method, DeepDream<sup>188</sup> is an algorithm that combines feature visualization and neural style transfer principles. As with feature visualization, an input image is iteratively changed to maximize the activation of the selected layer. However, various regularization techniques are used to ensure the generated image remains visually coherent and to control the level of abstraction. The resulting image is characterized by intricate patterns and surreal motifs from the abstract features learned by the neural network. While primarily an artistic tool, it also offers insights into the features and patterns learned by CNNs, contributing to the understanding of their internal representations. An example of a DeepDream CXR using ImageNet weights is shown in Figure 6.15.

# 6.4 Dimensionality Reduction and Analysis of Extracted Features from a Trained COVID-19 Model

As emphasized earlier, interpreting the inner workings and decision-making processes of increasingly complex deep learning models can be challenging due to the abstract nature of the features. An alternative approach is to extract and analyze the feature vectors from input images. As detailed in Chapter 3, this method involves obtaining feature vectors, which represent the model's learned features at various abstraction levels, and analyzing them to gain insights into the model's decision-making rationale. To further investigate the HF-trained models, feature vectors from test images from the external datasets were examined using UMAP dimensionality reduction and K-Means clustering, as displayed in Figure 6.16 with sample images shown in Figure 6.17.

Figure 6.16 demonstrates that different model architectures, as discussed in Appendix A, embed features in unique ways, resulting in varied shapes within the UMAP dimensionality reduction. This observation is complex yet expected since different architectures influence how features are



**Figure 6.16:** To further explore HF-trained models, feature vectors from test images in external datasets were analyzed using UMAP dimensionality reduction and K-Means clustering. Different model architectures uniquely embed features, leading to diverse UMAP graph shapes. Distinct clusters in the graph represent groups of similar high-dimensional data points, while denser regions imply greater similarity. All models show a dense region of COVID-19 positive points, suggesting that each model extracts features that make these cases alike. EfficientNet and Swin Transformer models exhibit similar UMAP distributions with more diffuse and continuous high-dimensional feature variations. In contrast, the DenseNet model displays tighter clustering, especially for the MIDRC dataset, indicating its tendency to extract distinct characteristics from data.



**Figure 6.17:** Randomly sampled images from the K-means clusters (outline color corresponds to cluster) in Figure 6.16. Despite the absence of perfect separation between COVID-19 labels, all models exhibit consistent groupings, with COVID-19 positive cases generally clustered to the right and negative cases to the left. This outcome aligns with expectations and previous findings, considering that the models did not achieve perfect accuracy, and perfect separation should not be expected. When examining the sample images associated with K-Means clustering labels (border color corresponds to cluster label in Figure 6.16), no discernible patterns in cluster groupings are detected.

extracted. In UMAP, distinct clusters in the graph signify groups of similar data points in highdimensional space, and denser regions suggest a higher degree of similarity among these points. All models exhibit a dense region of COVID-19 positive points, indicating that each model has extracted features that render these cases similar. The EfficientNet and Swin Transformer models display comparable UMAP distributions characterized by more diffuse and continuous variations in high-dimensional features across data points. In contrast, the DenseNet model exhibits tighter clustering, particularly for the MIDRC dataset, suggesting that the DenseNet architecture is more inclined to extract unique characteristics from data.

Although the COVID-19 labels do not show perfect separation, all models reveal consistent groupings, with COVID-19 positive cases typically clustered on the right and negative cases on the left. This finding corresponds with intuition and prior results, as none of the models achieved perfect accuracy, and perfect separation is not anticipated.

When the UMAP graph is labeled according to the dataset source, no evident biases toward specific sources emerge, as indicated by the predominantly uniform distribution of individual dataset labels throughout the graph. Upon inspecting the sample images corresponding to K-Means clustering labels, no apparent trends in cluster groupings are observed. While drawing concrete conclusions remains challenging, the extracted feature graphs exhibit no clear signs of shortcut learning or other anomalies, suggesting that the HF-trained model is likely free of overt indications of bias. Further exploration of these extracted features and the factors influencing them, such as model architecture, remains a task for future research.

### 6.5 Discussion

Deep learning models, particularly in the realm of neural networks, have grown increasingly complex and sophisticated, enabling them to excel in a wide array of tasks. However, this complexity often comes at the cost of interpretability, making it challenging to understand their inner workings and decision-making processes. This lack of transparency can hinder trust in these models, especially when applied to critical domains such as healthcare.

Saliency methods have emerged as a means to analyze and visualize what a network is doing by identifying which input features contribute most significantly to a model's predictions. Nevertheless,

these methods are not without their limitations. As demonstrated in this chapter, they can be unreliable, sensitive to small perturbations, and provide inconsistent results across different techniques. Moreover, interpreting saliency maps can be subjective and challenging, as they may not always provide clear explanations for a model's behavior. Consequently, the quest for more robust, reliable, and interpretable approaches to better understand and trust deep learning models remains an active area of research in the AI community.

Heatmap methods hold appeal as they offer visual representations of network predictions, enabling users to identify influential factors in images. However, certain underlying features, such as contrast, sharpness, or high-frequency elements, can be challenging to discern or represent visually. In these cases, visual maps may prove inadequate, while a quantitative metric could be more informative. The shortcut detective framework serves as a complementary saliency method by providing an interpretable quantitative metric (AUC) to assess a network's sensitivity to specific features. By supplementing existing saliency methods, the shortcut detective framework can contribute to the development of trustworthy and robust deep learning models, bridging the gap between visual insights and quantitative evaluation.

# Chapter 7

# **Conclusions, Limitations, and Future Directions**

This thesis presents a general strategy to develop and optimize an accurate and generalizable deep learning strategy for COVID-19 classification in chest X-ray radiography. This strategy focused on the major themes of initial dataset curation and analysis, identification of shortcut features to enhance dataset quality, and the employment of training strategies for a generalizable deep learning algorithm using a high-quality dataset. Furthermore, the interpretability of the deep learning framework was scrutinized. The workflow for this process is illustrated in Figure 7.1.

Although the primary objectives of this work have been largely achieved, certain limitations and potential avenues for future research are discussed.



**Figure 7.1:** A general strategy to develop and optimize an accurate and generalizable deep learning strategy for COVID-19 classification in chest X-ray radiography. This strategy focused on the major themes of initial dataset curation and analysis, identification of shortcut features to enhance dataset quality, and the employment of training strategies for a generalizable deep learning algorithm using a high-quality dataset.

### 7.1 Conclusions

### 7.1.1 Dataset Curation and Quality Analysis

In Chapter 3, the crucial role of data quality and diversity for AI was emphasized, underscoring the necessity for meticulous dataset curation and analysis. The significance of high-quality data cannot be overstated, as it is just as crucial for AI success as model architecture and training strategies. Comprehensive data curation extends beyond the mere amalgamation of data, requiring in-depth examination of available information by domain experts. In the context of COVID-19 CXR, patient metadata served as an indispensable resource for understanding data quality and mitigating bias in this work. A thorough assessment of relevant metadata for each COVID-19 dataset was performed, enabling the identification of potential biases. Subsequently, image enalyses, including contrast, edge, and high-frequency content, were conducted. Lastly, image features and outliers were scrutinized using a deep learning network, as well as various dimensionality reduction and clustering techniques. Excluding the COVID dataset, these evaluations confirmed that the COVID-19 datasets employed in this study were of adequate quality while exhibiting substantial diversity to achieve the objectives of training and validating a deep learning algorithm for COVID-19 classification.

This work illustrates the primacy of "good data" over "big data" in achieving optimal results. By addressing potential biases and promoting the development of more accurate and robust AI models, the full potential of AI can be harnessed to tackle complex real-world challenges and devise more equitable and inclusive solutions. Nonetheless, as seen in Chapter 4 even well-curated datasets may contain subtle biases, necessitating further analysis for detection.

#### 7.1.2 Shortcut Feature Detection

Deep learning has made significant strides in object detection, image classification, and natural language processing over the past ten years, largely due to its ability to learn intricate features from data. Despite its exceptional performance on benchmark datasets, it is becoming increasingly clear that there are constraints and practical difficulties when implementing these models in real-world situations. One critical problem is weak generalizability, where performance significantly drops when models are applied to external datasets. This obstructs the adoption and implementation

of deep learning models in high-stakes areas, such as healthcare applications. The COVID-19 pandemic poignantly emphasized this issue, as numerous machine learning models were created, but very few showed strong performance in real-world clinical tests or had any meaningful clinical impact.

The concept of shortcut learning has surfaced as a recent focus in deep learning research. Studies have revealed that poor model generalizability can stem from shortcut learning when training datasets include unnoticed shortcuts, which are spurious correlations between irrelevant image features and their corresponding training labels. As a result, models rapidly gravitate toward these spurious correlations instead of the intended image features, establishing inaccurate connections between input image data and output labels. Medical imaging is especially prone to shortcut learning due to limited data availability. To tackle this problem, smaller datasets are often merged with other sources to maximize training size for deep learning. However, differing characteristics among data sources can cause networks to learn distinctions between datasets rather than the desired features for classification tasks.

To illustrate this point, Chapter 4 emphasizes that almost any population or imaging characteristic can become a powerful shortcut if correlated with the classification label. Alarmingly, these shortcuts might lack any visual explanation, making them unidentifiable through qualitative assessment or model saliency techniques. To address this issue, a methodical approach for training and validating shortcut detective models was devised. These shortcut detectives further ensure data quality by pinpointing potential shortcuts in a dataset. If intrinsic shortcuts are suspected within a dataset, the general framework proposed in this work can be employed to develop similar shortcut detectives and identify the purported intrinsic shortcuts.

A crucial takeaway from this chapter is the old adage, "if something seems too good to be true, it probably is." Shortcut learning can enable networks to achieve exceptional, even near-perfect accuracy, but such a metric is meaningless if the model lacks generalizability. When developing a deep learning model, it is vital to properly assess its performance. As demonstrated in this work, using a hold-out test set alone is insufficient to accurately measure a model's true performance.

In rock climbing, there is a term called "soft for the grade," which implies that a climb is easier than its assigned difficulty level suggests. Many climbers seek out these "soft" climbs to boost their perceived ability, much like how some models have been published with results based on a hold-out test set, which proved too "soft" and did not reflect their true performance. Chapter 5 delves into the challenge of evaluating network performance and generalizability.

### 7.1.3 Training and Evaluation of a Generalizable Network

Generalizability is arguably the most crucial aspect of a deep learning solution, yet it remains a persistent challenge, particularly in the field of medicine. The ability of a model to generalize beyond its training data is essential for its practicality and applicability in real-world situations, such as medical diagnostics and treatment planning. Despite significant advancements in the field, the current limitations of deep learning models in generalizing and adapting to new data hinder their widespread adoption in the medical sector. Nevertheless, deep learning continues to be a powerful problem-solving tool with numerous successes under its belt.

In Chapter 5, generalizability was investigated in relation to dataset size and quality. To address the common issue of limited data in medical imaging and the risk of shortcut learning when merging datasets, a model was developed using a single data source, showcasing its capacity to generalize to real-world external data. Challenging the popular notion that diverse data sources are necessary for highly generalizable AI models, this work demonstrated that, through careful data curation, it is possible to train AI models with strong generalizability using just one data source. The model's generalizability was assessed across various patient and imaging demographics, validating its consistency in diverse settings.

Additionally, generalizability was explored for different training dataset sizes, revealing that even a small dataset comprising a few hundred images can attain a robust baseline performance. This finding was found to depend on pretraining schemes, which were examined and shown to mitigate the challenges associated with small datasets. Furthermore, pretraining with data related to the target data led to improved performance and generalization, highlighting the potential of meticulously curated data and pretraining strategies in tackling the generalization challenge in deep learning. This result is particularly pertinent for medical imaging applications where data can be hard to come by.

The majority of COVID-19 models were published but remained confined to charts and figures in research papers. They didn't identify potential patients in real clinics or aid radiologists in diagnosis. For a deep learning solution to be clinically viable, it must exhibit generalizability, enabling it to be deployed effectively with new data. In addition to generalizability, model interpretability is also crucial, particularly in the medical field, where unanticipated outcomes can have significant consequences for patients. However, a model's generalizability does not guarantee its interpretability, a topic explored in Chapter 6.

The complexity and obscure decision-making processes of deep learning models often lead to them being considered "black boxes." This limited comprehension of their inner mechanisms can result in unforeseen and sometimes catastrophic outcomes, causing increased wariness and distrust in deep learning models, especially in areas like medicine. While errors may be acceptable when predicting text or making game moves, mistakes involving human lives can result in severe consequences.

The ability to comprehend and convey the rationale behind an AI system's predictions and decisions is generally referred to as model interpretability. Various techniques have been devised to enhance explainability and shed light on the intricate inner mechanics of these sophisticated models. In Chapter 6, several saliency approaches were examined using shortcut models as well as the generalizable HF-trained model created in Chapter 5.

While these saliency approaches are attractive, especially in medical imaging, due to their visual representation of network predictions, they also have certain drawbacks. As shown in Chapter 6, they can generate inconsistent outcomes across different methods, and interpreting saliency maps can be subjective and difficult, as they may not consistently offer lucid explanations for a model's actions. Additionally, some inherent characteristics, such as contrast, sharpness, or high-frequency components, may be hard to detect or visually portray. In these instances, visual maps might be insufficient, and a quantitative measure could provide more valuable information. The shortcut detective framework developed in this work complements saliency techniques by delivering an interpretable quantitative metric (AUC) to evaluate a network's responsiveness to particular features. By supplementing existing saliency approaches, the shortcut detective framework aids in creating reliable and resilient deep learning models, bridging the gap between visual understanding and quantitative assessment.

### 7.2 Limitations and Future Work

There are several limitations to address in this thesis work. First, as discussed in Chapter 2, COVID-19 diagnosis is determined by PCR testing, and CXR is generally not recommended for COVID-19 diagnosis. Although AI models are not suitable as standalone diagnostic tools for COVID-19, their clinical value lies in optimizing radiologists' clinical workflows by prioritizing reading lists in clinics. Deep learning networks can rapidly learn the unique features of a disease, assisting in screening patients suspected of infection to curb disease spread during a pandemic – a situation where the clinical utility of this work is most apparent. However, evaluating their value requires deploying models in the actual clinical workflow, which remains a future task of this work.

While the COVID-19 pandemic has largely subsided due to vaccine development and administration, COVID-19 is likely to remain a critical differential diagnosis for individuals presenting flu-like symptoms, for whom CXR is commonly prescribed. More crucial than COVID-19 detection in this work is the establishment of a general framework for developing deep-learning classification solutions. The pandemic served as a valuable learning opportunity, revealing the challenges AI faces in medicine. This thesis aims to address these issues so that when the next crisis arises, common mistakes can be avoided, enabling the swift development of AI solutions to provide aid.

One potential limitation to consider is the evolving virulence of the SARS-CoV-2 virus during the pandemic, as well as the majority of the population gaining some degree of immunity through vaccination or prior infection. Although beyond the scope of this work, further investigation is necessary to explore the impact of COVID-19 pneumonia's changing nature on the AI model presented in this work.

Another constraint is the absence of a comparison between the model's performance and that of human radiologists. Since diagnosing COVID-19 from chest radiographs was not a routine clinical task for radiologists, such a comparison holds limited clinical significance. Nevertheless, radiologists could contribute to quantifying the accuracy of the saliency methods detailed in Chapter 6, thereby enhancing the understanding of network predictions and improving model interpretability.

