Deep Learning Techniques for the Radiological Imaging of COVID-19

by

Robert Hertel

A thesis

presented to Lakehead University

in partial fulfillment of the requirements for the degree of

Master of Science

in the Program of

Electrical and Computer Engineering

Thunder Bay, Ontario, Canada, 2021

EXAMINING COMMITTEE MEMBERSHIP

The following served on the Examining Committee for this thesis:

Supervisor: Dr. Rachid Benlamri

Professor, Dept. of Software Engineering, Lakehead University

Internal Member: Dr. Thangarajah Akilan

Assistant Professor, Dept. of Software Engineering, Lakehead University

Internal Member: Dr. Abdulsalam Yassine

Associate Professor, Dept. of Software Engineering, Lakehead University

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

ABSTRACT

The AI research community has recently been intensely focused on diagnosing COVID-19 by applying deep learning technology to the X-ray scans taken of COVID-19 patients. COVID-19 shares many of the same imaging characteristics as other common forms of bacterial and viral pneumonia. Differentiating COVID-19 from other common pulmonary infections, therefore, is a non-trivial task. While RT-PCR tests are the first viral tests commonly performed on COVID-19 patients, radiological tests are often reserved for further study of the illness in patients presenting with increased risk factors. To help offset what commonly requires hours of tedious manual annotation, our work uses Convolutional Neural Networks and other machine learning techniques to decrease the time radiologists spend interpreting COVID-19 radiological scans.

Deep learning experts commonly use transfer learning to offset the small number of images typically available in medical imaging tasks. Our first study's architecture included a deep neural network that was pretrained on over one hundred thousand X-ray images. We incorporated this architecture into two models with the purpose of diagnosing COVID-19. The experimental results demonstrate the robustness of our deep learning models, ultimately achieving sensitivities of 95% and 96% for our three-class and two-class models respectively.

To help further clarify the diagnosis of suspected COVID-19 patients, in our second study, we have designed a deep learning pipeline with a segmentation module and ensemble classifier. After performing a thorough comparative analysis, we demonstrate that our best model can successfully obtain an accuracy of 91% and a sensitivity of 92%. Following a detailed description of our deep learning pipeline, we present the strengths and shortcomings of our approach and compare our model with other similarly constructed models. Finally, we conclude with possible future directions for this research.

ACKNOWLEDGMENTS

First and foremost, I would like to express my most sincere gratitude to my advisor, Professor Rachid Benlamri, for his guidance and constructive suggestions throughout my study. I could not have completed my research without his invaluable input and support.

I also thank the other committee members, Professor Thangarajah Akilan, and Professor Abdulsalam Yassine for their invaluable feedback.

I wish to thank the Graduate Coordinator of the Electrical and Computer Engineering department, Professor Krishnamoorthy Natarajan, for his support and helpful advice.

I am also extremely grateful to the Natural Sciences and Engineering Research Council of Canada (NSERC), which was a significant funding source supporting my study (Canada Graduate Scholarship - CGS M) during 2020-2021.

Most importantly, I must thank my mother, Tracy Parsons, for providing me with much-needed support and guidance over the years. I would also like to thank all of my friends and family who have encouraged me to continue in my research.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

$1D$	One-Dimensional
$2D$	Two-Dimensional
$2.5D$	Two-and-One-Half-Dimensional
$3D$	Three-Dimensional
$AC-GAN$	Auxiliary Classifier Generative Adversarial Network
AI	Artificial Intelligence
ANN	Artificial Neural Network
$ARDS$	Acute Respiratory Distress Syndrome
AUC	Area Under The Curve
CAD	Computer-Aided Diagnosis
CAP	Community-Acquired Pneumonia
$CGAN$	Conditional Generative Adversarial Networks
CNN	Convolutional Neural Network
$COVID-19$	Coronavirus Disease 2019
CPU	Central Processing Unit
CT	Computer Tomography
CXR	Chest X-Ray
GAN	Generative Adversarial Networks

GGO	Ground-Glass Opacity
GPU	Graphics Processing Unit
$Grad-CAM$	Gradient-weighted Class Activation Mapping
ICU	Intensive Care Unit
$LASSO$	Least Absolute Shrinkage and Selection Operator
ML	Machine Learning
MLP	Multilayer Perceptron
PCA	Principal Component Analysis
RAM	Random Access Memory
$ReLU$	Rectified Linear Unit
RF	Random Forest
ROI	Region of Interest
$RT-PCR$	Reverse Transcriptase Polymerase Chain Reaction
$SARF$	Size Aware Random Forest
$SARS-CoV-2$	Severe Acute Respiratory Syndrome Coronavirus 2
TB	Tuberculosis
$t-SNE$	t-distributed stochastic neighbor embedding
$VRAM$	Video Random Access Memory
WHO	World Health Organization

Chapter 1

Introduction

1.1 COVID-19 Diagnositic Testing - An Urgent Need

The medical industry and researchers around the world have been urgently seeking new modalities to diagnose COVID-19. A lack of testing supplies in countries around the world has left many COVID-19 patients without a diagnosis, leading to the further spread of the illness. To help alleviate this exponentially growing need, deep learning researchers have been attempting to image COVID-19 with the use of radiological techniques. COVID-19 is the disease caused by SARS-CoV-2 and is an airborne illness that can be rapidly spread between individuals. The COVID-19 outbreak was officially recognized by the WHO as being the cause of a pandemic on March 11, 2020.

The AI research community has recently invested considerable time and resources into developing deep learning models based on chest radiographs for the purpose of diagnosing COVID-19. Many medical institutions are finding themselves in difficult positions when faced with countless numbers of patients presenting with symptoms of the illness. In Canada, there have been over 1.4 million reported COVID-19 cases and over 26,000 COVID-19 deaths since the start of the pandemic [1]. Many of these deaths may have been avoided if COVID-19 cases were detected sufficiently early and the disease was not allowed to spread rapidly. AI systems that can process chest radiographs hold promise as potential tools for fighting the coronavirus pandemic.

Recently deep learning techniques have come to permeate "the entire field of medical image analysis" [2]. With deep learning methodologies, AI researchers have made considerable progress in improving the quality of automated diagnostic medical imaging systems. Because of their pioneering work, many promising directions are now opening up that could potentially help diagnose COVID-19. Here in this work, we present two state-of-the-art X-ray-based deep learning models that can diagnose COVID-19. Both systems can perform as well as available molecular tests on the market and offer an independent means of testing COVID-19 patients.

1.2 Background

1.2.1 Competing Molecular Tests

There are several kinds of COVID-19 tests that are currently on the market. Molecular tests (polymerase chain reaction tests), Antigen tests (rapid tests), and antibody tests (blood tests) have seen widespread use. Of these three tests, the RT-PCR test is considered the present gold standard for diagnosing COVID-19 [3]. RT-PCR tests are not perfect, however, and reports have been made considering problems with the test's overall sensitivity [4]. Luo et al. [5] in a study including 4653 participants found that RT-PCR tests have a sensitivity of around 71%. Other studies have reported a range of sensitivities between 70 and 90 percent [6]. Kucirka et al. [7] in a John's Hopkins study reported that an RT-PCR test's sensitivity has wide variability over the 21 days after a patient is first exposed to SARS-CoV-2. They also noted that "although the false-negative rate is minimized 1 week after exposure, it remains high at 21%" [7]. Kucirka et al. [7] therefore ultimately found that it takes about a week from the time of symptom onset, for RT-PCR testing to deliver the lowest false-negative rate. This leaves room for other tests that may work better over the time that RT-PCR tests are less accurate. Radiological testing is a leading contender in the research community for such a scenario. Research has shown it to be useful over the time that a patient has obtained a negative RT-PCR test [8]. Radiological testing can therefore

be used in conjunction with other tests and possibly give more clarity regarding a patient's current diagnosis.

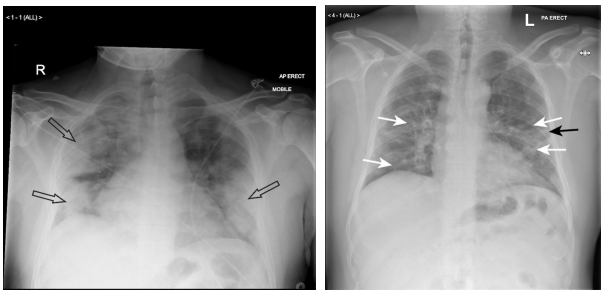
1.2.2 Identifying COVID-19 in Radiological scans

Before diving into the details of deep learning algorithms that may assist in diagnosing COVID-19, it is beneficial to first consider what imaging details radiologists have cited in determining a COVID-19 diagnosis. These image characteristics are of considerable importance during the process of validating COVID-19 deep learning models with saliency maps. A common feature of COVID-19 in radiological imaging includes bilateral GGOs with peripheral predominance [9]. A GGO is an infected pulmonary location in a radiological scan with increased attenuation. Song et al. [10] have additionally discovered that consolidation can commonly be observed in patients as the disease worsens. These consolidated areas in radiology represent regions where a patient's lung is filled with pus, liquid, and other materials that normally would not be present.

Song et al. have reported that "patients older than 50 years had more consolidated lung lesions than did those aged 50 years or younger." [10] Older patients, therefore, have clinical radiological evidence that shows they are at greater risk of negative health outcomes when they are infected. Cozzi et al. have likewise published research involving X-ray scans indicating that COVID-19 patients "show patchy or diffuse reticular–nodular opacities and consolidation, with basal, peripheral and bilateral predominance." [11] The same authors have additionally established that in cases where only one lung is infected, the right lung typically is more often affected. To obtain a visual appreciation for the manifestations of COVID-19 inside an infected patient's lungs, Fig. 1.1 shows the chest X-rays of two COVID-19 patients with some of the visual markers that have been discussed.

1.3 Objectives and Scope

The objective of our work is to build a diagnostic model capable of detecting COVID-19 with the use of chest radiographs. We originally also wanted to work on determining



(a) (b)

Figure 1.1: Lungs of 2 men with COVID-19 pneumonia in their 50s showing (a) bilateral consolidation and (b) GGOs (white arrows) and linear opacity (black arrow). [12]

the severity of a patient's disease with images and metadata. This was not practicable for reasons that will be explained in the literature review. There is a debate among researchers and medical professionals in the field concerning which radiological modality would best be suited for COVID-19 testing. Chest X-rays and thoracic computed tomographic scans are the most common modalities radiologists use in detecting COVID-19 related pneumonia in individuals. Both technologies have their merits and shortcomings. In comparing CXRs and CT scans, CXRs are generally less expensive and hence more widely used. This is especially true in developing countries where budgeting for a CT scanner can be more of a challenge. X-ray machines have another advantage over CT scanners in that they are commonly manufactured to be portable. They can be physically carted into ICUs and patients can remain in their physical location. There is also the question of how closely a radiologist should be exposed to suspected COVID-19 patients. X-rays require a single flat surface to be placed down onto a patient, while for a CT, a patient needs to be brought into a 3D enclosure and positioned properly within it. We initially considered both modalities when setting out to design a system capable of diagnosing COVID-19. We eventually settled on choosing X-rays as our chief modality in developing a deep-learning algorithm for diagnosing COVID-19. Our reasons for making this choice will be clear following our literature review in the following chapter.

Prior to beginning our work, many teams at universities and institutions around the world had already designed deep learning models for detecting COVID-19. Several highly

cited papers made claims of achieving extraordinarily high-performance metrics. Hemdan et al. [13], Rajaraman et al. [14], and Apostolopoulos and Mpesiana [15] all made claims of achieving common performance metrics (F1-Scores, Accuracies, Sensitivities, etc.) above 90 percent. Many of these papers on careful analysis, however, missed important details that allowed their classifiers to achieve overly optimistic results. Many papers incorporated a dataset from Kermany et al. [16] that contained the chest X-rays of young children suffering from various forms of bacterial and viral pneumonia. These X-rays had the effect of biasing a large number of COVID-19 datasets that were used in academic studies. There were additional dataset composition issues that existed in many studies where multiple scans of the same patients were mixed in their training, validation, and testing sets. The datasets of many studies additionally were quite small and prone to overfitting. Since these dataset issues were common to the majority of studies, we realized that there was a good opportunity to compete with the limited number of studies published so far that have constructed their datasets correctly. In our first publication [17], we focused on attempting to achieve higher COVID-19 sensitivities than other studies [18, 19, 20] without dataset composition issues.

In our second publication [21], we employed segmentation in preprocessing X-ray scans prior to classifying them. Segmentation can help to remove many of the irrelevant portions of an image that should not affect a classifier's final output. We have found that there are significantly fewer deep learning COVID-19 X-ray studies that perform segmentation. Many teams have likely shied away from performing segmentation due to the difficulty involved in creating a segmentation-classification pipeline. We experimented with several segmentation units and eventually worked towards optimizing the dice similarity coefficients of these units. We wanted to analyze any potential effect that segmentation might have on a classifier that is trained on X-ray scans. To create a more robust system, we also investigated the effect of ensembling multiple CNNs in our deep learning pipeline. Ultimately, we wanted to build a deep learning segmentation-classification pipeline that can achieve the highest possible evaluation metrics.

1.4 Study Limitations

Our research has experienced limitations in terms of the available data that can be used in training our deep learning models. Our final deep learning models were ultimately trained on datasets that contained three to four thousand COVID-19 lung X-rays. While this may seem like a significant number of images, in computer vision research, biomedical imaging experts prefer to have hundreds of thousands or even millions of images when training a deep network.

Current available public datasets also do not include the metadata that would be helpful in increasing the performance of our classifiers. Useful metadata would include information concerning a patient's age, sex, exposure location, bloodwork, etc. All of these metadata categories would not have only been useful diagnostically, but also may have been helpful in predicting a patient's prognosis. This lack of metadata accompanying COVID-19 radiographs has led us to focus on diagnosis alone.

1.5 Research Outline

Our research begins in chapter 2 with an in-depth literature review. In this literature review, we set out to discover the various deep learning methodologies that have been implemented to process X-ray and CT scans. A major objective of this section also includes finding the best possible datasets that can help with determining the diagnosis or prognosis of COVID-19 patients. Following the literature review, in chapter 3, we introduce the background information required for understanding how to design the deep learning systems that we modeled in chapters 4 and 5. This section provides an introduction to the structure of the neural network, CNN, and segmentation layers that are used in both studies. In chapter 4, we introduce our "COV-SNET" X-ray-based deep learning model which was built to diagnose COVID-19. To the best of our knowledge, the COV-SNET model has a higher sensitivity than all other deep learning models in the COVID-19 imaging literature not trained on improperly biased datasets. In this section, we also describe in detail how pretraining on related images can help when training a model on a smaller dataset. Our

work in chapter 5 takes a different approach to classifying COVID-19. In our second study, we have added segmentation as an extra preprocessing step prior to classifying COVID-19 images. We additionally show how ensembling can effectively be used to slightly increase the overall performance of a COVID-19 deep learning pipeline. Finally, in chapter 6, we conclude with our research studies' main findings and we discuss possible future directions for our work.

1.6 Publications Produced Throughout Master's Research Work

The following is a list of publications that were produced throughout the master's research work:

1. Deep Learning Techniques for COVID-19 Diagnosis and Prognosis Based on Radiological Imaging. [22] Sent to ACM Computing Surveys Dec. 21, 2020.
2. COV-SNET: A deep learning model for X-ray-based COVID-19 classification. [17] Published in Informatics in Medicine Unlocked on Jun. 03, 2021.
3. A Deep Learning Segmentation-Classification Pipeline for X-Ray-Based COVID-19 Diagnosis. [21] Published in Informatics in Medicine Unlocked on Jul. 06, 2021.

Chapter 2

Literature Review

2.1 Background

This literature review summarizes the current methods generated by the medical imaging AI research community which are focused on resolving lung imaging problems related to COVID-19. The literature reviewed here is a summary of the X-ray-based and CT-based AI COVID-19 detection systems that have been published so far. This literature review also contains a summary of current machine learning studies that have been focused on determining the prognosis of COVID-19 patients.