Furthermore, while Chapter 4 devised a method for detecting shortcut features, addressing them once identified remains a topic for future research. Finally, deep learning development has many parameters, hyperparameters, or "knobs to turn," so to speak. While the methods presented in this work have been optimized to a certain extent, there is still ample room for refining the parameters and techniques further.

ImageNet pretraining has become a standard practice for numerous computer vision tasks involving CNNs. Although this initialization method has outperformed random initialization in many instances, its optimality for CXR or other medical imaging tasks requires further investigation. Medical images differ significantly from ImageNet's natural images, not only because they are typically grayscale rather than RGB but also due to their tomographic or projection-based nature, which affects the features present. As demonstrated in Chapter 6, many ImageNet-learned features persist throughout training, which may not be ideal for CXR image analysis. Given the availability of large CXR datasets (although smaller than ImageNet), training on these datasets to establish pretrained weights could potentially enhance performance and minimize shortcut learning compared to ImageNet-based pretraining. Further research is needed to validate this hypothesis.

# Appendix A

# Deep Neural Network Architecture, Parameters, and Training Strategies

In this appendix the various network architectures, parameters and general training strategies used in this thesis are outlined. Models were developed and trained using the PyTorch<sup>189</sup> platform.

## A.1 Network Architecture

In this work a variety of well-established models were employed to ensure the generalizability of results across different CNN architectures. ResNet and EfficientNet were utilized in Chapter 3 for extracting features from COVID-19 datasets. In Chapter 4, VGG, DenseNet, EfficientNet, Swin Transformer, and ConvNeXt were employed as shortcut detectors to analyze datasets for the presence of intrinsic shortcut features and to study the effect of network architecture on shortcut learning. These architectures were also applied in Chapter 5 to showcase generalizability and investigate the influence of input image size on performance. Lastly, Chapter 6 examined the trained networks using various saliency methods to study the image features they extracted. The following sections provide a brief overview of these architectures, emphasizing their key features.

Throughout this work, the default network parameters (e.g., kernel size and stride, batch normalization, activation) were employed for each model. These default parameters have been established to work effectively for a wide range of computer vision tasks, and further optimization was beyond the scope of this research.

### A.1.1 VGG-16

VGG-16 is a deep convolutional neural network architecture introduced by the Visual Geometry Group at the University of Oxford in 2014.<sup>132</sup> It was designed to perform image recognition and classification tasks and achieved top results in the ImageNet Large Scale Visual Recognition Challenge that year.

Key features:

- 1. 16 weight layers comprised of 13 convolutional layers and 3 fully connected layers.
- 2. 3x3 pixel convolutional filters.
- 3. Multiple convolutional layers stacked before introducing a max-pooling layer.
- 4. Contains three fully connected layers at the end of the network, which are responsible for combining the features learned by the convolutional layers to perform classification tasks.



Figure A.1: VGG architecture. Image from https://neurohive.io/en/popular-networks/vgg16/.

Layer (type:depth-idx)	Output Shape	 Param #
VGG -Sequential: 1-1 L-Conv2d: 2-1 L-ReLU: 2-2 L-Conv2d: 2-3 L-ReLU: 2-4 L-MaxPool2d: 2-5 L-Conv2d: 2-6 L-ReLU: 2-7 L-Conv2d: 2-8 L-ReLU: 2-9 L-MaxPool2d: 2-10 L-Conv2d: 2-11 L-ReLU: 2-12 L-Conv2d: 2-13 L-ReLU: 2-14 L-Conv2d: 2-15 L-ReLU: 2-16 L-MaxPool2d: 2-17 L-Conv2d: 2-18 L-ReLU: 2-19 L-Conv2d: 2-20 L-ReLU: 2-20 L-ReLU: 2-21 L-Conv2d: 2-22 L-ReLU: 2-23 L-MaxPool2d: 2-24 L-Conv2d: 2-25 L-ReLU: 2-26 L-Conv2d: 2-27 L-ReLU: 2-28 L-Conv2d: 2-29 L-ReLU: 2-30 L-MaxPool2d: 2-31 -AdaptiveAvgPool2d: 1-2 -Sequential: 1-3 L-Linear: 2-32 L-ReLU: 2-36 L-Dropout: 2-37 L-relU: 2-37 L	$ \begin{bmatrix} 64, 1000 \end{bmatrix} \\ \begin{bmatrix} 64, 512, 7, 7 \end{bmatrix} \\ \begin{bmatrix} 64, 64, 224, 224 \end{bmatrix} \\ \begin{bmatrix} 64, 64, 112, 112 \end{bmatrix} \\ \begin{bmatrix} 64, 128, 56, 56 \end{bmatrix} \\ \begin{bmatrix} 64, 256, 56, 56 \end{bmatrix} \\ \begin{bmatrix} 64, 512, 28, 28 \end{bmatrix} \\ \begin{bmatrix} 64, 512, 14, 14 \end{bmatrix} \\ \\ \\ \begin{bmatrix} 64, 512, 14, 14 \end{bmatrix} \\ \\ \\ \\ \\ \end{bmatrix} \\ \\ \end{bmatrix} \\ \begin{bmatrix} 64, 512, 7, 7 \end{bmatrix} \\ \\ \\ \\ \end{bmatrix} \\ \begin{bmatrix} 64, 4096 \end{bmatrix} \\ \\ \\ \\ \end{bmatrix} \\ \begin{bmatrix} 64, 4096 \end{bmatrix} \\ \\ \\ \\ \end{bmatrix} \\ \begin{bmatrix} 64, 4096 \end{bmatrix} \\ \\ \\ \\ \end{bmatrix} \\ \end{bmatrix} $	
Total params: 138,357,544 Trainable params: 138,357,544 Non-trainable params: 0 Total mult-adds (G): 990.96 ====================================	), stride=(1, 1), padding= ing=0, dilation=1, ceil_mod put_sz, bias=True)	(1, 1)) de=False)

VGG-16 Architecture Summary

### A.1.2 ResNet-101

ResNet-101 is a deep convolutional neural network architecture that is part of the ResNet (Residual Network) family, which was introduced by Kaiming He and his team at Microsoft Research in 2015.<sup>118</sup> ResNet architectures were designed to address the degradation problem that occurs when training very deep networks. ResNet-101, with its 101 layers, is an extended version of the original ResNet architecture.

Key features:

- 101 layers that include convolutional layers, batch normalization layers, and ReLU activation functions.
- 2. Residual connections, also known as skip connections which allow the output of a layer to be added to the output of a later layer, enabling the network to learn residual functions.
- Bottleneck layers to reduce the number of parameters and computational complexity. Each bottleneck layer consists of three convolutional layers: a 1x1 layer for reducing dimensions, a 3x3 layer, and another 1x1 layer for restoring dimensions.



**Figure A.2:** Resnet architecture. Image adapted from He et al., "Deep residual learning for image recognition." *CVPR*, 2015.

Layer (type:depth-idx)	Output Shape	 Param #
PasNat		
$L_{Conv2d}$ , 1-1	[64 64 112 112]	9 108
$\square$	$\begin{bmatrix} 0 + i \\ 0 + i \\ 0 + i \\ 1 + 2 \\ 1 $	128
Dolli, 1-2	$\begin{bmatrix} 04, 04, 112, 112 \end{bmatrix}$	120
MenDeel2d. 1 (		
MaxPool20: 1-4	$\begin{bmatrix} 04, 04, 50, 50 \end{bmatrix}$	
- Sequencial: 1-5	$\begin{bmatrix} 04, & 250, & 50, & 50 \end{bmatrix}$	75 0.00
Bottleneck: 2-1	$\begin{bmatrix} 04, & 250, & 50, & 50 \end{bmatrix}$	75,008
Bottleneck: 2-2	[64, 256, 56, 56]	70,400
-Bottleneck: 2-5	[64, 236, 36, 36]	70,400
-Sequential: 1-6	[64, 512, 28, 28]	
Bottleneck: 2-4	[64, 512, 28, 28]	3/9,392
-Bottleneck: 2-5	[64, 512, 28, 28]	280,064
-Bottleneck: 2-6	[64, 512, 28, 28]	280,064
-Bottleneck: 2-7	[64, 512, 28, 28]	280,064
-Sequential: 1-7	[64, 1024, 14, 14]	
-Bottleneck: 2-8	[64, 1024, 14, 14]	1,512,448
-Bottleneck: 2-9	[64, 1024, 14, 14]	1,117,184
-Bottleneck: 2-10	[64, 1024, 14, 14]	1,117,184
-Bottleneck: 2-11	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-12	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-13	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-14	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-15	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-16	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-17	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-18	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-19	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-20	[64, 1024, 14, 14]	1,117,184
-Bottleneck: 2-21	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-22	[64, 1024, 14, 14]	1,117,184
-Bottleneck: 2-23	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-24	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-25	[64, 1024, 14, 14]	1,117,184
L-Bottleneck: 2-26	[64, 1024, 14, 14]	1,117,184
Bottleneck: 2-27	[64, 1024, 14, 14]	1,117,184
L_Bottleneck: 2-28	[64, 1024, 14, 14]	1,117,184
L-Bottleneck: 2-29	[64, 1024, 14, 14]	1,117,184
L-Bottleneck: 2-30	[64, 1024, 14, 14]	1,117,184
-Sequential: 1-8	[64, 2048, 7, 7]	
Bottleneck: 2-31	[64, 2048, 7, 7]	6,039,552
L-Bottleneck: 2-32	[64, 2048, 7, 7]	4,462,592
L-Bottleneck: 2-33	[64, 2048, 7, 7]	4,462,592
-AdaptiveAvgPool2d: 1-9	[64, 2048, 1, 1]	
Linear: 1-10	[64, 1000]	2,049,000
Total params: 44,549,160 Trainable params: 44,549,160 Non-trainable params: 0 Total mult-adds (G): 499.30		
Input size (MB): 38.54 Forward/backward pass size (MB): 16622.02 Params size (MB): 178.20 Estimated Total Size (MB): 16838.75	1	

## ResNet-101 Architecture Summary

```
*Bottleneck(
 (conv1): Conv2d(in_sz, out_sz, kernel_size=(1, 1), stride=(1, 1), bias=False)
 (bn1): BatchNorm2d(in_sz, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 (conv2): Conv2d(in_sz, out_sz, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(in_sz, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 (conv3): Conv2d(in_sz, out_sz, kernel_size=(1, 1), stride=(1, 1), bias=False)
 (bn3): BatchNorm2d(in sz, eps=1e-05, momentum=0.1, affine=True, track running stats=True)
 (relu): ReLU(inplace=True)
)
*Bottleneck(
 (conv1): Conv2d(in sz, out sz, kernel size=(1, 1), stride=(1, 1), bias=False)
 (bnl): BatchNorm2d(in sz, eps=1e-05, momentum=0.1, affine=True, track running stats=True)
 (conv2): Conv2d(in_sz, out_sz, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
 (bn2): BatchNorm2d(in_sz, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 (conv3): Conv2d(in_sz, out_sz, kernel_size=(1, 1), stride=(1, 1), bias=False)
 (bn3): BatchNorm2d(in_sz, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 (relu): ReLU(inplace=True)
 (downsample): Sequential(
   (0): Conv2d(in sz, out sz, kernel size=(1, 1), stride=(2, 2), bias=False)
   (1): BatchNorm2d(in sz, eps=1e-05, momentum=0.1, affine=True, track running stats=True)
)
*(avgpool): AdaptiveAvgPool2d(output size=(1, 1))
```

```
*(fc): Linear(in_features=2048, out_features=1000, bias=True)
```
#### A.1.3 DenseNet-121

DenseNet-121 is a deep convolutional neural network architecture that belongs to the DenseNet (Densely Connected Convolutional Networks) family, introduced by Gao Huang and his team in 2016.<sup>127</sup> DenseNet architectures were designed to improve information flow between layers and enhance feature reuse.

- 121 layers that include convolutional layers, batch normalization layers, and ReLU activation functions.
- 2. Dense connectivity between layers where each layer receives input from all preceding layers, allowing the network to reuse features and reduce the number of parameters.
- Transition layers between dense blocks to reduce spatial dimensions and the number of feature maps. These layers typically consist of a 1x1 convolutional layer followed by a 2x2 average pooling layer.



**Figure A.3:** DenseNet architecture. Image adapted from Huang et al., "Densely connected convolutional networks." *CVPR*, 2016.