The current gold standard for diagnosing COVID-19 is the real-time RT-PCR test [3]. The test has been reported to suffer from sensitivity issues [4]. Depending on when an RT-PCR test is administered, the false-negative rate of an RT-PCR COVID-19 test can vary substantially. A John Hopkins led study found that RT-PCR tests elicit false-negative rates close to 100 percent at the time of symptom onset. This number falls to 61 percent by day 4. On the 8th day since COVID-19 symptoms onset, the false-negative rate drops to 26 percent. After this however, the false-negative rate increases to 61 percent on the 21st day [7]. This large variation in RT-PCR test accuracy leaves a lot of room for other tests that could be helpful over the time that RT-PCR tests are inaccurate. Medical imaging techniques are at times able to detect COVID-19 pneumonia when RT-PCR tests are unable to. This was especially true in China during the early days of the outbreak. CT scans are still used there

regularly to contain the illness [23]. In cases where patients suspected of having COVID-19 initially receive negative RT-PCR test results, radiography may be helpful [8].

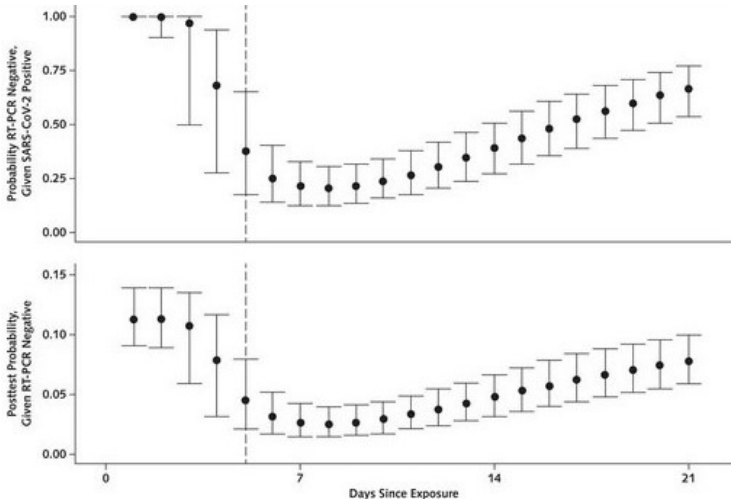


Figure 2.1: Variations in false-negative RT-PCR tests since a patient's time of exposure. [7]

2.2 A Choice Between Modalities

Chest X-Rays and thoracic computed tomographic scans are the conventional imaging modalities radiologists use to detect pneumonia in COVID-19 patients. Both technologies have their advantages and disadvantages. Healthcare practitioners would prefer to use contactless imaging workflows in the detection of pathologies associated with transmissible diseases. Imaging specialists and radiologists are highly important to medical institutions and all possible hazards that could expose them to COVID-19 need to be reduced. It is often not possible to fully reduce contact between radiologists and patients, depending on the imaging technique being used and the imaging protocol being followed.

Around the world, CXRs are more easily accessible than CTs and used more widely in low-income areas. X-ray machines are often portable and can be taken directly inside ICUs. X-rays may therefore be the first imaging modality doctors turn to when diagnosing COVID-19 patients. CT scans require a patient to be physically moved to a room with a

CT scanner. This leads to additional possible vectors of viral transmission within a medical institution. X-ray units are easier to clean and can be quickly disinfected for use on another patient. CT scans conversely require a patient to enter inside an imaging apparatus. After the necessary imaging is complete, the entire room and scanner need to be disinfected. The use of CT scans for diagnosing COVID-19 in patients varies between countries. In China, CT scans are used routinely, while the largest radiological societies in Western countries have published various opinion pieces suggesting that CT scans be used only in specific circumstances [24]. This concern in western countries mostly stems from concern over the transmission of COVID-19 in hospitals.

X-ray imaging may have fewer viral transmission issues in diagnosing COVID-19, but it also does not provide the same kind of high-resolution 3D imaging that a CT scanner does. X-rays produce 2D images of a chest pathology. CT scans produce a batch of 2D slices forming a 3D volume inside a patient. This increased imaging quality may be necessary for determining the nature and severity of a patient's illness. In the context of imaging COVID-19, the detection of pulmonary nodules and ground-glass opacity lesions are better detected across multiple CT slices in 3D information. A recent study reported that X-rays can often show normal in early or mild cases of COVID-19 [25]. The workflow for obtaining 3D scans from CT scanners is substantially different from how X-rays are performed for obtaining 2D lung images. During preparation for a CT scan, a patient is assisted by a technician to pose on a bed using a specific CT protocol. A CT scan takes place over a single breath-hold. The acquired data is thereafter processed and sent to an archive. There it can be analyzed by radiologists with the help of a computer-aided detection system for finding the patient's diagnosis or prognosis. CT scan protocols can take a long time to develop. Many intensities, resolutions, and patient positions are investigated for optimal imaging characteristics. CT scan protocols are still being fine-tuned in hospitals around the world for better COVID-19 detection. Beyond infection control issues there are a couple of other disadvantages in using CT scans for diagnosing COVID-19. One major disadvantage is that CT imaging is more costly than using CXRs. Another disadvantage is that the radiation dose a patient typically receives on a chest CT scan is typically 55 times higher than if a CXR was used [26]. Low-dose computed tomographic scans have been shown to reduce the radiation exposure of a

patient, but they are less effective than normal CTs at detecting the pulmonary features that are required in a COVID-19 diagnosis [27].

2.3 Division of COVID-19 Machine Learning Literature

There are several deep learning CAD systems that have been developed for screening and classifying COVID-19 patients using lung X-rays and CT scans. These systems fall into multiple categories in this review: system dimensionality, system purpose, deep learning methods/segmentation network, size/type of dataset, and system evaluation/results. The papers reviewed following this section will be presented in order of system dimensionality.

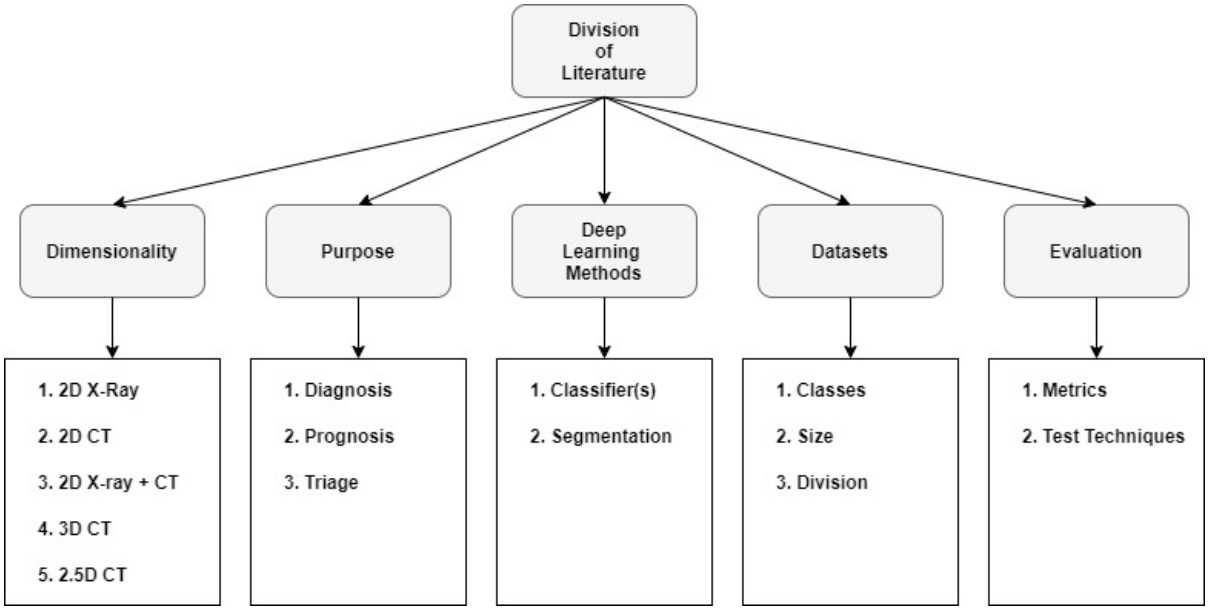


Figure 2.2: Division of summarized literature.

2.3.1 Dimensionality

The dimensionality of the data being processed by a computer vision system changes how an entire system is built. All X-ray CAD classifiers in the literature are based on 2D images. Many CT-based CAD classifiers in the literature are also 2D. They are often based on CNNs and operate over entire CT volumes using a slice by slice analysis. Most of the systems require preprocessing where images/volumes are normalized in terms of intensity and volume. A preprocessing step may also include removing images that have obstructions or have a closed lung. The form of preprocessing required in the system can change depending on the system's dimensionality. Many systems use 3D classifiers for CT-based CAD systems. 3D classifiers can be extremely data intensive. Adding a third dimension to a CNN adds more parameters to the model and increases the number of operations that are computed during implementation. The last kind of CT system that is used in the COVID-19 AI medical imaging literature is a 2.5D classification system. A 2.5D system in this literature review includes any system that somehow combines 2D and 3D algorithmic elements into a hybrid system. These hybrid systems often can reduce the computational cost incurred by a 3D system, while maintaining the same system accuracy. 2D and 3D segmentation models are also often combined with the 2D, 2.5D, and 3D classifiers in the studies reviewed here.

2.3.2 Purpose

The purpose of designing AI CAD systems varies between studies. Some studies are concerned primarily with developing a system for the detection or diagnosis of COVID-19. In some COVID-19 detection studies, the authors build systems for detecting COVID-19 alone. Other COVID-19 detection studies build computer vision systems for differentiating COVID-19 patients from patients suffering from other kinds of pneumonia. Some of the studies primarily interested in diagnosis perform binary classification, while other studies perform multiclass classification for helping to sort through multiple pathologies. Many studies are interested in developing imaging tools for the management of patients within a hospital. Creating image-based biomarkers can be an important step in determining the severity of a patient's illness. This may or may not be combined with predictive modeling

techniques that combine imaging data with other metadata (age, sex, patient history, etc.) to get a better picture of a patient's prognosis.

2.3.3 Deep Learning Methods

There are many classifiers used in the COVID-19 imaging literature, but the majority of classifiers tend to be CNNs. This is not a surprise given the recent performance results CNN models have achieved in many imaging competitions and real-life applications around the world. Various classes of CNNs mentioned in the COVID-19 imaging literature include AlexNets [28], ResNets [29], VGGs [30], DenseNets [31], SqueezeNets [32], EfficientNets [33] and Inception Networks [34]. Multilayer perceptrons, generative adversarial networks [35], and encoder-decoder systems are also used in many models. Some machine learning techniques are used in addition to deep learning techniques in a few of the reviewed papers. Random forest classifiers feature prominently in a couple of papers.

Segmentation is relevant to the study of 2D and 3D medical scans in that it can assist a learning algorithm by reducing the total area of a scan to analyze. It is a critical procedure in many medical imaging tasks. Lung segmentation is used to isolate regions of interest from other regions that have less useful spatial information. It can help a classifier focus on the regions that matter the most for diagnosing COVID-19. In problems with high spatial complexity, this reduction in superfluous data is very important. The main region of interest in lung segmentation algorithms at times only includes the lungs. At other times the lesions, nodules, and lobes in a COVID-19 patient's lungs are segmented out as well. Many of the authors in the studies reviewed here believe segmentation to be a necessary prerequisite component in COVID-19 diagnosis/prognosis systems.

Segmentation is not used as often in X-ray systems compared with CT systems in the COVID-19 detection literature. This may be because frontal 2D scans offer unique challenges that CT scans can avoid. 2D X-ray lung segmentation is a difficult task because a patient's ribs present image contrast problems. This is due to their composition and position in front of the lungs. There are however some studies in this review that perform lung segmentation on X-rays.

2.3.4 Datasets

Many papers have obtained datasets from hospitals or private institutions that are not public. Others have been using publicly available datasets. Most of the papers reviewed mention the sizes of their datasets and inform readers whether data augmentation methods were used. Since COVID-19 is such a new disease, datasets tend to be small (in the range of hundreds or thousands of images). Finding annotated images is difficult. Data augmentation methods are very important to help offset the lack of data in many computer vision tasks. Many papers reviewed here have also reported needing to construct systems that correct for extremely imbalanced datasets.

2.3.5 Evaluation

The results reported in the COVID-19 computer vision imaging literature typically are based on several evaluation metrics. While accuracy is almost always discussed, sensitivity, specificity, F1-score, and area under the receiver operating characteristic curve are all also metrics used in the literature. The literature for the most part always includes what kind of validation methodology the authors used. Below are a few of the equations used in calculating the most popular metrics in the reviewed literature:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.1}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.2}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.3}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

$$Recall = \frac{TP}{TP + TN} \tag{2.5}$$

$$NPV = \frac{TN}{TN + FN} \tag{2.6}$$

$$PPV = \frac{TP}{TP + FP} \tag{2.7}$$

$$F1 - Score = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall} \tag{2.8}$$

2.4 Interpreting COVID-19 in CXRs and CTs

The data required for interpreting COVID-19 in a deep learning algorithm needs to be taken from radiologists directly. This is especially important when researchers are validating their deep learning systems by using saliency maps. The main features of COVID-19 are mentioned here for easy reference. For CTs, radiologists have found that COVID-19 typically has bilateral GGOs with peripheral predominance [9]. A GGO in radiology is an area with increased attenuation during a scan of a patient's lungs. It is a region of hazy lung radiopacity. Song et al. [10] have found that consolidation is more common while the disease progresses. Consolidation combined with GGOs is found in the majority of patients. Consolidation in radiology represents the replacements of normal air in the lungs with fluid, pus, and other substances. The vast majority of patients have bilateral lung involvement [36]. Their infections are often in the posterior part of the lungs and can be seen peripherally as well. Song et al. have also found that "patients older than 50 years had more consolidated lung lesions than did those aged 50 years or younger." [10] Age therefore also plays a role in how the CT images present in the radiological findings. Using CXRs, Cozzi et al. [11] have published research proving that they "show patchy or diffuse reticular–nodular opacities and consolidation, with basal, peripheral and bilateral predominance." Cozzi et al. [11] have

shown that if the infection is in one lung alone that the right lung typically is more often infected. The same authors and Guan et al. [37] have shown that the consolidations, GGOs, and interstitial abnormalities in COVID-19 CTs also appear in COVID-19 X-rays.

2.5 Reviewed Computer Vision COVID-19 Studies

2.5.1 X-ray Studies

The X-ray studies reviewed here all use 2D classification techniques for diagnosing COVID-19. The ideas in the 2D X-ray studies below are all mostly applicable to 2D and 2.5D CT imaging applications as well.

Zhang et al. [38] have released a paper where they have built "a confidence-aware anomaly detection (CAAD) model to distinguish viral pneumonia cases from non-viral pneumonia cases." It does not specifically diagnose COVID-19 but helps healthcare providers eliminate potential pathologies from a patient's final diagnosis. Due to the extremely small datasets presently available for viral pneumonia, an extreme class-imbalance exists. Zhang et al. use anomaly detection on the X-VIRAL dataset. The dataset contains 37393 individuals with non-viral pneumonia and 5977 individuals with viral pneumonia. The datasets used in this study were in-house datasets. Rather than using binary classification, they use anomaly detection, which is a one-classification approach. An anomaly detection module is used that assigns every X-ray an anomaly score. The viral COVID-19 pneumonia scores should ideally be much higher than the negative samples that are used to train the system. Their system uses a 2D EfficientNet that has been pre-trained on ImageNet for feature extraction. The feature extractor forks out into "an anomaly detection module, and a confidence prediction module." [38] The anomaly detector has no error correction mechanism but uses the confidence aware module to predict when the anomaly detection module will fail. With no previous training on COVID-19 images, on a small COVID-19 dataset (X-COVID), the system achieves a sensitivity of 0.717 and an AUC of 0.836. No mention of image segmentation is found in the study although gradient-weighted class activation maps [39] are being used to ensure the virus is being localized in the lungs.

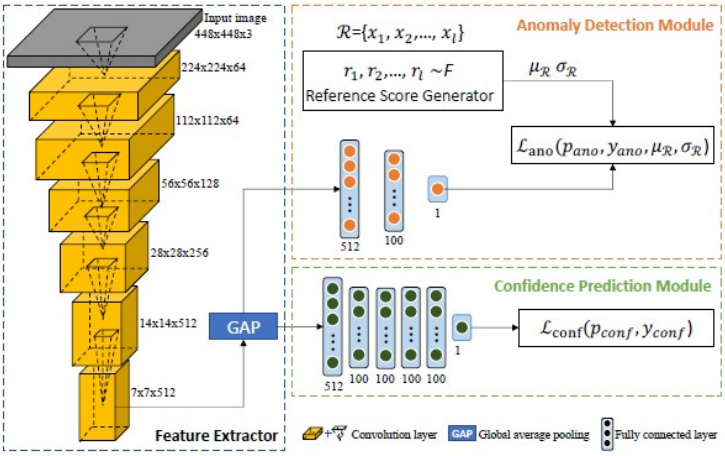


Figure 2.3: Zhang et al.'s [38] hybrid classifier and anomaly detection system.