Layer (type:depth-idx)	Output Shape	Param #
DenseNet	[64, 1000]	
-Sequential: 1-1	[64, 1024, /, /]	
-CONVZQ: 2-1	[64, 64, 112, 112]	9,408
BatChNorm2d: 2-2	[64, 64, 112, 112]	128
-ReLU: 2-3	[04, 04, 112, 112]	
L DenseBlock: 2-5	[64 256 56 56]	
DenseLaver: 3-1	[64, 32, 56, 56]	45,440
DenseLaver: 3-2	[64, 32, 56, 56]	49,600
DenseLaver: 3-3	[64, 32, 56, 56]	53,760
DenseLayer: 3-4	[64, 32, 56, 56]	57,920
DenseLayer: 3-5	[64, 32, 56, 56]	62,080
DenseLayer: 3-6	[64, 32, 56, 56]	66,240
L Transition: 2-6	[64, 128, 28, 28]	
BatchNorm2d: 3-7	[64, 256, 56, 56]	512
L_ReLU: 3-8	[64, 256, 56, 56]	
Conv2d: 3-9	[64, 128, 56, 56]	32,768
AvgPool2d: 3-10	[64, 128, 28, 28]	
LDenseBlock: 2-7	[64, 512, 28, 28]	
DenseLayer: 3-11	[64, 32, 28, 28]	53,760
DenseLayer: 3-12	[64, 32, 28, 28]	57,920
DenseLayer: 3-13	[64, 32, 28, 28]	62,080
DenseLayer: 3-14	[64, 32, 28, 28]	66,240
DenseLayer: 3-15	[64, 32, 28, 28]	70,400
DenseLayer: 3-16	[64, 32, 28, 28]	/4,560
DenseLayer: 3-17	[64, 32, 28, 28]	/8,/20
DenseLayer: 3-18	[64, 32, 28, 28]	82,880
DenseLayer: 3-19	[04, 32, 20, 20] [64, 32, 20, 20]	87,040 01 200
Densel aver: 3-21	[04, 32, 20, 20]	91,200
DenseLayer: 3-22	$\begin{bmatrix} 04, 32, 20, 20 \end{bmatrix}$	99,520
Transition: 2-8	[64, 256, 14, 14]	
$\square$	[64, 512, 28, 28]	1,024
└_ReLU: 3-24	[64, 512, 28, 28]	
└─Conv2d: 3-25	[64, 256, 28, 28]	131,072
└─AvgPool2d: 3-26	[64, 256, 14, 14]	
L_ DenseBlock: 2-9	[64, 1024, 14, 14]	
DenseLayer: 3-27	[64, 32, 14, 14]	70,400
DenseLayer: 3-28	[64, 32, 14, 14]	74,560
DenseLayer: 3-29	[64, 32, 14, 14]	78,720
DenseLayer: 3-30	[64, 32, 14, 14]	82,880
DenseLayer: 3-31	[64, 32, 14, 14]	87,040
DenseLayer: 3-32	[64, 32, 14, 14]	91,200
DenseLayer: 3-33	[64, 32, 14, 14]	95,360
DenseLayer: 3-34	[64, 32, 14, 14]	99,520
DenseLayer: 3-35	[64, 32, 14, 14]	103,680
DenseLayer: 3-36	$\begin{bmatrix} 64, 32, 14, 14 \end{bmatrix}$	112 000
DenseLayer: 3-37	$\begin{bmatrix} 04, & 32, & 14, & 14 \end{bmatrix}$	116,160
Densel aver: 3-39	[04, 32, 14, 14] [64, 32, 14, 14]	120 320
DenseLayer: 3-40	$\begin{bmatrix} 04, 52, 14, 14 \end{bmatrix}$	124 480
DenseLayer: 3-41	[64, 32, 14, 14]	128-640
DenseLaver: 3-42	[64, 32, 14, 14]	132,800
DenseLaver: 3-43	[64, 32, 14, 14]	136,960
DenseLaver: 3-44	[64, 32, 14, 14]	141,120
DenseLayer: 3-45	[64, 32, 14, 14]	145,280
DenseLayer: 3-46	[64, 32, 14, 14]	149,440
DenseLayer: 3-47	[64, 32, 14, 14]	153,600
DenseLayer: 3-48	[64, 32, 14, 14]	157,760
DenseLayer: 3-49	[64, 32, 14, 14]	161,920
DenseLayer: 3-50	[64, 32, 14, 14]	166,080
└─_Transition: 2-10	[64, 512, 7, 7]	

```
[64, 1024, 14, 14]
         └─BatchNorm2d: 3-51
                                                           2,048
         └_ReLU: 3-52
                                   [64, 1024, 14, 14]
                                                            ___
         └─Conv2d: 3-53
                                   [64, 512, 14, 14]
                                                            524,288
                                   [64, 512, 7, 7]
[64, 1024, 7, 7]
         └─AvgPool2d: 3-54
     - DenseBlock: 2-11
                                                            ___
         L DenseLayer: 3-55
                                   [64, 32, 7, 7]
                                                           103,680
         └─_DenseLayer: 3-56
                                   [64, 32, 7, 7]
                                                           107,840
         └─_DenseLayer: 3-57
                                   [64, 32, 7, 7]
                                                           112,000
         └─_DenseLayer: 3-58
└─_DenseLayer: 3-59
                                   [64, 32, 7, 7]
[64, 32, 7, 7]
[64, 32, 7, 7]
                                                           116,160
                                                            120,320
         └─_DenseLayer: 3-60
                                                           124,480
         └──DenseLayer: 3-61
                                   [64, 32, 7, 7]
                                                           128,640
         └──DenseLayer: 3-62
                                   [64, 32, 7, 7]
                                                           132,800
         └─_DenseLayer: 3-63
                                                           136,960
                                   [64, 32, 7, 7]
                                   [64, 32, 7, 7]
[64, 32, 7, 7]
[64, 32, 7, 7]
         DenseLayer: 3-64
                                                           141,120
         DenseLayer: 3-65
                                                            145,280
         └─ DenseLayer: 3-66
                                                           149,440
         └─_DenseLayer: 3-67
                                   [64, 32, 7, 7]
                                                           153,600
         └──DenseLayer: 3-68
                                   [64, 32, 7, 7]
                                                           157,760
         └─_DenseLayer: 3-69
                                   [64, 32, 7, 7]
[64, 32, 7, 7]
                                                           161,920
         DenseLayer: 3-70
                                                           166,080
    L-BatchNorm2d: 2-12
                                    [64, 1024, 7, 7]
                                                            2,048
Linear: 1-2
                                    [64, 1000]
                                                            1,025,000
Total params: 7,978,856
Trainable params: 7,978,856
Non-trainable params: 0
Total mult-adds (G): 181.39
_____
Input size (MB): 38.54
Forward/backward pass size (MB): 11554.64
Params size (MB): 31.92
Estimated Total Size (MB): 11625.09
_____
*DenseLayer(
 (norm1): BatchNorm2d(in sz, eps=1e-05, momentum=0.1, affine=True, track running stats=True)
 (relu1): ReLU(inplace=True)
 (conv1): Conv2d(in sz, out sz, kernel size=(1, 1), stride=(1, 1), bias=False)
 (norm2): BatchNorm2d(in sz, eps=1e-05, momentum=0.1, affine=True, track running stats=True)
 (relu2): ReLU(inplace=True)
 (conv2): Conv2d(in sz, out sz, kernel size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
*Transition(
   (norm): BatchNorm2d(in sz, eps=1e-05, momentum=0.1, affine=True, track running stats=True)
   (relu): ReLU(inplace=True)
   (conv): Conv2d(in sz, out sz, kernel size=(1, 1), stride=(1, 1), bias=False)
   (pool): AvgPool2d(kernel size=2, stride=2, padding=0)
)
*BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track running stats=True)
```

```
*Linear(in features=1024, out features=1000, bias=True)
```

#### A.1.4 EfficientNet-V2 Medium

EfficientNet-V2 Medium is a deep convolutional neural network architecture that belongs to the EfficientNet family, which was introduced by Mingxing Tan and his team at Google Research in 2019.<sup>119</sup> The EfficientNet-V2 series,<sup>190</sup> launched in 2021, builds on the original EfficientNet architecture and features improved training speed and efficiency.

- Fused-MBConv layers, which combine depthwise and pointwise convolutions into a single fused convolution operation. This reduces the number of layers and improves training speed while maintaining model accuracy.
- 2. MBConv blocks, which are building blocks that consist of a series of depthwise separable convolutions, batch normalization, and swish (SiLU) activation functions.
- 3. EfficientNet-V2 architectures are discovered through a Neural Architecture Search (NAS) approach, which involves searching for efficient network structures that achieve high performance with minimal resource consumption.
- 4. Progressive learning strategy that involves training smaller models first, then scaling up and fine-tuning the larger models.



**Figure A.4:** Structure of MBConv and Fused-MBConv. Image from Tan et al., "EfficientNetV2: smaller models and faster training." *ArXiv*, 2021.

EfficientNet-V2	Architecture	Summary
EIIICIENCNEL-V2	Archittecture	Summary

Layer (type:depth-idx)	Output Shape	Param #
EfficientNet	[64, 1000]	
-Sequential: 1-1	[64, 1280, 7, 7]	
Conv2dNormActivation: 2-1	[64, 24, 112, 112]	
Conv2d: 3-1	[64, 24, 112, 112]	648
BatchNorm2d: 3-2	[64, 24, 112, 112]	48
└─SiLU: 3-3	[64, 24, 112, 112]	
-Sequential: 2-2	[64, 24, 112, 112]	
-FusedMBConv: 3-4	[64, 24, 112, 112]	5,232
-FusedMBConv: 3-5	[64, 24, 112, 112]	5,232
FusedMBConv: 3-6	[64, 24, 112, 112]	5,232
Sequential: 2-3	[64, 48, 56, 56]	
FusedMBConv: 3-7	[64, 48, 56, 56]	23,632
FusedMBConv: 3-8	[64, 48, 56, 56]	92,640
FusedMBConv: 3-10		92,640
FusedMBConv: 3-11		92,640
-rusediaconv. 5-11		92,040
L EusedMBConv. 3-12	[64, 80, 28, 28]	98 8/8
LFusedMBConv: 3-13	[64, 80, 28, 28]	256 800
LFusedMBConv: 3-14	[64, 80, 28, 28]	256,800
FusedMBConv: 3-15	[64 80 28 28]	256,800
FusedMBConv: 3-16	[64 80 28 28]	256,800
-Sequential: 2-5	$\begin{bmatrix} 04, 00, 20, 20 \end{bmatrix}$	230,000
$ = \frac{1}{1} = \frac{1}{1} = \frac{1}{1}$	[64, 160, 14, 14]	94.420
$\square$	[64, 160, 14, 14]	265.320
$\square$	[64, 160, 14, 14]	265,320
$\square$	[64, 160, 14, 14]	265,320
└_MBConv: 3-21	[64, 160, 14, 14]	265,320
$\square$	[64, 160, 14, 14]	265,320
└─MBConv: 3-23	[64, 160, 14, 14]	265,320
-Sequential: 2-6	[64, 176, 14, 14]	
MBConv: 3-24	[64, 176, 14, 14]	413,192
MBConv: 3-25	[64, 176, 14, 14]	479,820
└─MBConv: 3-26	[64, 176, 14, 14]	479,820
MBConv: 3-27	[64, 176, 14, 14]	479,820
MBConv: 3-28	[64, 176, 14, 14]	479,820
MBConv: 3-29	[64, 176, 14, 14]	479,820
MBConv: 3-30	[64, 176, 14, 14]	479,820
MBConv: 3-31	[64, 176, 14, 14]	479,820
MBConv: 3-32	[64, 176, 14, 14]	479,820
MBConv: 3-33	[64, 176, 14, 14]	479,820
MBConv: 3-34	[64, 176, 14, 14]	479,820
MBConv: 3-35	[64, 176, 14, 14]	479,820
MBConv: 3-36	[64, 176, 14, 14]	479,820
MBConv: 3-37	[64, 176, 14, 14]	479,820
L-Sequential: 2-7	[64, 304, 7, 7]	
MBConv: 3-38	[64, 304, 7, 7]	615,244
MBConv: 3-39	[64, 304, 7, 7]	1,412,460
-MBCONV: 3-40	[64, 304, /, /]	1,412,460
MBConv: 3-41	[64, 304, 7, 7]	1,412,460
MBConv: 3-42	[64, 304, 7, 7]	1,412,460
MBCONV: 3-43	[64, 304, 7, 7]	1,412,460
-MPConv: 3-44	[04, 304, /, /] [6/ 30/ 7 7]	1,412,40U
$-MBCONV \cdot 3-45$	[04, 304, 7, 7] [67 307 7 7]	1,412,400 1 /12 /60
$-MBConv \cdot 3-47$	[04, 304, 7, 7] [64 307 7 7]	1 412 400
	[64 304 7 7]	1 412 /60
	[64, 304, 7, 7]	1,412 /60
	[64, 304, 7, 7]	1,412,460
	[64, 304, 7, 7]	1,412,460
$\square \square BConv: 3-52$	[64, 304, 7, 7]	1,412,460
$\square \square $	[64, 304, 7, 7]	1,412,460
1 1		_, 112, 100

```
└─MBConv: 3-54
                                                     [64, 304, 7, 7]
                                                                             1,412,460
         └─MBConv: 3-55
                                                     [64, 304, 7, 7]
                                                                             1,412,460
                                                     [64, 512, 7, 7]
[64, 512, 7, 7]
      -Sequential: 2-8
                                                                              ___
         └─MBConv: 3-56
                                                                             1,792,268
                                                     [64, 512, 7, 7]
         └─MBConv: 3-57
                                                                             3,976,320
         └─MBConv: 3-58
                                                     [64, 512, 7, 7]
                                                                             3,976,320
         └─MBConv: 3-59
                                                     [64, 512, 7, 7]
                                                                             3,976,320
         └─MBConv: 3-60
                                                     [64, 512, 7, 7]
                                                                             3,976,320
                                                     [64, 1280, 7, 7]
[64, 1280, 7, 7]
[64, 1280, 7, 7]
      -Conv2dNormActivation: 2-9
                                                                              _ _
         └─Conv2d: 3-61
                                                                             655,360
         └─BatchNorm2d: 3-62
                                                                             2,560
                                                     [64, 1280, 7, 7]
         └─SiLU: 3-63
                                                                             ___
 -AdaptiveAvgPool2d: 1-2
                                                     [64, 1280, 1, 1]
                                                                             --
 -Sequential: 1-3
                                                     [64, 1000]
                                                                              ___
     L-Dropout: 2-10
                                                     [64, 1280]
                                                                              ___
    L-Linear: 2-11
                                                     [64, 1000]
                                                                             1,281,000
Total params: 54,139,356
Trainable params: 54,139,356
Non-trainable params: 0
Total mult-adds (G): 343.20
_____
Input size (MB): 38.54
Forward/backward pass size (MB): 20054.26
Params size (MB): 216.56
Estimated Total Size (MB): 20309.35
_____
*FusedMBConv(
 (block): Sequential(
   (0): Conv2dNormActivation(
    (0): Conv2d(in sz, out sz, kernel size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
     (1): BatchNorm2d(in sz, eps=0.001, momentum=0.1, affine=True, track running stats=True)
     (2): SiLU(inplace=True))
   (1): Conv2dNormActivation(
     (0): Conv2d(in sz, out sz, kernel size=(1, 1), stride=(1, 1), bias=False)
     (1): BatchNorm2d(in sz, eps=0.001, momentum=0.1, affine=True, track running stats=True)))
)
*MBConv(
  (block): Sequential(
   (0): Conv2dNormActivation(
      (0): Conv2d(in_sz, out_sz, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(in sz, eps=0.001, momentum=0.1, affine=True, track running stats=True)
      (2): SiLU(inplace=True))
    (1): Conv2dNormActivation(
      (0): Conv2d(in sz, out sz, kernel size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=640, bias=False)
      (1): BatchNorm2d(in sz, eps=0.001, momentum=0.1, affine=True, track running stats=True)
      (2): SiLU(inplace=True))
    (2): SqueezeExcitation(
      (avgpool): AdaptiveAvgPool2d(output_size=1)
      (fc1): Conv2d(in_sz, out_sz, kernel_size=(1, 1), stride=(1, 1))
(fc2): Conv2d(in_sz, out_sz, kernel_size=(1, 1), stride=(1, 1))
      (activation): SiLU(inplace=True)
      (scale activation): Sigmoid())
    (3): Conv2dNormActivation(
      (0): Conv2d(in_sz, out_sz, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(in_sz, eps=0.001, momentum=0.1, affine=True, track_running_stats=True))
)
*(avgpool): AdaptiveAvgPool2d(output size=1)
*(classifier): Sequential(
   (0): Dropout(p=0.3, inplace=True)
    (1): Linear(in features=1280, out features=1000, bias=True)
)
```

#### A.1.5 Swin Transformer

Swin Transformer B is a deep convolutional neural network architecture that belongs to the Swin Transformer family, introduced by Ze Liu and his team at Microsoft Research Asia in 2021.<sup>133</sup> The Swin Transformer is designed to bring the power of the Transformer architecture, originally developed for natural language processing tasks, to computer vision.

- 1. Hierarchical structure that consists of multiple stages, each with a different spatial resolution.
- 2. Window-based self-attention, which divides the input into non-overlapping local windows and applies self-attention independently within each window.
- 3. Shifted windows which involve shifting the windows across stages to capture long-range dependencies.
- 4. Patch merging operation to reduce spatial dimensions and increase the number of channels between stages.
- 5. Linear layer token fusion which combines the outputs of multiple self-attention heads to form a new token.
- 6. Pre-normalization design where layer normalization is applied before the self-attention and feed-forward layers.



**Figure A.5:** Swin Transformer architecture. Image from Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows." *ICCV*, 2021.