Hemdan et al. [13] have designed COVIDX-Net to diagnose COVID-19 using X-ray images. The designers of COVIDX-Net compare seven 2D off-the-shelf architectures and compare these pre-trained architectures using the same training and test methods. The training and test sets are distributed in an 80 percent - 20 percent split. The paper does not explain how the off-the-shelf models used by the authors have been trained. It does not explain what CNN layers were frozen during transfer learning or if the CNNs were trained in a fully end-to-end fashion using the ImageNet weights for initialization. It mentions that data augmentation was not used in the study. They achieved their best results with the VGG-19 and DenseNet-201 architectures achieving F1-scores of .89 and .91 respectively. Many of the other models they used however also obtained high F1-scores. It is unsurprising their paper achieved their best results with a DenseNet-201 architecture. That network is the deepest off-the-shelf network the authors tested. Near the beginning of the pandemic, there were multiple instances of teams rushing to find 2D classifiers that can be quickly applied to diagnosing COVID-19 patients. While the COVID-19 patient vs. non-COVID-19 patient distinction is important, it is a non-trivial task to differentiate COVID-19 from other possible lung pathologies. This work only scratched the surface of the kind of classifiers that would be needed in a true clinical setting. The biggest criticism of this paper is that their dataset was likely too small to generate truly meaningful results.

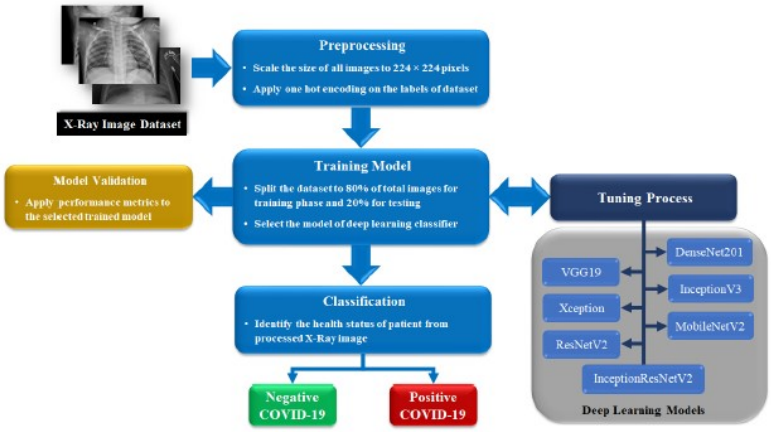


Figure 2.4: Hemdan et al.'s [13] various off-the-shelf-models approach

Apostolopoulos and Mpesiana [15] took a similar approach to Hemdan et al. [13] in terms of trying as many standard off-the-shelf classifiers as possible. Their dataset was composed of 224 COVID-19 patients, 714 bacterial/viral pneumonia patients, and 504 normal patients. Their best model was found to be a VGG-19 with a 2-class (COVID vs. non-COVID) accuracy of 98.5 percent and a 3-class (COVID vs. pneumonia vs. normal) accuracy of 93.48 percent. This work explains which layers were frozen to perform transfer learning. Using 3 classes was a better approach than Hemdan et al. [13] used in that it is very important to distinguish COVID-19 from other forms of pneumonia in a clinical setting. Both Hemdan et al. [13] and this paper could have improved their performances if they had initialized their weights using modality-specific pretraining in their systems. CNNs that have been trained to detect other lung pathologies with the use of X-rays would have been better models to use while applying transfer learning. Instead, both papers focused on just using ImageNet pretrained models. Both papers report research done at the beginning of the COVID-19 pandemic and provided proof of concept in showing the capability of deep learning models detecting COVID-19 with X-rays. Apostolopoulos and Mpesiana [15] made the mistake of using Kermany et al.'s [16] pneumonia dataset of children between the ages of one to five years old. Papers that use this dataset we have found tend to report unrealistic evaluation metrics.

Ozturk et al. [40] cited Hemdan et al. [13] as proof that further research was required that was focused on X-rays as a modality for diagnosing COVID-19. Their article is widely cited for researching whether a DarkNet19 classifier [41] could be a good candidate for transfer learning when performing COVID-19 disease classification with X-rays. DarkNet first gained popularity as being the classifier used in the real-time object detection model YOLO (you only look once) [42]. The paper is mentioned here because DarkNet classifiers have not been researched by any other studies as a CNN that can be used for diagnosing COVID-19. The learning model the authors used had a total of 1,164,434 parameters. The authors used an extraordinarily skewed dataset with 127 COVID-19 X-rays, 500 pneumonia X-rays, and 500 normal X-rays. While training their system this had the effect of the team seeing their model's training loss increase during the first epochs of training. That training loss eventually decreased and stabilized closer to 100 epochs after the network had been exposed to all the samples repeatedly. Their system could have used a segmentation network, but without it still achieved good binary classification in detecting COVID-19 with an accuracy of 98.08 percent. When adding the extra pneumonia category however the network's performance degraded as it struggled to differentiate COVID-19 from common pneumonia and ARDS. A trained radiologist examined the team's saliency maps and found that the model is not useful in this regard. The model is very good at diagnosing pneumonia, but otherwise makes errors. The model's accuracy for all three classes was reported to be 87.02 percent.

Haghanifar et al. [43] and Mangal et al. [44] have created two thoroughly validated CAD systems for diagnosing COVID-19. Their articles are the two main examples in the COVID-19 deep learning literature that use the ChexNet model [45] for transfer learning. ChexNet is a model that has been made famous in past X-ray classification competitions for diagnosing 14 different pathologies. Both papers also use ChexNet's pretrained weights for transfer learning. This is a reasonable pretrained model to use as ChexNet has been designed with a DenseNet-121 architecture and has been trained on over 200,000 X-rays. Hagnifar et al. [43] designed their model for diagnosing and differentiating between COVID-19 patients, normal patients, and CAP patients. Both binary and trinary classifiers are built to predict these categories. Their system has a large dataset with 780 images of COVID-19 patients,

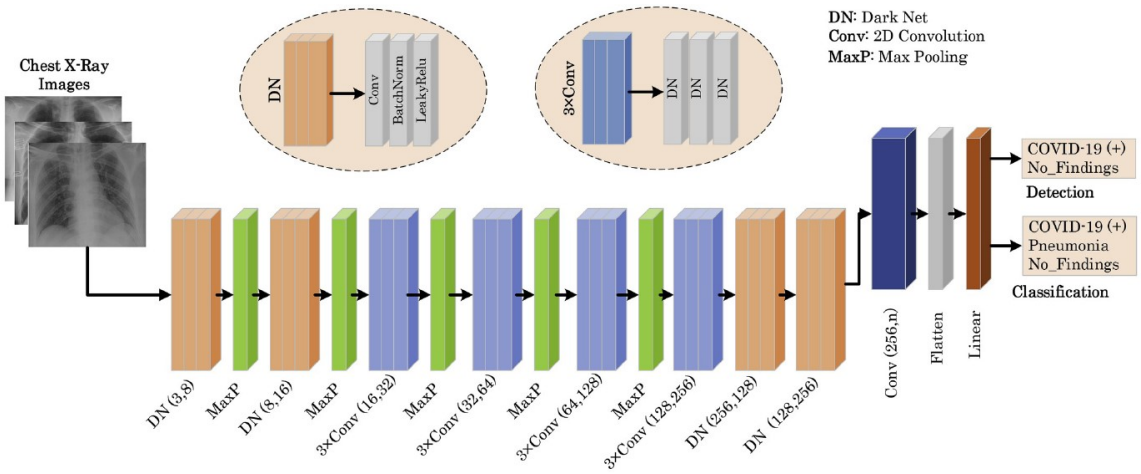


Figure 2.5: Ozturk et al.'s [40] various off-the-shelf-models approach.