Swin	Transformer	в	Architecture	Summary
		_		<u> </u>

Layer (type:depth-idx)	Output Shape	Param #								
SwinTransformer	[64, 1000]									
-Sequential: 1-1	[64, 7, 7, 1024]									
L_Sequential: 2-1	[64, 56, 56, 128]									
Conv2d: 3-1	[64, 128, 56, 56]	6,272								
LPermute: 3-2	[64, 56, 56, 128]									
LaverNorm: 3-3	[64, 56, 56, 128]	256								
-Sequential: 2-2	[64 56 56 128]									
SwinTransformerBlock: 3-4	$\begin{bmatrix} 64 & 56 & 56 & 128 \end{bmatrix}$	198 9/8								
SwinTransformerBlock: 3-5	$\begin{bmatrix} 0 \\ -7 \\ -7 \\ -7 \\ -7 \\ -7 \\ -7 \\ -7 \\ $	198 948								
DetabMorging: 2-2	$\begin{bmatrix} 04, 50, 50, 120 \end{bmatrix}$	190,940								
PatchMerging: 2-5	[64, 20, 20, 206]	1 004								
LayerNorm: 3-6	[64, 28, 28, 512]	1,024								
Linear: 3-7	[64, 28, 28, 256]	131,072								
Sequential: 2-4	[64, 28, 28, 256]									
SwinTransformerBlock: 3-8	[64, 28, 28, 256]	791,112								
SwinTransformerBlock: 3-9	[64, 28, 28, 256]	791,112								
L-PatchMerging: 2-5	[64, 14, 14, 512]									
LayerNorm: 3-10	[64, 14, 14, 1024]	2,048								
Linear: 3-11	[64, 14, 14, 512]	524,288								
L-Sequential: 2-6	[64, 14, 14, 512]									
SwinTransformerBlock: 3-12	[64, 14, 14, 512]	3,155,088								
└─SwinTransformerBlock: 3-13	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-14	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-15	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-16	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-17	$\begin{bmatrix} 64 & 14 & 14 & 512 \end{bmatrix}$	3 155 088								
SwinTransformerBlock: 3-18	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 6 & 1 & 1 & 1 & 512 \end{bmatrix}$	3 155 088								
SwinTransformerBlock: 3-10	$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 5 & 1 \\ \end{bmatrix}$	2 155 000								
	$\begin{bmatrix} 04, 14, 14, 512 \end{bmatrix}$	3,155,000 3 1EE 000								
	$\begin{bmatrix} 04, 14, 14, 512 \end{bmatrix}$	3,155,000								
SwintransformerBlock: 3-21	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-22	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-23	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-24	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-25	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-26	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-27	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-28	[64, 14, 14, 512]	3,155,088								
SwinTransformerBlock: 3-29	[64, 14, 14, 512]	3,155,088								
L-PatchMerging: 2-7	[64, 7, 7, 1024]									
LayerNorm: 3-30	[64, 7, 7, 2048]	4,096								
Linear: 3-31	[64, 7, 7, 1024]	2,097,152								
-Sequential: 2-8	[64, 7, 7, 1024]									
SwinTransformerBlock: 3-32	[64, 7, 7, 1024]	12,601,632								
└─SwinTransformerBlock: 3-33	[64, 7, 7, 1024]	12,601,632								
LaverNorm: 1-2	[64, 7, 7, 1024]	2.048								
$-Permute \cdot 1-3$	[64, 1024, 7, 7]									
AdaptiveAvgPool2d: 1-4	$[64 \ 1024 \ 1 \ 1]$									
Flatten: 1-5	[64 1024]									
Lincar: 1-6	[64 1000]	1 025 000								
	[04, 1000]	1,023,000								
Total params: 87,768,224 Trainable params: 87,768,224 Non-trainable params: 0 Total mult-adds (G): 5.08										
Input size (MB): 38.54 Forward/backward pass size (MB): 12126.24										
Estimated Total Size (MB): 12403.73										

```
*SwinTransformerBlock(
        (norm1): LayerNorm((in_sz,), eps=1e-05, elementwise_affine=True)
        (attn): ShiftedWindowAttention(
          (qkv): Linear(in features=in sz, out features=out sz, bias=True)
          (proj): Linear(in_features=in_sz, out_features=out_sz, bias=True))
        (stochastic_depth): StochasticDepth(p=0.043478260869565216, mode=row)
        (norm2): LayerNorm((in sz,), eps=le-05, elementwise affine=True)
        (mlp): MLP(
          (0): Linear(in_features=in_sz, out_features=out_sz, bias=True)
          (1): GELU(approximate='none')
          (2): Dropout (p=0.0, inplace=False)
          (3): Linear(in features=in sz, out features=out sz, bias=True)
          (4): Dropout(p=0.0, inplace=False))
)
*PatchMerging(
      (reduction): Linear(in_features=in_sz, out_features=out_sz, bias=False)
      (norm): LayerNorm((in_sz,), eps=1e-05, elementwise_affine=True)
)
*(permute): Permute()
*(avgpool): AdaptiveAvgPool2d(output size=1)
```

\*(head): Linear(in\_features=1024, out\_features=1000, bias=True)

#### A.1.6 ConvNeXt

The ConvNeXT model was proposed in A ConvNet for the 2020s by Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie. ConvNeXT is a pure convolutional model (ConvNet), inspired by the design of Vision Transformers.<sup>134</sup> They propose that by borrowing ideas from the successes of the Vision transformer and CNNs, one can build a pure ConvNet whose performance match state-of-the-art-models like the Vision transformer. ConvNeXt takes a standard neural network, a ResNet-50, and morphs it such that the design approaches a Vision Transformer.

- 1. Sliding windows in ResNet behave more similarly to the patches of the vision transformer.
- 2. Large kernel size and a stride such that the sliding window does not overlap, like the non-overlapping patches in a transformer.
- 3. Depthwise convolution where the number of groups equals the number of channels.
- 4. Replaces the ReLU activation function with the Gaussian Error Linear Unit (GELU).
- 5. Replaces batch normalization with layer normalization.



**Figure A.6:** ConvNeXt architecture. Image from Chen et al., "Large-scale individual building extraction from open-source satellite imagery via super-resolution-based instance segmentation approach." *ISPRS*, 2023.

ConvNeXt	Architecture	Summary

Layer (type:depth-idx)	Output Shape	Param #
ConvNeXt	[64, 1000]	
-Seguential: 1-1	[64, 1024, 7, 7]	
L-Conv2dNormActivation: 2-1	[64, 128, 56, 56]	
$\Box$	[64, 128, 56, 56]	6.272
$\Box$	[64, 128, 56, 56]	256
Sequential: 2-2	[64, 128, 56, 56]	
$\square$	[64, 128, 56, 56]	138.496
$\square$ CNBlock: 3-4	[64, 128, 56, 56]	138,496
$\square$ CNBlock: 3-5	[64, 128, 56, 56]	138.496
Sequential: 2-3	[64, 256, 28, 28]	
$\Box$	[64, 128, 56, 56]	256
$\square$	[64, 256, 28, 28]	131.328
Sequential: 2-4	[64, 256, 28, 28]	
$\square$	[64, 256, 28, 28]	539.136
$\square$	[64, 256, 28, 28]	539.136
$\Box$ CNBlock: 3-10	[64, 256, 28, 28]	539,136
-Sequential: 2-5	$\begin{bmatrix} 61 \\ 200$	
$\Box$	[64 256 28 28]	512
$\square$	$\begin{bmatrix} 0 & 1 \\ 2 & 0 \\ 6 & 5 & 12 \end{bmatrix}$	524 800
Sequential: 2-6	$\begin{bmatrix} 0 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ $	524,000
LCNBlock: 3-13	$\begin{bmatrix} 0 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ $	2 126 848
$\square$	$\begin{bmatrix} 0 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ $	2 126 848
$\square$	$\begin{bmatrix} 04, & 512, & 14, & 14 \end{bmatrix}$	2 126 848
$\square$ CNBlock: $3-16$	$\begin{bmatrix} 04, & 512, & 14, & 14 \end{bmatrix}$	2,120,040
$ = \frac{-\text{CNBLOCK}}{-\text{CNBLOCK}} = \frac{3-17}{-17} $	[04, J12, 14, 14] [64 512 14 14]	2,120,040
$\square$	$\begin{bmatrix} 04, & 512, & 14, & 14 \end{bmatrix}$	2 126 848
-CNBlock: 3-10	$\begin{bmatrix} 04, & 512, & 14, & 14 \end{bmatrix}$	2,120,040
-CNBlock: 3-19	$\begin{bmatrix} 04, & 512, & 14, & 14 \end{bmatrix}$	2,120,040
$\square$ CNBlock: 3-20	$\begin{bmatrix} 64, 512, 14, 14 \end{bmatrix}$	2,120,040
$\square \square $	[64, 512, 14, 14]	2,126,848
$\square$ CNBlock: 3-22	[64, 512, 14, 14]	2,126,848
$\square$ CNBLOCK: 3-23	[64, 512, 14, 14]	2,126,848
$\square$ CNBLOCK: 3-24	[64, 512, 14, 14]	2,126,848
$\square$ CNBLOCK: 3-25	[64, 512, 14, 14]	2,126,848
$\square$ CNBLOCK: 3-26	[64, 512, 14, 14]	2,126,848
$\square \square $	[64, 512, 14, 14]	2,126,848
$\square$ CNBlock: 3-28	[64, 512, 14, 14]	2,126,848
$\square$ CNBLOCK: 3-29	[64, 512, 14, 14]	2,126,848
$\square$ CNBLOCK: 3-30	[64, 512, 14, 14]	2,126,848
$\square$ CNBlock: 3-31	[64, 512, 14, 14]	2,126,848
CNBlock: 3-32	[64, 512, 14, 14]	2,126,848
CNBlock: 3-33	[64, 512, 14, 14]	2,126,848
$\square$ CNBlock: 3-34	[64, 512, 14, 14]	2,126,848
$\square$ CNBlock: 3-35	[64, 512, 14, 14]	2,126,848
CNBlock: 3-36	[64, 512, 14, 14]	2,126,848
CNBlock: 3-37	[64, 512, 14, 14]	2,126,848
CNBlock: 3-38	[64, 512, 14, 14]	2,126,848
CNBlock: 3-39	[64, 512, 14, 14]	2,126,848
Sequential: 2-7	[64, 1024, 7, 7]	
LayerNorm2d: 3-40	[64, 512, 14, 14]	1,024
Conv2d: 3-41	[64, 1024, /, /]	2,098,176
Sequential: 2-8	[64, 1024, 7, 7]	
CNBLock: 3-42	[64, 1024, 7, 7]	8,448,000
CNBLock: 3-43	[64, 1024, 7, 7]	8,448,000
CNBlock: 3-44	[64, 1024, 7, 7]	8,448,000
-AdaptiveAvgPool2d: 1-2	[64, 1024, 1, 1]	
-Sequential: 1-3	[64, 1000]	
LayerNorm2d: 2-9	[64, 1024, 1, 1]	2,048
-Flatten: 2-10	[64, 1024]	
Linear: 2-11	[64, 1000]	1,025,000

```
Total params: 88,591,464
Trainable params: 88,591,464
Non-trainable params: 0
Total mult-adds (G): 41.38
_____
Input size (MB): 38.54
Forward/backward pass size (MB): 17675.83
Params size (MB): 354.29
Estimated Total Size (MB): 18068.66
_____
*CNBlock(
  (block): Sequential(
    (0): Conv2d(in sz, out sz, kernel size=(7, 7), stride=(1, 1), padding=(3, 3), groups=256)
    (1): Permute()
    (2): LayerNorm((in_sz,), eps=1e-06, elementwise_affine=True)
    (3): Linear(in features=in sz, out features=out sz, bias=True)
    (4): GELU(approximate='none')
    (5): Linear(in_features=in_sz, out_features=out_sz, bias=True)
    (6): Permute()
)
*PatchMerging(
     (reduction): Linear(in features=in sz, out features=out sz, bias=False)
     (norm): LayerNorm((in sz,), eps=1e-05, elementwise affine=True)
)
*(avgpool): AdaptiveAvgPool2d(output size=1)
*(classifier): Sequential(
   (0): LayerNorm2d((in_sz,), eps=1e-06, elementwise_affine=True)
   (1): Flatten(start dim=1, end dim=-1)
   (2): Linear(in features=1024, out features=1000, bias=True)
)
```

### A.2 Training Strategies

In this section a brief overview of the deep network training strategies are outlined. Independent of network architecture, these strategies were employed for most results presented in this thesis and constitute a general workflow of deep learning model training.

#### A.2.1 Training/Validation Split and Class Imbalance

In deep learning, the traditional training-validation split is a fundamental technique for dividing a dataset into separate subsets to effectively train and evaluate a model. This approach ensures that the model generalizes well to unseen data and helps to prevent overfitting. Generally, the entire dataset is divided into two subsets - the training set and the validation set. The training set is used to train the model, while the validation set is used to evaluate the model's performance during training. The split ratio is usually around 70-80% of the data for the training set and 20-30% for the validation set, although this can vary depending on the specific problem and dataset size. At regular intervals during training, typically after each epoch, the model's performance is evaluated on the validation set. This helps to monitor the model's progress and determine if the model is overfitting or underfitting. Note the model's parameters are not updated when evaluating the validation dataset allows for hyperparameter tuning, optimizations, and comparisons such that the test set can act as a completely unbiased evaluation of the model performance.

It is not uncommon for CXR datasets to contain multiple images from a single patient. However, this can pose a problem when partitioning data into training, validation, and test sets due to data leakage. Data leakage occurs when information used during the model training process is not expected to be available at prediction time, causing predictive scores (metrics) to overestimate the model's utility in a production environment. For instance, consider a medical imaging dataset containing CXR images from multiple patients, with each patient having several images taken at different time points. If the data split is performed randomly without considering patient information, it is possible that images from the same patient end up in all three sets. This can lead to an overly optimistic estimation of the model's performance on unseen data and may result in poor generalization when applied to new patients.

In this work a 80%/20% training/validation ratio was used to train the models. To avoid data

leakage during model training, Algorithm 1 was employed to partition the training and validation

sets based on patient ID.

Algorithm 1 Patient-Wise Data Splitting for Training and Validation Sets

<b>Require:</b> Dataset <i>D</i> containing images and their labels (positive or negative), Patient ID information
for each image
<b>Ensure:</b> Training set <i>T</i> , Validation set <i>V</i>
1: Initialize empty lists for positive images $(P)$ and negative images $(N)$
2: for each image $i$ in dataset $D$ do
3: <b>if</b> <i>i</i> has a positive label <b>then</b>
4: Append $i$ to the list $P$
5: else
6: Append $i$ to the list $N$
7: end if
8: end for
9: Initialize empty sets for unique positive patient IDs ( $P_{unique}$ ) and unique negative patient IDs
$(N_{unique})$
10: <b>for</b> each image <i>i</i> in lists <i>P</i> and <i>N</i> <b>do</b>
11: <b>if</b> <i>i</i> belongs to list <i>P</i> <b>then</b>
12: <b>if</b> <i>i</i> 's patient ID is not in $P_{unique}$ <b>then</b>
13: Append its patient ID to the set $P_{unique}$
14: end if
15: <b>else</b>
16: <b>if</b> <i>i</i> 's patient ID is not in $N_{unique}$ <b>then</b>
17: Append its patient ID to the set $N_{unique}$
18: end if
19: end if
20: end for
21: Randomly shuffle the elements in sets $P_{unique}$ and $N_{unique}$
22: Compute the split indices for 80% and 20% in both $P_{unique}$ and $N_{unique}$
23: $idx_P \leftarrow \lfloor 0.8 * \operatorname{len}(P_{unique}) \rfloor$
24: $idx_N \leftarrow \lfloor 0.8 * \operatorname{len}(N_{unique}) \rfloor$
25: Split the patient IDs in $P_{unique}$ and $N_{unique}$ into training and validation subsets
26: $P_{train\_IDs} \leftarrow P_{unique}[:idx_P]$
27: $P_{val\_IDs} \leftarrow P_{unique}[idx_P:]$
28: $N_{train\_IDs} \leftarrow N_{unique}[:idx_N]$
29: $N_{val\_IDs} \leftarrow N_{unique}[idx_N:]$
30: Initialize empty lists for training set $T$ and validation set $V$
31: <b>for</b> each image <i>i</i> in lists <i>P</i> and <i>N</i> <b>do</b>
32: if <i>i</i> 's patient ID is in $P_{train\_IDs}$ or $N_{train\_IDs}$ then
33: Append <i>i</i> to the list $T$
34: else
35: Append <i>i</i> to the list $V$
36: end it
37: end for
38: return Training set T, Validation set V

Another common challenge in creating datasets is the presence of imbalanced class distributions. This issue is especially significant in medical imaging, such as in cases where a disease is comparatively rare. For instance, there is a substantial class imbalance in the COVID-19 datasets used in this work, with a considerably higher number of COVID-19 negative images compared to COVID-19 positive ones. In such situations, the model develops a bias towards the majority class, leading to subpar performance on the minority class. Numerous methods exist to address class imbalance,<sup>191</sup> and in this work, an oversampling approach was employed. The number of minority class samples was augmented by resampling from the minority class, adhering to Algorithm 2.

#### Algorithm 2 Balance Dataset

**Require:** Dataset D containing images and their labels (positive or negative) **Ensure:** Balanced dataset *D*<sub>balanced</sub> 1: Initialize empty lists for positive images (P) and negative images (N)2: for each image *i* in dataset *D* do if *i* has a positive label then 3: Append i to the list P4: 5: else Append i to the list N6: 7: end if 8: end for 9: Calculate the number of positive and negative images 10:  $n_{pos} \leftarrow \text{length}(P)$ 11:  $n_{neq} \leftarrow \text{length}(N)$ 12: Compute the ratio r of negative to positive images 13:  $r \leftarrow n_{pos}/n_{neq}$ 14: Upsample the minority class by duplicating according to ratio and then randomly sample for the remaining difference 15: if r > 2 then 16:  $f \leftarrow |r|$  $P_{duplicates} \leftarrow duplicate(P, f)$ 17:  $P \leftarrow \text{concatenate}(P, P_{duplicates})$ 18: 19:  $P_{remaining} \leftarrow \text{sample}(P, n_{neg} - n_{pos} * f)$  $D_{balanced} \leftarrow \text{concatenate}(P, P_{remaining}, N)$ 20: 21: else if r < 0.5 then  $f \leftarrow |1/r|$ 22: 23:  $N_{duplicates} \leftarrow duplicate(N, f)$  $N \leftarrow \text{concatenate}(N, N_{duplicates})$ 24: 25:  $N_{remaining} \leftarrow \text{sample}(N, n_{pos} - n_{neg} * 1/f)$ 26:  $D_{balanced} \leftarrow \text{concatenate}(P, N, N_{remaining})$ 27: end if 28: return  $D_{balanced}$ 

#### A.2.2 Ensemble Learning

Ensemble learning is a widely-used deep learning technique that merges predictions from multiple models to enhance overall performance and achieve more accurate results.<sup>185–187</sup> This approach is grounded in the idea that a diverse group of models can make better collective decisions than any single model. Combining the predictions from various models can decrease the chances of errors stemming from the limitations or biases of individual models, often leading to improved performance. Ensemble learning contributes to more robust and stable final predictions by averaging out the noise or errors from separate models. This can help mitigate overfitting since ensembles are less prone to rely on a single model's idiosyncrasies. Ensemble learning fosters diversity among models by integrating different architectures, training strategies, or subsets of the training data. This diversity enables the ensemble to capture a wider range of data patterns and facilitates a deeper understanding of the underlying problem.

In this work, for the majority of the presented results, an ensemble of N models (10 for dataset size study, 5 for other studies) was trained on the same dataset, but with different training/validation splits according to Algorithm 1. To ensure that each model in the ensemble had a distinct data split, unique random seeds were used for each model. To obtain the final prediction score, a total of N predictions (corresponding to the N models) were made for each image, and the final prediction  $\bar{p}$  was calculated using the quadratic mean:

$$\bar{p} = \sqrt{\frac{\sum_{i=1}^{N} p_i^2}{N}}.$$
(A.1)

#### A.2.3 Model Pretaining

Model pretraining is a widely adopted technique in deep learning that involves training a model on a large-scale dataset before fine-tuning it on a specific task or a smaller dataset. This approach has proven to be particularly effective in improving model performance, especially in scenarios with limited labeled data. In the supervised fine-tuning stage, the pretrained model is further trained on a smaller labeled dataset specific to the target task. During this stage, the model refines its learned representations and adapts them to the task at hand. Fine-tuning can be performed on the entire model or on a subset of layers, depending on the similarity between the pretraining and target tasks, as well as the size of the labeled dataset.

Pretraining offers several benefits, such as enhanced overall performance, accelerated convergence, and a reduced likelihood of overfitting on smaller datasets. These advantages are discussed in greater detail in Section 5.3.4 of Chapter 5. In this work, a two-stage pretraining scheme was employed for the majority of the models. The first stage involved pretraining on the ImageNet dataset. This training was not carried out explicitly; instead, the weights were obtained from the PyTorch torchvision library. In the second stage, the models were pretrained on the NIH CXR dataset, as described in Chapter 3. This pretraining was conducted by the author, Ran Zhang, and Xin Tie. These pretrained weights served as the initialization for the COVID-19 classification models used throughout this work.

#### A.2.4 Learning Rate Adjustment and Regularization

Learning rate is a crucial hyperparameter in deep learning, as it determines the magnitude of updates applied to the model's weights during the training process. The learning rate influences the speed and accuracy of convergence in the optimization process, playing a vital role in the overall performance of the model.

Adjusting the learning rate during training is a common practice in deep learning, as it enables the model to benefit from the advantages of both high and low learning rates. This approach is known as learning rate scheduling or adaptive learning rates.<sup>192</sup> By employing a higher learning rate at the beginning of training, the model can learn rapidly and escape suboptimal regions in the loss landscape. As the training progresses, reducing the learning rate allows the model to converge more accurately to the optimal solution.

Various learning rate scheduling techniques have been proposed, <sup>193</sup> such as step decay, exponential decay, cosine annealing, and cyclical learning rates. These techniques adjust the learning rate according to specific rules or heuristics, often based on the number of training epochs or iterations. Additionally, adaptive learning rate optimization algorithms, such as AdaGrad, RMSProp, and Adam, automatically adjust the learning rate for each weight based on the observed gradients during training, allowing the model to adapt more effectively to the problem at hand. In this work, learning rate  $\alpha = 5e - 5$  was used for the pretrained models as was adjusted according to Algorithm 3. Regularization is a critical technique used in machine learning and deep learning to prevent overfitting, enhance model generalization, and improve performance on unseen data. Overfitting occurs when a model learns the noise and idiosyncrasies of the training data, resulting in poor performance on new, unseen data. Regularization methods introduce constraints or penalties to the learning process, encouraging the model to learn simpler, more generalizable representations.

In this work three main forms of regularization were used including dropout, <sup>194</sup> weight decay, and early stopping. During training, dropout randomly deactivates a certain percentage of neurons in each layer at each iteration, effectively training a subnetwork of the original neural network. The default dropout parameters were used for each architecture described previously (generally p = [0.3, 0.5]). Weight decay, also known as L2 regularization, adds a penalty term to the loss function based on the squared magnitude of the model's weights. This penalty encourages the model to learn smaller, more constrained weights, leading to simpler and more interpretable models. In this work weight decay strength  $\lambda = 1e - 5$  was used. Last, early stopping is a regularization technique that monitors the model's performance on a validation set during training. If the model's performance on the validation set does not improve (or worsens) for a predefined number of consecutive epochs, the training process is halted, and the best-performing model is selected. In this work the learning rate was adjusted and early stopping accomplished according to Algorithm 3.

#### A.2.5 Optimizer

Optimizers govern the process of updating model parameters to minimize the loss function. The choice of an optimizer can significantly impact the speed and accuracy of convergence, as well as the overall performance of the model. Optimizers aim to find the optimal set of model parameters that yield the lowest loss on the training data while maintaining good generalization performance on unseen data.

There are several optimization algorithms available, ranging from simple methods like gradient descent to more sophisticated adaptive algorithms. Gradient descent and its variants, such as stochastic gradient descent and mini-batch gradient descent, are foundational optimization techniques that update the model parameters based on the gradient of the loss function. While these methods are easy to implement and computationally efficient, they may suffer from slow convergence or get stuck in suboptimal local minima.

Algorithm 3 Adaptive Training Loop with Early Stopping

**Require:** *E* – maximum number of epochs to train the model **Ensure:** The trained model

1: Initialize variables:

```
2:
       loss_{min} \leftarrow \infty
 3:
       N_{save} \leftarrow 0
 4:
       N_{stop} \leftarrow 0
       \lambda \leftarrow 5e-5
 5:
       \lambda_{min} \leftarrow 1e - 6
 6:
 7: Begin training loop:
 8: for epoch in range(1, E + 1) do
 9:
          Train the model for one epoch and compute current loss
10:
       if loss < loss_{min} then
             Save the model
11:
12:
              N_{save} \leftarrow epoch
13:
             loss_{min} \leftarrow loss
       else
14:
15:
              N_{stop} \leftarrow epoch
16:
       end if
17:
       if N_{stop} - N_{save} > 3 then
              \lambda \leftarrow \lambda * 0.5
18:
19:
           if \lambda < \lambda_{min} then
20:
                 \lambda \leftarrow \lambda_{min}
           end if
21:
       end if
22:
23:
       if N_{stop} - N_{save} > 5 then
              Break the training loop
24:
25:
        end if
26: end for
27: return The trained model
```

To address these issues, adaptive optimization algorithms have been developed, which automatically adjust the learning rate for each parameter during training. In this work, the Adaptive Moment Estimation (Adam)<sup>64</sup> optimizer method was used. Adam computes adaptive learning rates for each parameter by estimating the first and second moments of the gradients. Specifically, Adam maintains an exponentially decaying average of past gradients (similar to momentum) and an exponentially decaying average of past squared gradients (similar to RMSProp). These estimates are then used to compute the adaptive learning rates, allowing the optimizer to adapt more effectively to different parts of the loss landscape. A weight decay value  $\lambda = 1e - 5$  was empirically chosen in this work.

#### A.2.6 Image Augmentation

Image augmentation is a widely used technique in the training of CNNs to improve their performance and generalization capabilities. The process involves applying various transformations to the input images, such as rotation, scaling, flipping, or changing brightness and contrast. These transformations effectively increase the size and diversity of the training dataset, enabling the model to learn more robust and invariant features.

In this work, various augmentation techniques were investigated to enhance generalization capabilities and homogenize the data, thereby reducing the potential for shortcut learning. Generally, random image rotations (range =  $[0^\circ, 30^\circ]$ ) and random horizontal flipping (p=0.5) were applied to all trained models. Besides these standard augmentations, other techniques such as random adjustments to contrast, brightness, sharpness, and blurring were explored but found to have minimal impact on overall performance.

As demonstrated in Chapter 4, augmentations like histogram equalization and segmentation were also found to have little or even detrimental effects on the overall model performance. A comprehensive investigation of additional augmentations remains an area for future exploration and is one of the remaining tasks of this work.

## Appendix B

# Analysis of Deep Learning Extracted Features from CXR-Trained Models

In this appendix, a comparative analysis of deep learning extracted features from the COVID-19 datasets used in this work (excluding COVIDx, due to its substantial differences compared to other datasets) is presented. Pretrained DenseNet-121 models from the TorchXRayVision library<sup>167</sup> were employed to extract features from the combined COVID-19 datasets. These models were trained on the following datasets: ImageNet,<sup>46</sup> CheXpert,<sup>170</sup> MIMIC,<sup>101</sup> NIH ChestX-ray14,<sup>100</sup> PadChest,<sup>169</sup> RSNA Pneumonia Detection Challenge,<sup>168</sup> and a model trained on their combination. Adhering to the procedures outlined in Chapter 3, UMAP was used to reduce the dimensionality of the feature vectors for the trained models. K-means clustering (K=5) was performed, and the centroid labels were plotted on the UMAP graph. Randomly selected sample images from each cluster are displayed according to the centroid color on the corresponding plots. Finally, the isolated forest algorithm was employed to identify outlying images, and a random sample is presented.

This study reveals some intriguing observations. It is evident that the features extracted from ImageNet training differ from those obtained from CXR dataset training, as natural images vary considerably from radiographs of human anatomy. Chapter 5 showed that CXR pretraining (using the NIH dataset) led to better performance than ImageNet pretraining, which was anticipated since the NIH data and classification task are closely related to COVID-19 classification. However, the following results indicate that there are differences between the CXR datasets. Even though they all contain roughly the same type of image and classification task, the extracted features and outlying

images can be quite distinct.

This initial investigation implies that CXR pretraining can be further optimized by selecting different dataset(s) for pretraining. Since the extracted features and consequently the learned kernels are different, various CXR datasets might offer better pretraining for a model. This approach can be particularly beneficial when the target training dataset is very small. As illustrated in Figure 5.9 in Chapter 5, the performance in this region is significantly influenced by pretraining. A comprehensive study of the performance and generalizability of COVID-19 models based on CXR pretraining datasets remains a future research direction, with the preliminary data presented here providing support for the hypothesis.