5000 normal CXRs and 4600 images of CAP patients. These classes are later weighted in the loss function of the models they design to deal with the class imbalance. The authors show how the ChexNet model they use attains high accuracy metrics while localizing incorrect features in the patients it diagnoses. They show this with the use of Grad-CAMs. They go on to take the ChexNet model, add data augmentation and image segmentation (U-Net-based [46]) and achieve improved results. A knowledge of how to read CXRs and the use of Grad-CAMs is shown to be necessary within their paper. This is because a system can achieve high accuracies while not generalizing correctly. This paper is a significant improvement compared with some of the previous papers in this regard. Their final models for binary classification (COVID-19 vs. non-COVID-19) and trinary classification (COVID-19 Pneumonia vs. Normal vs. CAP) achieve f1-scores of 0.94 and 0.85 respectively. A deficiency in this model was that it used a dataset from Kermany et al. which contains "5,232 chest X-ray images from children" [16]. The dimensions of the lungs of these X-rays therefore likely caused their final classifier to produce unpredictable results. Mangel et al. [44] took the same approach as Haghanifar et al. [43] but split their data into four diagnosis groups: normal (1583 patients), bacterial pneumonia (2780 patients), viral pneumonia (1493 patients), and COVID-19 (155 patients). They also made the mistake of inluding Kermany et al.'s [16] dataset in their study. Mangel et al. [44] did not use a segmentation unit and obtained a 3-class accuracy of 90.5 percent and a 4-class accuracy of 87.2 percent. These

results show that differentiating between more types of pneumonia can be a more difficult task for CNNs trained on X-rays. Mangel et al.'s [44] results were later validated using saliency maps to ensure their system was localizing lung infections correctly. Al-Waisy et al. [47] likewise published a paper using a ChexNet model and made the same mistake as Haghanifar et al. [43] and Mangel et al. [44] in using Kermany et al.'s [16] dataset to train their model. They achieved a two-class accuracy of 99.99 percent, f1-score of 99.99 percent, and sensitivity of 99.98 percent. Unfortunately, the use of Kermany et al.'s [16] dataset is widespread and this has created a major flaw in all of these ChexNet models.

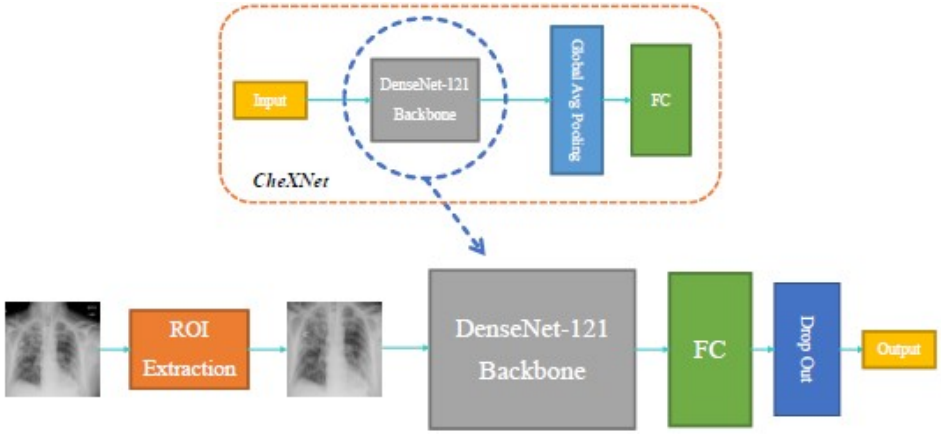


Figure 2.6: Haghanifar et al.'s [43] use ChexNet for pretrained weights and architecture.

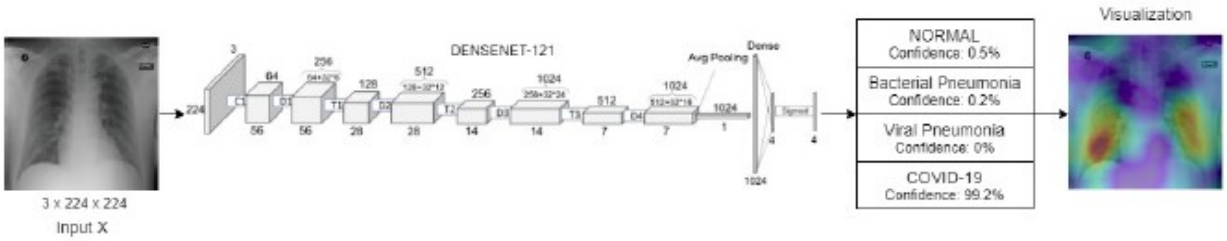


Figure 2.7: Mangal et al.'s [44] use ChexNet for pretrained weights and architecture.

Khalifa et al. [48] proposed using a GAN as a form of data augmentation for increasing the accuracy of an X-ray classifier that was designed to diagnose patients with pneumonia (it

cannot differentiate COVID-19 pneumonia from normal pneumonia). A GAN is composed of both a generator and a discriminator. The discriminator determines whether a sample belongs to a false distribution. The generator attempts to deceive the discriminator by generating false image distributions. The generative network designed by the team had 5 transposed convolutional layers, 4 ReLU layers, 4 batch normalization layers, and a final Tanh layer. The discriminator network designed by the team had 5 convolutional layers, 4 leaky ReLU layers and 3 batch normalization layers. The authors of this paper mention that the GAN helped them overcome their overfitting problem and increased the size of their dataset by a factor of 10. Their dataset contains 5863 patients that are separated into normal and pneumonia categories. Their research used only 624 images to prove the efficacy of their technique. The team experimented with several popular deep learning models but eventually settled on a ResNet-18 as their final classifier of choice because the total system combined with a ResNet-18 achieved an accuracy of 99 percent. The team disappointingly did not perform any saliency analysis on their final system or use a segmentation unit. Unfortunately, Khalifa et al. [48] made the mistake many others have of using Kermany et al.'s [16] pneumonia dataset to train their model. This paper was reviewed because it was the earliest paper made available to the research community where the authors used a GAN in their machine learning system for diagnosing COVID-19 related pneumonia.

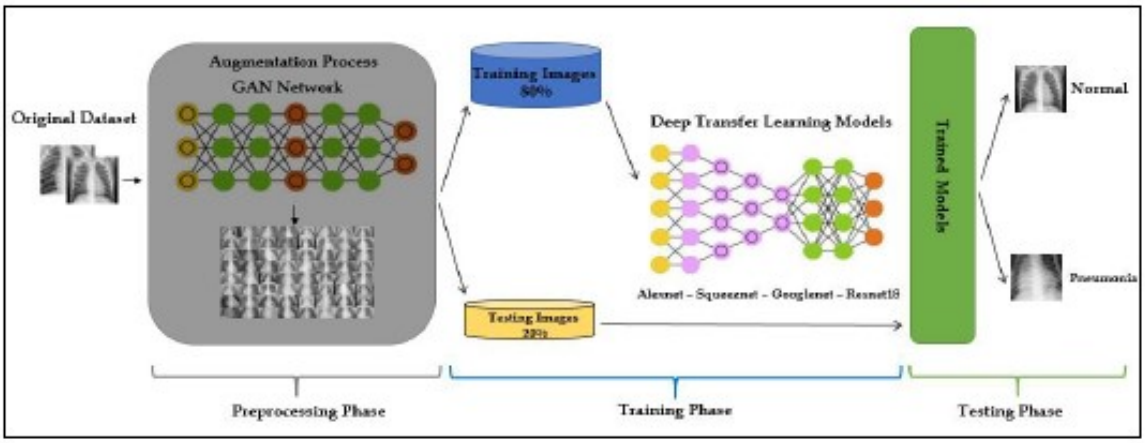


Figure 2.8: Khalifa et al.'s [48] GAN for data augmentation together with ResNet-18.