**ImageNet Training Extracted Features** 





ImageNet Training Outliers



**CheXpert Training Extracted Features** 

15	and the second	A.	10			20	1	No.			12		No P	1	AN		inter-	and a second	A State
NE T	11	183	and a	- and	Contraction of the second		NA	AN		18	Anna I		The second	1		18	and a	11	Nuclear Section 1997
1	73	-	A.	Carl Carl	Non	Num.	and a	Summer of	Cooper L		ANNA ANNA		and .			- Ann		and a	
ZS	1		and I				-	N.X.	CORNEL CORNEL	10				A A					
KA	15	60	and a second	And A		Rul .			AN					ax.	28	A	and .		ZA
21	15	North			1	11		1						Thus.		A	11	76	71
10		-	78	11	A Real	123		11			-			25	15	AN	11	15	16
75	ZŊ	AND A	X	Kenn	18		41	11	11		19	78	1	19	11	30	Nucl.	1	
-	12	n	11	1	25			11		1		North Contraction	11	X		1	75		18
	(83)		11	X	1	21	1		-		11	1		Since of	AN	11	11	1	129
	Autor Contraction			h	1		1		1		1	M	-	25			ANNA ANA		11
1		1		11		10	1		11		11	SHORN -	78					dr	11
1	15	11	25	-	- And	IN		1.00	15	Antes -		T	Mini	NGIA	Sum.		TI	Sine of	
11								71		Anna -	1			15	1				1
1	and it			1		1		1	And A			I	anna .	and a second			1	1	11
1			25		19			21			1	-	25		11				
	1	1		15	- Same	10	25	MA		75	and the second		A Real		35	1		11	
18		TIN		-	Control of	100	35	-	-	11	RS		N'S	M				100	113
	1	AND A	11	11	1	78		11	18	1	25			11	1		N.S.	1	11
		-			and a	78	-	29	New York	75	20	100	21	1	18	The	No.K	The second	78
10		196	71	10 8	198	125		25	100		28	1	11	19.01	15	105	To T		1
78	TAN.	10		the state	79	1	20	75	Par al	11	24		X	Sec.	18	1	11	1	(FA)
The second	A A		and the second	and the second		MAN	197	-	a f	121	19	TAN	3.		Kar		1	11	
578	75	1	-	75	100		14	19 8			1		11	591	11		-	nn	
-		75	15	19	10	ZN	71		A STATE	1	1	St. D	-	N		20	The all	-	11



**CheXpert Training Outliers** 



**MIMIC Training Extracted Features** 





**MIMIC Training Outliers** 



**NIH Training Extracted Features** 





**NIH Training Outliers** 





**PadChest Training Extracted Features** 




**PadChest Training Outliers** 



**RSNA Training Extracted Features** 





**RSNA Training Outliers** 



**All Datasets Training Extracted Features** 





**All Datasets Training Outliers** 

## Appendix C

## **Additional Saliency Heatmap Examples**

In this appendix, supplementary saliency heatmaps generated from various models referenced in Chapter 6 are presented. For each model, 20 randomly chosen images per class are displayed in the subsequent pages. Unless otherwise noted, all heatmaps were produced using DenseNet-121.

As elaborated in Chapter 6 and further demonstrated in this appendix, the interpretation of saliency heatmaps can be challenging due to inconsistencies, with different techniques highlighting distinct features even within the same image. Frequently, the heatmaps exhibit poor localization, emphasizing broader regions, or appear noisy, which obscures the underlying features. Apart from the most evident models (e.g., marker and patient sex), no clear patterns emerge. The limitation of a small sample size of merely 20 per class should be acknowledged; however, as mentioned in various sources, saliency techniques generally lack the specificity and localization required for reliability. <sup>174,176–179,179,180,184</sup> While these methods might be helpful in certain cases, the overarching conclusion in this study is that they do not offer conclusive evidence for identifying shortcut features.



Marker & CV-19 Pos/CV-19 Neg 
Marker & CV-19 Pos Examples



Marker & CV-19 Pos/CV-19 Neg 
Marker & CV-19 Pos Examples



Male & CV-19 Pos/Female & CV-19 Neg ■ Male & CV-19 Pos Examples



Male & CV-19 Pos/Female & CV-19 Neg ■ Female & CV-19 Neg Examples



>70 & CV-19 Pos/<50 & CV-19 Neg = >70 & CV-19 Pos Examples



>70 & CV-19 Pos/<50 & CV-19 Neg = <50 & CV-19 Neg Examples



White & CV-19 Pos/Black & CV-19 Neg ■ White & CV-19 Pos Examples



White & CV-19 Pos/Black & CV-19 Neg 
Black & CV-19 Neg Examples



AP & CV-19 Pos/PA & CV-19 Neg ■ AP & CV-19 Pos Examples



AP & CV-19 Pos/PA & CV-19 Neg ■ PA & CV-19 Neg Examples



DX & CV-19 Pos/CR & CV-19 Neg DX & CV-19 Pos Examples



DX & CV-19 Pos/CR & CV-19 Neg ■ CR & CV-19 Neg Examples



Vendor 1 & CV-19 Pos/Vendor 2 & CV-19 Neg ■ Vendor 1 & CV-19 Pos Examples



Vendor 1 & CV-19 Pos/Vendor 2 & CV-19 Neg 
Vendor 2 & CV-19 Neg Examples



BIMCV+/HF- DenseNet-121 
COVID-19 Positive Examples



BIMCV+/HF- DenseNet-121 Model 
COVID-19 Negative Examples



BIMCV+/HF- Center Mask Model 
COVID-19 Positive Examples



BIMCV+/HF- Center Mask Model 
COVID-19 Negative Examples



BIMCV+/HF- Segmented Model 
COVID-19 Positive Examples



BIMCV+/HF- Segmented Model ■ COVID-19 Negative Examples



Sharpness Shortcut Model 
Added Sharpness Examples



Sharpness Shortcut Model 
No Added Sharpness Examples



Contrast Shortcut Model 
Added Contrast Examples



Contrast Shortcut Model 
No Added Contrast Examples



HF-Trained DenseNet-121 
COVID-19 Positive Examples



HF-Trained DenseNet-121 
COVID-19 Negative Examples



HF-Trained EfficientNet 
COVID-19 Positive Examples



HF-Trained EfficientNet 
COVID-19 Negative Examples


HF-Trained Swin Transformer 
COVID-19 Positive Examples



HF-Trained Swin Transformer 
COVID-19 Negative Examples

## References

- [1] World Health Organization, "WHO Coronavirus (COVID-19) Dashboard." https://covid19.who.int/, 2020.
- [2] J. R. Cleverley, J. Piper, and M. M. Jones, "The role of chest radiography in confirming COVID-19 pneumonia," *BMJ*, vol. 370, 2020.
- [3] A. Kohli, P. C. Hande, and S. Chugh, "Role of chest radiography in the management of COVID-19 pneumonia: An overview and correlation with pathophysiologic changes," *The Indian Journal of Radiology & Imaging*, vol. 31, pp. S70 – S79, 2021.
- [4] A. Cozzi, S. Schiaffino, F. Arpaia, G. D. Pepa, S. Tritella, P. Bertolotti, L. Menicagli, C. G. Monaco, L. A. Carbonaro, R. Spairani, B. B. Paskeh, and F. Sardanelli, "Chest X-ray in the COVID-19 pandemic: Radiologists' real-world reader performance," *European Journal of Radiology*, vol. 132, pp. 109272 109272, 2020.
- [5] D. Cozzi, M. Albanesi, E. Cavigli, C. Moroni, A. Bindi, S. Luvarà, S. Lucarini, S. Busoni, L. N. Mazzoni, and V. Miele, "Chest X-ray in new Coronavirus Disease 2019 (COVID-19) infection: Findings and correlation with clinical outcome," *La Radiologia Medica*, vol. 125, pp. 730 737, 2020.
- [6] S. H. Yoon, K. Y. Lee, J. Y. Kim, Y. K. Lee, H. Ko, K. H. Kim, C. M. Park, and Y. H. Kim, "Chest radiographic and CT findings of the 2019 Novel Coronavirus Disease (COVID-19): Analysis of nine patients treated in Korea," *Korean Journal of Radiology*, vol. 21, pp. 494 – 500, 2020.
- [7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.
- [8] J. Born, D. J. Beymer, D. Rajan, A. Coy, V. V. Mukherjee, M. Manica, P. Prasanna, D. Ballah, M. Guindy, D. Shaham, P. L. Shah, E. Karteris, J. L. Robertus, M. Gabrani, and M. Rosen-Zvi, "On the role of artificial intelligence in medical imaging of COVID-19," *Patterns*, vol. 2, 2020.
- [9] N. C. Achaiah, S. B. Subbarajasetty, and R. M. Shetty, "R0 and Re of COVID-19: Can we predict when the pandemic outbreak will be contained?," *Indian Journal of Critical Care Medicine : Peer-reviewed, Official Publication of Indian Society of Critical Care Medicine*, vol. 24, pp. 1125 – 1127, 2020.
- [10] G. Meyerowitz-Katz and L. Merone, "A systematic review and meta-analysis of published research data on COVID-19 infection fatality rates," *International Journal of Infectious Diseases*, vol. 101, pp. 138 – 148, 2020.

- [11] C. Cheng, D. Zhang, D. Dang, J. Geng, P. Zhu, M. Yuan, R. Liang, H. Yang, Y. Jin, J. Xie, S. Chen, and G. Duan, "The incubation period of COVID-19: A global meta-analysis of 53 studies and a Chinese observation study of 11,545 patients," *Infectious Diseases of Poverty*, vol. 10, 2021.
- [12] S. A. Lauer, K. H. Grantz, Q. Bi, F. K. Jones, Q. Zheng, H. R. Meredith, A. S. Azman, N. G. Reich, and J. Lessler, "The incubation period of Coronavirus Disease 2019 (COVID-19) from publicly reported confirmed cases: Estimation and application," *Annals of Internal Medicine*, 2020.
- [13] N. C. P. E. R. E. Team, "The epidemiological characteristics of an outbreak of 2019 Novel Coronavirus Diseases (COVID-19) in China," *Zhonghua liu xing bing xue za zhi = Zhonghua liuxingbingxue zazhi*, vol. 41 2, pp. 145–151, 2020.
- [14] Q. Li, X. hua Guan, P. Wu, X. Wang, L. Zhou, Y. Tong, R. Ren, K. S. M. Leung, E. H. Y. Lau, J. Y. T. Wong, X. sen Xing, N. juan Xiang, Y. Wu, C. Li, Q. Chen, D. Li, T. Liu, J. Zhao, M. Li, W. Tu, C.-T. Chen, L. Jin, R. Yang, Q. Wang, S. Zhou, R. Wang, H. X. Liu, Y. Luo, Y. Liu, G. J. Shao, H. Li, Z. Tao, Y. Yang, Z. Deng, B. Liu, Z. Ma, Y. Zhang, G. qing Shi, T. T. Y. Lam, J. T. S. Wu, G. F. Gao, B. J. Cowling, B. Yang, G. Leung, and Z. Feng, "Early transmission dynamics in Wuhan, China, of Novel Coronavirus–infected pneumonia," *The New England Journal of Medicine*, vol. 382, pp. 1199 1207, 2020.
- [15] W.-J. Guan, Z. yi Ni, Y. Hu, W. hua Liang, C. Ou, J. He, L. Liu, H. Shan, C. liang Lei, D. S. C. Hui, B. Du, L. Li, G. qiao Zeng, K. Y. Yuen, R. chong Chen, C. li Tang, T. Wang, P. yan Chen, J. Xiang, S. yue Li, J. lin Wang, Z. Liang, Y. xiang Peng, L. Wei, Y. Liu, Y. hua Hu, P. Peng, J. ming Wang, J. yang Liu, Z. Chen, G. Li, Z. jian Zheng, S. qin Qiu, J. Luo, C. jiang Ye, S. yong Zhu, and N. Zhong, "Clinical characteristics of Coronavirus Disease 2019 in China," *The New England Journal of Medicine*, 2020.
- [16] C. Robba, D. Battaglini, P. Pelosi, and P. R. M. Rocco, "Multiple organ dysfunction in SARS-CoV-2: MODS-CoV-2," Expert Review of Respiratory Medicine, pp. 1 – 4, 2020.
- [17] L. A. Vaira, G. Salzano, G. Deiana, and G. D. Riu, "Anosmia and ageusia: Common findings in COVID-19 patients," *The Laryngoscope*, vol. 130, pp. 1787 – 1787, 2020.
- [18] U. Food and D. Administration, "Accelerated emergency use authorization (EUA) summary COVID-19 RT-PCR test (Laboratory Corporation of America)," U.S. Food and Drug Administration, 2020.
- [19] W. Feng, A. M. Newbigging, C. Le, B. Pang, H. Peng, Y. Cao, J. Wu, G. Abbas, J. Song, D. Wang, M. Cui, J. Tao, D. L. Tyrrell, X.-E. Zhang, H. Zhang, and X. C. Le, "Molecular diagnosis of COVID-19: Challenges and research needs," *Analytical Chemistry*, 2020.
- [20] Y. Fang, H. Zhang, J. Xie, M. Lin, L. Ying, P. Pang, and W.-B. Ji, "Sensitivity of chest CT for COVID-19: Comparison to RT-PCR," *Radiology*, 2020.
- [21] J. P. Kanne, B. P. Little, J. H. Chung, B. M. Elicker, and L. H. Ketai, "Essentials for radiologists on COVID-19: An update—radiology scientific expert panel," *Radiology*, 2020.
- [22] L. M. Kucirka, S. A. Lauer, O. Laeyendecker, D. Boon, and J. Lessler, "Variation in falsenegative rate of reverse transcriptase polymerase chain reaction–based SARS-CoV-2 tests by time since exposure," *Annals of Internal Medicine*, 2020.

- [23] A. S. Fomsgaard and M. W. Rosenstierne, "An alternative workflow for molecular detection of SARS-CoV-2 – escape from the na extraction kit-shortage, Copenhagen, Denmark, March 2020," *Eurosurveillance*, vol. 25, 2020.
- [24] P. B. van Kasteren, B. van der Veer, S. van den Brink, L. A. Wijsman, J. de Jonge, A.-M. van den Brandt, R. Molenkamp, C. B. Reusken, and A. Meijer, "Comparison of seven commercial RT-PCR diagnostic kits for COVID-19," *Journal of Clinical Virology*, vol. 128, pp. 104412 – 104412, 2020.
- [25] H. Péré, I. Podglajen, M. Wack, E. Flamarion, T. Mirault, G. Goudot, C. Hauw-Berlemont, L. M. M. Lê, E. Caudron, S. Carrabin, J. Rodary, T. Ribeyre, L. Bélec, and D. Veyer, "Nasal swab sampling for SARS-CoV-2: A convenient alternative in times of nasopharyngeal swab shortage," *Journal of Clinical Microbiology*, vol. 58, 2020.
- [26] M. N. Esbin, O. N. Whitney, S. Chong, A. C. Maurer, X. Darzacq, and R. Tjian, "Overcoming the bottleneck to widespread testing: A rapid review of nucleic acid testing approaches for COVID-19 detection," *RNA*, vol. 26, pp. 771 – 783, 2020.
- [27] E. S. Amirian, "Prioritizing COVID-19 test utilization during supply shortages in the late phase pandemic," *Journal of Public Health Policy*, vol. 43, pp. 320 324, 2022.
- [28] A. C. of Radiology, "ACR recommendations for the use of chest radiography and computed tomography (CT) for suspected COVID-19 infection," *American College of Radiology*, 2020.
- [29] G. D. Rubin, C. J. Ryerson, L. B. Haramati, N. Sverzellati, J. P. Kanne, S. Raoof, N. W. Schluger, A. Volpi, J.-J. Yim, I. B. K. Martin, D. J. Anderson, C. E. Kong, T. A. Altes, A. Bush, S. R. Desai, J. G. Goldin, J. M. Goo, M. Humbert, Y. Inoue, H. U. Kauczor, F. Luo, P. J. Mazzone, M. Prokop, M. Rémy-Jardin, L. Richeldi, C. Schaefer-Prokop, N. Tomiyama, A. U. Wells, and A. N. C. Leung, "The role of chest imaging in patient management during the COVID-19 pandemic: A multinational consensus statement from the Fleischner Society," *Radiology*, 2020.
- [30] H. Y. F. Wong, H. Y. S. Lam, A. H.-T. Fong, S. T. Leung, T. W.-Y. Chin, C. S.-Y. Lo, M. M.-S. Lui, J. C. Y. Lee, K. W.-H. Chiu, T. W.-H. Chung, E. Y. P. Lee, E. Y. F. Wan, I. F. Hung, T. P. W. Lam, M. D. Kuo, and M. yen Ng, "Frequency and distribution of chest radiographic findings in patients positive for COVID-19," *Radiology*, vol. 296, 2020.
- [31] S. Kooraki, M. Hosseiny, L. A. Myers, and A. Gholamrezanezhad, "Coronavirus (COVID-19) outbreak: What the Department of Radiology should know," *Journal of the American College of Radiology*, vol. 17, pp. 447 – 451, 2020.
- [32] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun, and L. Xia, "Correlation of chest CT and RT-PCR testing in Coronavirus Disease 2019 (COVID-19) in China: A report of 1014 cases," *Radiology*, 2020.
- [33] D. L. Smith, J.-P. Grenier, C. Batte, and B. M. Spieler, "A characteristic chest radiographic pattern in the setting of the COVID-19 pandemic," *Radiology: Cardiothoracic Imaging*, vol. 2, 2020.
- [34] The British Society of Thoracic Imaging, "COVID-19 BSTI reporting templates." https://www.bsti.org.uk/COVID-19-resources/COVID-19-bsti-reporting-templates/, 2020.

- [35] S. Simpson, F. U. Kay, S. Abbara, S. Bhalla, J. H. Chung, M. S. Chung, T. S. Henry, J. P. Kanne, S. J. Kligerman, J. P. Ko, and H. I. Litt, "Radiological Society of North America expert consensus statement on reporting chest CT findings related to COVID-19. Endorsed by the Society of Thoracic Radiology, the American College of Radiology, and RSNA," *Journal of Thoracic Imaging*, 2020.
- [36] R. Subramanian, Q. He, and M. Pascual, "Quantifying asymptomatic infection and transmission of COVID-19 in New York City using observed cases, serology, and testing capacity," *Proceedings of the National Academy of Sciences*, vol. 118, 2020.
- [37] J. P. Kanne, H. X. Bai, A. Bernheim, M. S. Chung, L. B. Haramati, D. F. Kallmes, B. P. Little, G. D. Rubin, and N. Sverzellati, "COVID-19 imaging: What we know now and what remains unknown.," *Radiology*, p. 204522, 2021.
- [38] W. S. McCulloch and W. H. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bulletin of Mathematical Biology, vol. 52, pp. 99–115, 2021.
- [39] H. J. Kelley, "Gradient theory of optimal flight paths," ARS Journal, vol. 30, pp. 947–954, 1960.
- [40] S. E. Dreyfus, "The numerical solution of variational problems," *Journal of Mathematical Analysis and Applications*, vol. 5, pp. 30–45, 1962.
- [41] C. Berners-Lee, "Cybernetics and forecasting," 1968.
- [42] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.
- [43] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, 1989.
- [44] C. Cortes and V. N. Vapnik, "Support-vector networks," Machine Learning, vol. 20, pp. 273–297, 1995.
- [45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, pp. 1735–1780, 1997.
- [46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255, 2009.
- [47] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 2016.
- [48] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. J. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *ArXiv*, vol. abs/2005.14165, 2020.

- [49] J. Markoff, "Scientists see promise in deep-learning programs," The New York Times, 2012.
- [50] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *ArXiv*, vol. abs/1508.06576, 2015.
- [51] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034, 2015.
- [52] W. Zhou, Y. Yang, C. Yu, J. Liu, X. Duan, Z. Weng, D. Chen, Q. Liang, Q. Fang, J. Zhou, H. Ju, Z. Luo, W. Guo, X. Ma, X. Xie, R. Wang, and L. yao Zhou, "Ensembled deep learning model outperforms human experts in diagnosing biliary atresia from sonographic gallbladder images," *Nature Communications*, vol. 12, 2020.
- [53] A. Buetti-Dinh, V. Galli, S. Bellenberg, O. Ilie, M. Herold, S. Christel, M. Boretska, I. Pivkin, P. Wilmes, W. Sand, M. Vera, and M. Dopson, "Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition," *Biotechnology Reports*, vol. 22, 2019.
- [54] K. Lloyd, "Bias amplification in artificial intelligence systems," ArXiv, vol. abs/1809.07842, 2018.
- [55] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, 2013.
- [56] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *ArXiv*, vol. abs/1803.08375, 2018.
- [57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84 90, 2012.
- [58] J. Hadamard, "Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées," Mémoires présentés par divers savants étrangers à l'Académie des sciences de l'Institut de France, vol. 33, 1908.
- [59] R. Courant, "Variational methods for the solution of problems of equilibrium and vibrations," Bulletin of the American Mathematical Society, vol. 49, pp. 1–23, 1943.
- [60] B. Polyak, "Some methods of speeding up the convergence of iteration methods," Ussr Computational Mathematics and Mathematical Physics, vol. 4, pp. 1–17, 1964.
- [61] Y. Nesterov, "A method for solving the convex programming problem with convergence rate  $o(1/k^2)$ ," *Proceedings of the USSR Academy of Sciences*, vol. 269, pp. 543–547, 1983.
- [62] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," in *Journal of machine learning research*, 2011.
- [63] G. Hinton, "Neural networks for machine learning." https://www.cs.toronto.edu/~tijme n/csc321/slides/lecture\_slides\_lec6.pdf, 2012.
- [64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

- [65] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2010.
- [66] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis.," Annual review of biomedical engineering, vol. 19, pp. 221–248, 2017.
- [67] S. K. Zhou, H. Greenspan, C. Davatzikos, J. S. Duncan, B. van Ginneken, A. Madabhushi, J. L. Prince, D. Rueckert, and R. M. Summers, "A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises," *Proceedings of the IEEE*, vol. 109, pp. 820–838, 2020.
- [68] M. P. McBee, O. A. Awan, A. T. Colucci, C. W. Ghobadi, N. Kadom, A. P. Kansagra, S. Tridandapani, and W. F. Auffermann, "Deep learning in radiology," *Academic radiology*, vol. 25 11, pp. 1472–1480, 2018.
- [69] V. Gulshan, L. H. Peng, M. Coram, M. C. Stumpe, D. J. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. A. Cuadros, R. Kim, R. Raman, P. Nelson, J. L. Mega, and D. R. Webster, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316 22, pp. 2402–2410, 2016.
- [70] G. J. S. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.
- [71] A. Esteva, B. Kuprel, R. A. Novoa, J. M. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, pp. 115–118, 2017.
- [72] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *ArXiv*, vol. abs/1505.04597, 2015.
- [73] Y. Li, K. Li, C. Zhang, J. Montoya, and G.-H. Chen, "Learning to reconstruct computed tomography images directly from sinogram data under a variety of data acquisition conditions," *IEEE Transactions on Medical Imaging*, vol. 38, pp. 2469–2481, 2019.
- [74] H. Chen, Y. Zhang, M. K. Kalra, F. Lin, Y. Chen, P. Liao, J. Zhou, and G. Wang, "Low-dose CT with a residual encoder-decoder convolutional neural network," *IEEE Transactions on Medical Imaging*, vol. 36, pp. 2524–2535, 2017.
- [75] J. Montoya, C. Zhang, Y. Li, K. Li, and G.-H. Chen, "Reconstruction of three-dimensional tomographic patient models for radiation dose modulation in CT from two scout views using deep learning," *Medical physics*, 2021.
- [76] C. Zhang, Y. Li, and G.-H. Chen, "Accurate and robust sparse-view angle CT image reconstruction using deep learning and prior image constrained compressed sensing (DL-PICCS)," *Medical physics*, 2021.
- [77] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *International Journal* of Computer Vision, vol. 128, pp. 336–359, 2016.
- [78] T. Kadir and M. Brady, "Saliency, scale and image description," International Journal of Computer Vision, vol. 45, pp. 83–105, 2001.

- [79] Y. Zhou, S. Booth, M. T. Ribeiro, and J. A. Shah, "Do feature attribution methods correctly attribute features?," in *AAAI Conference on Artificial Intelligence*, 2021.
- [80] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. S. Zemel, W. Brendel, M. Bethge, and F. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, pp. 665 – 673, 2020.
- [81] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should i trust you?": Explaining the predictions of any classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
- [82] S. Beery, G. V. Horn, and P. Perona, "Recognition in terra incognita," *ArXiv*, vol. abs/1807.04975, 2018.
- [83] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study," *PLoS Medicine*, vol. 15, 2018.
- [84] L. Wynants, B. van Calster, M. J. Bonten, G. S. Collins, T. P. A. Debray, M. de Vos, M. C. Haller, G. Heinze, K. G. M. Moons, R. D. Riley, E. Schuit, L. J. M. Smits, K. I. Snell, E. W. Steyerberg, C. Wallisch, and M. van Smeden, "Prediction models for diagnosis and prognosis of COVID-19: Systematic review and critical appraisal," *The BMJ*, vol. 369, 2020.
- [85] M. Roberts, D. Driggs, M. Thorpe, J. D. Gilbey, M. Yeung, S. Ursprung, A. I. Avilés-Rivero, C. Etmann, C. McCague, L. Beer, J. R. Weir-McCall, Z. Teng, E. Gkrania-Klotsas, A. A. E. E. G. G. G. H. J. R. K. J. A. L. G. Ya, A. Ruggiero, A. Korhonen, E. Jefferson, E. Ako, G. Langs, G. Gozaliasl, G. Yang, H. Prosch, J. Preller, J. Stanczuk, J. Tang, J. Hofmanninger, J. L. Babar, L. E. Sanchez, M. Thillai, P. M. Gonzalez, P. Teare, X. Zhu, M. N. Patel, C. Cafolla, H. Azadbakht, J. Jacob, J. Lowe, K. Zhang, K. Bradley, M. Wassin, M. Holzer, K. Ji, M. D. Ortet, T. Ai, N. Walton, P. Liò, S. Stranks, T. Shadbahr, W. Lin, Y. Zha, Z. Niu, J. H. F. Rudd, E. Sala, and C.-B. Schönlieb, "Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans," *Nature Machine Intelligence*, vol. 3, pp. 199 – 217, 2020.
- [86] J. Born, D. J. Beymer, D. Rajan, A. Coy, V. V. Mukherjee, M. Manica, P. Prasanna, D. Ballah, M. Guindy, D. Shaham, P. L. Shah, E. Karteris, J. L. Robertus, M. Gabrani, and M. Rosen-Zvi, "On the role of artificial intelligence in medical imaging of COVID-19," *Patterns*, vol. 2, 2020.
- [87] A. J. DeGrave, J. D. Janizek, and S.-I. Lee, "AI for radiographic COVID-19 detection selects shortcuts over signal," *medRxiv*, 2020.
- [88] L. O. Teixeira, R. M. Pereira, D. Bertolini, L. Oliveira, L. Nanni, and Y. M. G. Costa, "Impact of lung segmentation on the diagnosis and explanation of COVID-19 in chest X-ray images," *Sensors (Basel, Switzerland)*, vol. 21, 2020.
- [89] P. Rouzrokh, B. Khosravi, S. Faghani, M. Moassefi, D. V. V. Garcia, Y. Singh, K. Zhang, G. M. Conte, and B. J. Erickson, "Mitigating bias in radiology machine learning: 1. data handling.," *Radiology. Artificial intelligence*, vol. 4 5, p. e210290, 2022.
- [90] IDC, "Worldwide IDC global datasphere forecast, 2021-2025." https://www.idc.com/getdoc.jsp?containerId=US49018922, 2021.

- [91] A. Y. Halevy, P. Norvig, and F. C. Pereira, "The unreasonable effectiveness of data," IEEE Intelligent Systems, vol. 24, pp. 8–12, 2009.
- [92] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
- [93] K. Y. E. Aryanto, M. Oudkerk, and P. M. A. van Ooijen, "Free DICOM de-identification tools in clinical research: Functioning and safety of patient privacy," *European Radiology*, vol. 25, pp. 3685 – 3695, 2015.
- [94] I. Pérez-Díez, R. Pérez-Moraga, A. López-Cerdán, J. M. Salinas-Serrano, and M. de la Iglesia-Vayá, "De-identifying spanish medical texts - named entity recognition applied to radiology reports," *Journal of Biomedical Semantics*, vol. 12, 2020.
- [95] C. Schaefer-Prokop and M. Prokop, "Chest radiography in COVID-19: No role in asymptomatic and oligosymptomatic disease," *Radiology*, 2020.
- [96] M. Körner, C. H. Weber, S. Wirth, K. J. Pfeifer, M. Reiser, and M. Treitl, "Advances in digital radiography: Physical principles and system overview.," *Radiographics: A review publication of the Radiological Society of North America, Inc*, vol. 27 3, pp. 675–86, 2007.
- [97] J. A. Rowlands, "The physics of computed radiography," *Physics in medicine and biology*, vol. 47 23, pp. R123–66, 2002.
- [98] E. Ozçete, B. Boydak, M. Ersel, S. Kıyan, I. Uz, and O. Cevrim, "Comparison of conventional radiography and digital computerized radiography in patients presenting to emergency department," *Turkish Journal of Emergency Medicine*, vol. 15, pp. 8 – 12, 2015.
- [99] A. Tafti and D. W. Byerly, *X-ray Radiographic Patient Positioning*. StatPearls Publishing, Dec. 2021.
- [100] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-Ray8: Hospital-scale chest X-Ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3462–3471, 2017.
- [101] A. E. W. Johnson, T. J. Pollard, S. J. Berkowitz, N. R. Greenbaum, M. P. Lungren, C. ying Deng, R. G. Mark, and S. Horng, "MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports," *Scientific Data*, vol. 6, 2019.
- [102] M. de la Iglesia-Vayá, J. M. Saborit, J. A. Montell, A. Pertusa, A. Bustos, M. Cazorla, J. Galant, X. Barber, D. Orozco-Beltrán, F. García-García, M. Caparrós, G. González, and J. M. Salinas, "BIMCV COVID-19+: A large annotated dataset of RX and CT images from COVID-19 patients," *ArXiv*, vol. abs/2006.01174, 2020.
- [103] Medical Imaging and Data Resource Center, "Medical Imaging and Data Resource Center," 2023. Accessed: 2023-03-15.
- [104] L. Wang, Z. Q. Lin, and A. Wong, "COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," *Scientific Reports*, vol. 10, 2020.
- [105] J. P. Cohen, P. Morrison, and L. Dao, "COVID-19 image data collection," *ArXiv*, vol. abs/2003.11597, 2020.