Waheed et al. [49] designed a system using a GAN and released a paper describing their system about a month after Khalifa et al. [48] published their paper. Their dataset consisted of 403 COVID CXRs and 721 normal CXRs. The team settled on using a VGG-16 CNN for classification. The system's implementation is different from Khalifa et al. [48] in that they use an auxiliary classifier generative adversarial network architecture to generate synthetic images as opposed to using a regular GAN. AC-GANs rely on the idea of a conditional GAN, which inputs prior information to a GAN in the form of a class label. An AC-GAN builds on the idea of a CGAN by tasking the discriminator with reproducing the class label input. A simple example can be used to illustrate this. If a generator receives a class label to generate a cat, the discriminator not only has to predict whether the image is real or fake, but it also must label the generated image as a cat. This idea turns out to improve the quality of generated images compared to a normal GAN. An interesting property of ACGANs is that when ACGANs work with higher resolution images (ex: 64x64 to 128x128) they perform better. The team first uses a VGG-16 classifier on the original dataset obtaining an accuracy of 85 percent. They then go on to use their ACGAN and generate 1399 synthetic normal CXRs and 1669 synthetic COVID-19 CXRs. All of these images are then used to train and test the VGG-16 classifier and its COVID-19 detection accuracy is increased to 95 percent. This suggests that the synthetic images are helping the CNN find meaningful features for diagnosing patients correctly. This study has the same limitations as the last study in that saliency maps and a segmentation unit were not utilized. Waheed et al. [49] like Khalifa et al. [48] made the mistake of using Kermany et al.'s [16] pneumonia dataset of children between the ages of one to five years old. The authors of this study interestingly used PCA visualization for visualizing the synthetic vs. non-synthetic image distributions so as to see how the real and synthetic categories cluster. The team used an RTX 2060 GPU in this project but was limited to working with 112x112 images likely due to their GPU's memory constraints.

Oh et al. [50] have developed a system that uses patch-wise disease probabilities to generate a global saliency map. The model uses a patch-based CNN to provide interpretable saliency maps that can be used for COVID-19 diagnosis and patient triage. The team's 4 class dataset included 200 viral pneumonia (includes COVID-19) patients, 54 bacterial pneumonia

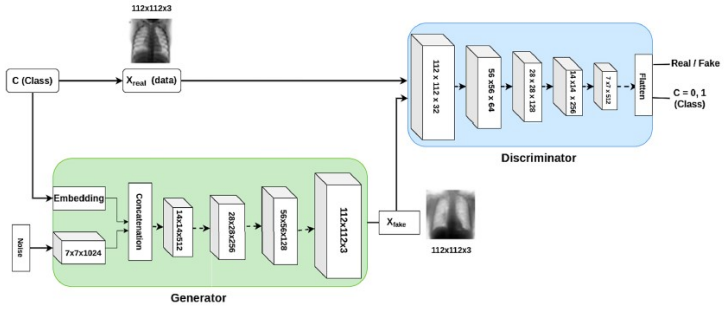


Figure 2.9: Waheed et al. [49] constructed this ACGAN for producing synthetic X-rays.

patients, 57 tuberculosis patients and 191 normal patients. The system initially uses a lung segmentation network developed using a fully convolutional DenseNet-103. An ImageNet pretrained ResNet18 is used on the segmented lungs for classification. An additional 100 patches are extracted from different parts of the segmented lungs and each of those patches is input through a separate ResNet18. The number of patches was selected to cover the number of lung pixels in each X-ray multiple times. The team afterward created a global saliency map with all of the patches using aggregated Grad-Cams. Each patch had a different COVID-19 score so the patch-wise saliency maps needed to be weighted with the probability of each disease class. The authors show that the probabilistic patches method they used allowed the team to differentiate multifocal lesions more accurately. The system using a local patch-based model achieved a total accuracy of 88.9 percent and COVID-19 sensitivity of 92.5 percent.

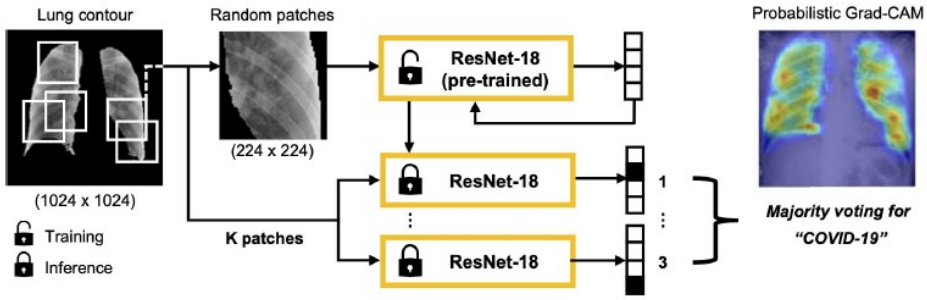


Figure 2.10: Oh et al.'s [50] Patch-Wise disease probability/saliency map model.

Wang et al. [51] designed a custom CNN they called "COVID-Net" for the purpose of diagnosing COVID-19. They did so by designing the custom residual network shown in Fig. 2.11. On March 22, 2020, when they released their paper, their system generated a lot of excitement regarding the possibility of using AI to diagnose COVID-19. The authors understood that the system they designed was by no means production ready but they did demonstrate promising results that lead to their work being widely cited and eventually published in nature. They generated their dataset using 13975 CXR images from 13780 patients. Their dataset consisted of several X-ray collections that contained in total 358 COVID-19 CXRs, 8066 normal CXRs, and 5538 CXRs with non-COVID-19 pneumonia. Interestingly, they mentioned their motivation for building their system was to help clinicians "better decide not only who should be prioritized for PCR testing for COVID-19 but also which treatment strategy to employ depending on the cause of infection, since COVID-19 and non-COVID19 infections require different treatment plans." [51] Three reasons they cite for the performance of their system was that it had a lightweight design pattern, selective-long-range connectivity and architectural diversity. The lightweight design pattern they used was discovered using a "machine-driven design exploration strategy" [51] that uses generative syntheses [52]. This design exploration strategy was being researched by one of the authors before the pandemic and is used to generate efficient deep neural networks automatically. Importantly, they also audited their system using an explainability method called GSInquire [53]. They found that their system was localizing COVID-19 features correctly and not being deceived by random peripheral objects in the X-ray images. Overall, their paper achieved an accuracy of 93.3 percent and a COVID-19 sensitivity of 91 percent.

Rajaraman et al. [14] proposed a method of building iteratively pruned ensembles of CNNs for diagnosing COVID-19. Their method of model building takes advantage of several ideas that papers for X-rays and CT slices have commonly not used in the existing COVID-19 deep learning literature. The authors built several popular pretrained CNNs and trained them on a separate lung X-ray task (a modality-specific task) for which more data is available. They used transfer learning on all these models to train them on binary classification (COVID vs. non-COVID) and multiclass classification (COVID vs bacterial pneumonia vs. normal). Their dataset had 7595 normal patient images, 2780 bacterial

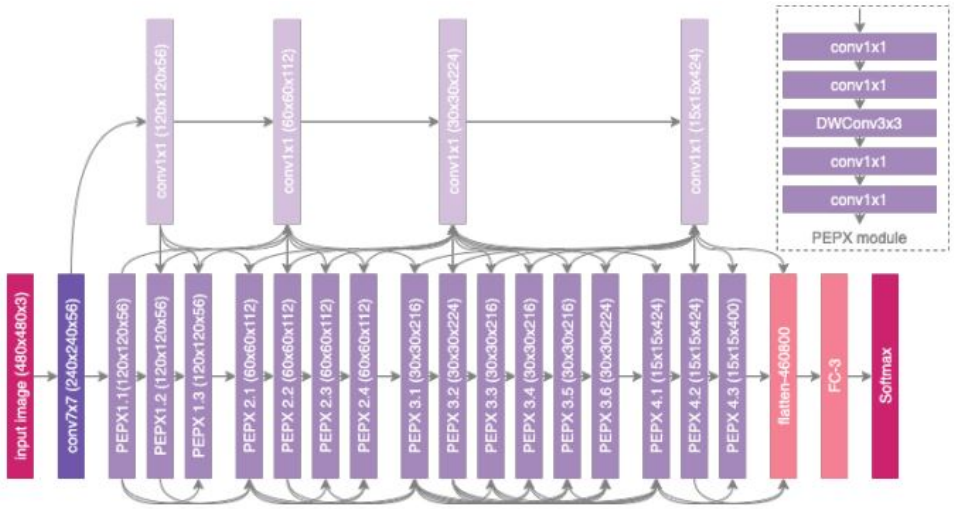


Figure 2.11: Wang et al's [51] "COVID-Net" model architecture.

pneumonia patient images, and 313 COVID-19 pneumonia images. To use fewer parameters and still maintain or even improve their CNNs accuracy the team iteratively pruned their CNNs. They used several ensemble strategies including max voting, stacking, averaging, and weighted averaging. They found weighted averaging to have the best performance of all of these strategies. Focusing their efforts on multiclass classification, the authors found that in this task the three best performing iteratively pruned networks were the VGG-16, VGG-19, and Inception-V3 network. This is surprising because the authors used a pruned deeper network (DenseNet-201) and the accuracy of this network was not as high. The criteria for using the three networks they chose came from the typical metrics derived from a confusion matrix. If the team chose their learning ensemble by paying additional attention to the saliency maps of these networks, it may have increased how generalizable their system is. The team only created saliency maps for analyzing the VGG-16, VGG-19 and Inception-V3 after their network was finally chosen. The saliency map of the Inception-V3 CNN had the best-looking performance as it did not focus on areas outside of the lungs. Watching the saliency maps change during iterative pruning may be another area of research the team could have invested time in. Overall this paper has some of the best results in all of the published COVID-19 X-ray literature. The author's overall pruned ensemble had an overall accuracy of 0.9901, an AUC of 0.9972, a sensitivity of 0.9901 and an F1-score of 0.9901.