- [106] G. Maguolo and L. Nanni, "A critic evaluation of methods for COVID-19 automatic detection from X-ray images," *An International Journal on Information Fusion*, vol. 76, pp. 1 7, 2020.
- [107] E. Tartaglione, C. A. Barbano, C. Berzovini, M. Calandri, and M. Grangetto, "Unveiling COVID-19 from chest X-Ray with deep learning: A hurdles race with small data," *International Journal of Environmental Research and Public Health*, vol. 17, 2020.
- [108] B. G. S. Cruz, M. N. Bossa, J. Sölter, and A. D. Husch, "Public Covid-19 X-ray datasets and their impact on model bias – a systematic review of a significant problem," *Medical Image Analysis*, vol. 74, pp. 102225 – 102225, 2021.
- [109] J. F. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, pp. 679–698, 1986.
- [110] J. Jo and Y. Bengio, "Measuring the tendency of CNNs to learn surface statistical regularities," *ArXiv*, vol. abs/1711.11561, 2017.
- [111] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. Wichmann, and W. Brendel, "ImageNettrained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," *ArXiv*, vol. abs/1811.12231, 2018.
- [112] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *ArXiv*, vol. abs/1905.02175, 2019.
- [113] H. Wang, X. Wu, P. Yin, and E. P. Xing, "High-frequency component helps explain the generalization of convolutional neural networks," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8681–8691, 2019.
- [114] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2014.
- [115] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," arXiv preprint arXiv:1802.03426, 2018.
- [116] S. P. Lloyd, "Least squares quantization in PCM," IEEE Trans. Inf. Theory, vol. 28, pp. 129–136, 1982.
- [117] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability (L. M. L. Cam and J. Neyman, eds.), vol. 1, pp. 281–297, University of California Press, 1967.
- [118] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2015.
- [119] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *ArXiv*, vol. abs/1905.11946, 2019.
- [120] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422, 2008.
- [121] O. Pfungst and C. Rahn, Clever Hans: (the Horse of Mr. Von Osten.) A Contribution to Experimental Animal and Human Psychology. Holt, Rinehart and Winston, 1911.

- [122] Wikipedia contributors, "Clever Hans," 2004. [Online; accessed 2023].
- [123] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," J. Mach. Learn. Res., vol. 10, pp. 2137–2155, 2009.
- [124] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. M. Mooij, "On causal and anticausal learning," in *International Conference on Machine Learning*, 2012.
- [125] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," CVPR 2011, pp. 1521–1528, 2011.
- [126] A. Syrowatka, M. Kuznetsova, A. Alsubai, A. L. Beckman, P. A. Bain, K. J. T. Craig, J. Hu, G. P. Jackson, K. B. Rhee, and D. W. Bates, "Leveraging artificial intelligence for pandemic preparedness and response: A scoping review to identify key use cases," NPJ Digital Medicine, vol. 4, 2021.
- [127] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269, 2016.
- [128] J. W. Gichoya, I. Banerjee, A. R. Bhimireddy, J. L. Burns, L. A. Celi, L.-C. Chen, R. Correa, N. Dullerud, M. Ghassemi, S.-C. Huang, P.-C. Kuo, M. P. Lungren, L. J. Palmer, B. J. Price, S. Purkayastha, A. Pyrros, L. Oakden-Rayner, C. Okechukwu, L. Seyyed-Kalantari, H. Trivedi, R. Wang, Z. Zaiman, and H. Zhang, "AI recognition of patient race in medical imaging: A modelling study," *The Lancet. Digital health*, vol. 4, pp. e406 – e414, 2022.
- [129] C. G. Robinson, A. Trivedi, M. Blazes, A. Ortiz, J. Desbiens, S. Gupta, R. Dodhia, P. K. Bhatraju, W. C. Liles, A. Lee, J. Kalpathy-Cramer, and J. M. L. Ferres, "Deep learning models for COVID-19 chest X-ray classification: Preventing shortcut learning using feature disentanglement," *medRxiv*, 2021.
- [130] G. G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S.-I. Lee, "Improving performance of deep learning models with axiomatic attribution priors and expected gradients," *Nature Machine Intelligence*, vol. 3, pp. 620 – 631, 2020.
- [131] J. D. Viviano, B. Simpson, F. Dutil, Y. Bengio, and J. P. Cohen, "Underwhelming generalization improvements from controlling feature attribution," *ArXiv*, vol. abs/1910.00199, 2019.
- [132] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
- [133] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9992–10002, 2021.
- [134] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11966–11976, 2022.
- [135] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in NIPS, 2014.
- [136] R. Gulakala, B. Markert, and M. Stoffel, "Generative adversarial network based data augmentation for CNN based detection of COVID-19," *Scientific Reports*, vol. 12, 2022.

- [137] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. M. Al-turjman, and P. R. Pinheiro, "CovidGAN: Data augmentation using auxiliary classifier GAN for improved COVID-19 detection," *IEEE Access*, vol. 8, pp. 91916–91923, 2020.
- [138] P. Chambon, C. Bluethgen, J.-B. Delbrouck, R. van der Sluijs, M. Polacin, J. M. Z. Chaves, T. Abraham, S. Purohit, C. P. Langlotz, and A. Chaudhari, "RoentGen: Vision-language foundation model for chest X-ray generation," *ArXiv*, vol. abs/2211.12737, 2022.
- [139] P. Rajpurkar, J. A. Irvin, R. L. Ball, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Y. Ding, A. Bagul, C. Langlotz, B. N. Patel, K. W. Yeom, K. S. Shpanskaya, F. Blankenberg, J. Seekins, T. J. Amrhein, D. A. Mong, S. S. Halabi, E. Zucker, A. Ng, and M. P. Lungren, "Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists," *PLoS Medicine*, vol. 15, 2018.
- [140] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," *Drug discovery today*, vol. 23 6, pp. 1241–1250, 2018.
- [141] G. G. González-Hernández, A. Sarker, K. O'Connor, and G. K. Savova, "Capturing the patient's perspective: A review of advances in natural language processing of health-related text," *Yearbook of Medical Informatics*, vol. 26, pp. 214 – 227, 2017.
- [142] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, "Artificial intelligence in healthcare: Past, present and future," *Stroke and Vascular Neurology*, vol. 2, pp. 230 – 243, 2017.
- [143] U.S. Food and Drug Administration, "Artificial intelligence and machine learning (AI/ML) enabled medical devices," 2023. Accessed: 2023-04-09.
- [144] N. Naik, B. M. Z. Hameed, D. K. Shetty, D. Swain, M. Shah, R. Paul, K. Aggarwal, S. Ibrahim, V. Patil, K. Smriti, S. Shetty, B. P. Rai, P. Chłosta, and B. K. Somani, "Legal and ethical consideration in artificial intelligence in healthcare: Who takes responsibility?," *Frontiers in Surgery*, vol. 9, 2022.
- [145] S. Gerke, T. Minssen, and G. Cohen, "Ethical and legal challenges of artificial intelligencedriven healthcare," *Artificial Intelligence in Healthcare*, pp. 295 – 336, 2020.
- [146] M. J. Rigby, "Ethical dimensions of using artificial intelligence in health care," AMA Journal of Ethics, 2019.
- [147] K. Murphy, E. D. Ruggiero, R. Upshur, D. J. Willison, N. Malhotra, J. C. Cai, N. Malhotra, V. Lui, and J. L. Gibson, "Artificial intelligence for good health: A scoping review of the ethics literature," *BMC Medical Ethics*, vol. 22, 2020.
- [148] S. J. Vollmer, B. A. Mateen, G. Bohner, F. J. Király, R. Ghani, P. Jónsson, S. Cumbers, A. Jonas, K. S. L. McAllister, P. Myles, D. Grainger, M. Birse, R. Branson, K. G. M. Moons, G. S. Collins, J. P. A. Ioannidis, C. Holmes, and H. Hemingway, "Machine learning and artificial intelligence research for patient benefit: 20 critical questions on transparency, replicability, ethics, and effectiveness," *BMJ*, vol. 368, 2020.
- [149] T. Eche, L. H. Schwartz, F.-Z. Mokrane, and L. Dercle, "Toward generalizability in the deployment of artificial intelligence in radiology: Role of computation stress testing to overcome underspecification," *Radiology: Artificial Intelligence*, 2021.

- [150] J. Yang, A. A. S. Soltan, and D. A. Clifton, "Machine learning generalizability across healthcare settings: Insights from multi-site COVID-19 screening," NPJ Digital Medicine, vol. 5, 2022.
- [151] X. Liang, D. Nguyen, and S. B. Jiang, "Generalizability issues with deep learning models in medicine and their potential solutions: Illustrated with cone-beam computed tomography (CBCT) to computed tomography (CT) image conversion," *Machine Learning: Science and Technology*, vol. 2, 2020.
- [152] G. V. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, 1989.
- [153] K. Hornik, M. B. Stinchcombe, and H. L. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.
- [154] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a non-polynomial activation function can approximate any function," *New York University Stern School of Business Research Paper Series*, 1991.
- [155] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. C. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," in *International Conference on Machine Learning*, 2017.
- [156] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," ArXiv, vol. abs/1710.05468, 2017.
- [157] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *ArXiv*, vol. abs/1611.03530, 2016.
- [158] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," in NIPS, 2017.
- [159] M. Haque, A. K. Dubey, and J. Hinkle, "The effect of image resolution on automated classification of chest X-rays," in *medRxiv*, 2021.
- [160] B. J. Kuo, Y. K. Lai, M. L. M. Tan, and X.-Y. C. Goh, "Utility of screening chest radiographs in patients with asymptomatic or minimally symptomatic COVID-19 in Singapore," *Radiology*, vol. 298, no. 3, pp. E131–E140, 2021.
- [161] J. Hestness, S. Narang, N. Ardalani, G. F. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," *ArXiv*, vol. abs/1712.00409, 2017.
- [162] S. Bozinovski, "Reminder of the first paper on transfer learning in neural networks, 1976," *Informatica (Slovenia)*, vol. 44, 2020.
- [163] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in NIPS, 2014.
- [164] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International Conference on Artificial Neural Networks*, 2018.
- [165] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, pp. 43–76, 2019.

- [166] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "ImageNet-21K pretraining for the masses," ArXiv, vol. abs/2104.10972, 2021.
- [167] J. P. Cohen, J. D. Viviano, P. Bertin, P. Morrison, P. Torabian, M. Guarrera, M. P. Lungren, A. Chaudhari, R. Brooks, M. Hashir, and H. Bertrand, "TorchXRayVision: A library of chest X-ray datasets and models," in *Medical Imaging with Deep Learning*, 2022.
- [168] G. Shih, C. C. Wu, S. S. Halabi, M. D. Kohli, L. M. Prevedello, T. S. Cook, A. Sharma, J. Amorosa, V. A. Arteaga, M. Galperin-Aizenberg, R. R. Gill, M. C. B. Godoy, S. Hobbs, J. Jeudy, A. Laroia, P. N. Shah, D. R. Vummidi, K. Yaddanapudi, and A. Stein, "Augmenting the National Institutes of Health chest radiograph dataset with expert annotations of possible pneumonia.," *Radiology. Artificial intelligence*, vol. 1 1, p. e180041, 2019.
- [169] A. Bustos, A. Pertusa, J. M. Salinas, and M. de la Iglesia-Vayá, "PadChest: A large chest X-ray image dataset with multi-label annotated reports," *Medical image analysis*, vol. 66, p. 101797, 2019.
- [170] J. A. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. Langlotz, B. N. Patel, M. P. Lungren, and A. Ng, "CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison," *ArXiv*, vol. abs/1901.07031, 2019.
- [171] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *FAT*, 2018.
- [172] P. J. Phillips, C. A. Hahn, P. C. Fontana, D. A. Broniatowski, and M. A. Przybocki, "Four principles of explainable artificial intelligence," *Gaithersburg, Maryland*, p. 18, 2020.
- [173] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *IEEE Access*, vol. 8, pp. 42200–42216, 2019.
- [174] P. Rajpurkar, J. A. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Y. Ding, A. Bagul, C. Langlotz, K. S. Shpanskaya, M. P. Lungren, and A. Ng, "CheXNet: Radiologist-level pneumonia detection on chest X-Rays with deep learning," *ArXiv*, vol. abs/1711.05225, 2017.
- [175] N. Bien, P. Rajpurkar, R. L. Ball, J. A. Irvin, A. Park, E. Jones, M. D. Bereket, B. N. Patel, K. W. Yeom, K. S. Shpanskaya, S. S. Halabi, E. Zucker, G. S. Fanton, D. F. Amanatullah, C. F. Beaulieu, G. Riley, R. Stewart, F. Blankenberg, D. B. Larson, R. Jones, C. Langlotz, A. Ng, and M. P. Lungren, "Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of MRNet," *PLoS Medicine*, vol. 15, 2018.
- [176] F. Eitel and K. Ritter, "Testing the robustness of attribution methods for convolutional neural networks in MRI-based Alzheimer's disease classification," in *iMIMIC/ML-CDS@MICCAI*, 2019.
- [177] K. Young, G. Booth, B. Simpson, R. Dutton, and S. Shrapnel, "Deep neural network or dermatologist?," in *iMIMIC/ML-CDS@MICCAI*, 2019.
- [178] R. Sayres, A. Taly, E. Rahimy, K. Blumer, D. Coz, N. Hammel, J. Krause, A. Narayanaswamy, Z. Rastegar, D. J. Wu, S. Xu, S. M. Barb, A. Joseph, M. Shumski, J. M. Smith, A. B. Sood, G. S. Corrado, L. H. Peng, and D. R. Webster, "Using a deep learning algorithm and Integrated Gradients explanation to assist grading for diabetic retinopathy," *Ophthalmology*, vol. 1264, pp. 552–564, 2019.

- [179] A. Saporta, X. Gui, A. Agrawal, A. Pareek, S. Truong, C. Nguyen, V. D. Ngo, Jayne, D. Seekins, F. Blankenberg, A. Y. Ng, M. P. Lungren, and P. Rajpurkar, "Benchmarking saliency methods for chest X-ray interpretation," *Nature Machine Intelligence*, vol. 4, pp. 867 878, 2022.
- [180] N. T. Arun, N. Gaw, P. P. Singh, K. Chang, M. Aggarwal, B. Chen, K. Hoebel, S. Gupta, J. B. Patel, M. Gidwani, J. Adebayo, M. D. Li, and J. Kalpathy-Cramer, "Assessing the (un)trustworthiness of saliency maps for localizing abnormalities in medical imaging," *Radiology. Artificial intelli*gence, vol. 3 6, p. e200267, 2020.
- [181] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in International Conference on Machine Learning, 2017.
- [182] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," ArXiv, vol. abs/1705.07874, 2017.
- [183] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*, 2017.
- [184] R. J. Tomsett, D. Harborne, S. Chakraborty, P. K. Gurram, and A. D. Preece, "Sanity checks for saliency metrics," ArXiv, vol. abs/1912.01451, 2019.
- [185] R. Maclin and D. W. Opitz, "Popular ensemble methods: An empirical study," J. Artif. Intell. Res., vol. 11, pp. 169–198, 1999.
- [186] R. Polikar, "Ensemble based systems in decision making," IEEE Circuits and Systems Magazine, vol. 6, pp. 21–45, 2006.
- [187] L. Rokach, "Ensemble-based classifiers," Artificial Intelligence Review, vol. 33, pp. 1–39, 2010.
- [188] A. Mordvintsev, C. Olah, and M. Tyka, "DeepDream a code example for visualizing neural networks." https://research.googleblog.com/2015/06/inceptionism-going-deepe r-into-neural.html, 2015.
- [189] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems* 32, pp. 8024–8035, Curran Associates, Inc., 2019.
- [190] M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training," ArXiv, vol. abs/2104.00298, 2021.
- [191] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, pp. 1–54, 2019.
- [192] J. Patterson and A. Gibson, Deep learning: A practitioner's approach. O'Reilly Media, Inc., 2017.
- [193] L. N. Smith, "Cyclical learning rates for training neural networks," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 464–472, 2015.
- [194] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, pp. 1929–1958, 2014.