These results seem promising although they made the mistake of using Kermany et al.'s [16] pneumonia dataset of children between the ages of one to five years old.

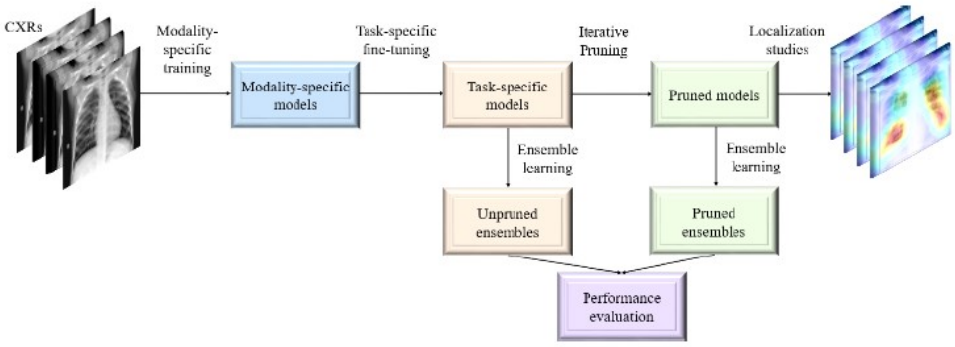


Figure 2.12: Rajaraman et al. [14] used this workflow for evaluating pruned CNN models to diagnose COVID-19.

Another study that deserves consideration is Wehbe et al.'s [19] paper that attempted to diagnose COVID-19 using a large private dataset in a US medical institution. This paper was similar to Rajaraman et al.'s paper [14] in that it constructed an ensemble of many CNNs to detect COVID-19. Their dataset, however, didn't suffer from the same deficiencies in size as other datasets. They also did not use Kermany et al.'s [16] dataset. The paper is noteworthy in that the authors assembled a team of five radiologists to determine the diagnosis of COVID-19 patients. They thereafter compared the predictions of the radiologists with their ensemble model. They found that the consensus of five radiologists was only able to detect COVID-19 with 81 percent accuracy. These results give a reasonable estimate of Bayes error for the task of determining the diagnosis of suspected COVID-19 patients. The author's ensemble model produced predictions with 82 percent accuracy, which is reasonable given the experts' consensus accuracy of 81 percent. Previous studies were unable to perform a comparison of their models against the predictions of working radiologists. The evaluation metrics mentioned in many of the previous papers were also liable to be skewed by the size of their datasets. Smaller datasets can sometimes lead to overly promising results.

Yeh et al. [20] used private datasets from several medical institutions and added on to Wang et al.'s dataset [51] when training their DenseNet-121 model [31]. They trained

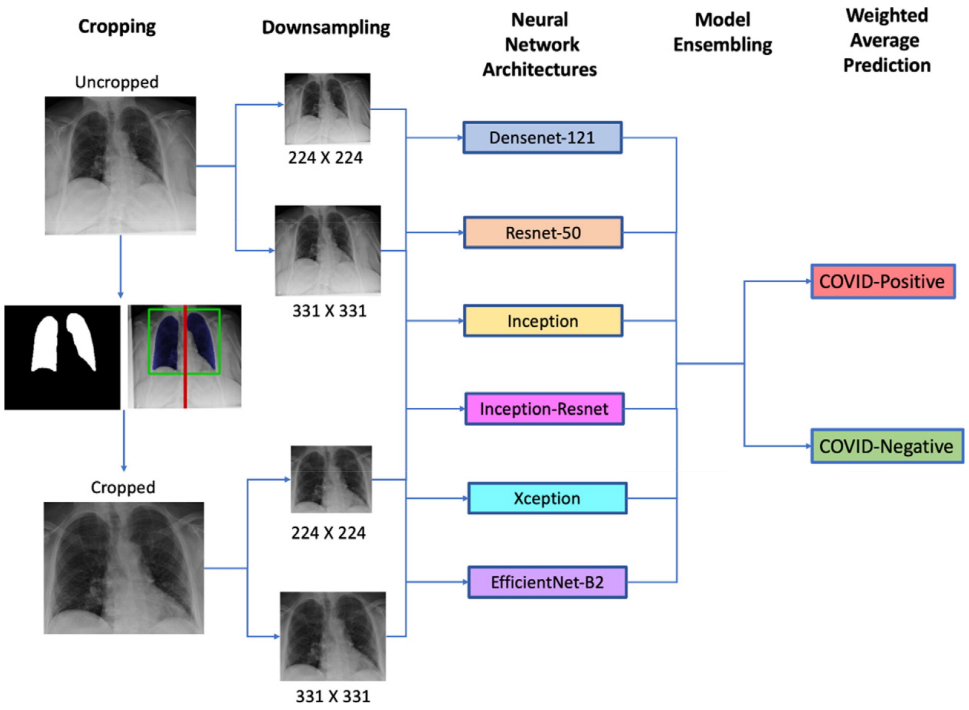


Figure 2.13: Wehbe et al.'s [14] ensemble model to diagnose COVID-19.

and tested their deep learning model initially using images from the same sources as Wang's COVIDx Dataset. They also used pneumonia, COVID-19, and normal X-ray images from two medical institutions. They obtained very promising results and achieved COVID-19 sensitivities between 95-100 percent. They held out a third much larger private dataset from a medical institution to see how their results would change with extra data. This larger dataset caused their accuracy to drop and they achieved an 81.82 percent COVID-19 sensitivity on their test set. This is evidence that using a small COVID-19 X-ray dataset leads to unrealistic evaluation metrics. The third private dataset only included 306 extra COVID-19 patients, but these added images caused a drastic change to the results of their deep learning model.

Horry et al. [54] developed a segmentation–classification deep learning pipeline for diagnosing COVID-19 that was trained and tested on a relatively small preprocessed dataset. While Horry et al.'s [54] final curated dataset was not biased, it contained only 100 COVID-19 images, so it is difficult to ultimately know how well their work would translate to a larger number of images. Horry et al. [54] additionally removed images from their dataset

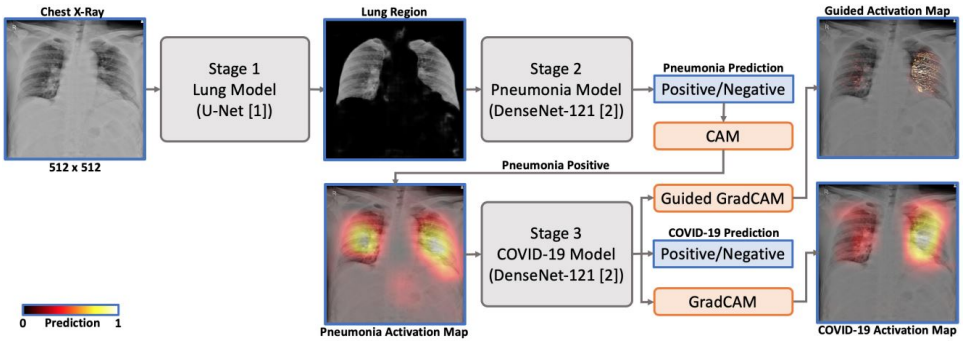


Figure 2.14: Yeh et al.'s [20] Densenet-121 models to diagnose COVID-19.

which contained features they believed their model would have difficulty classifying. The authors' segmentation model was not based on a deep learning model. They simply used OpenCV's GrabCut function and reasoned that "that the lung area could be considered the foreground of the X-Ray image" [54]. After preprocessing they trained five base models with their segmented images (VGG-16 [30], VGG-19 [30], Inception-V3 [34], Xception [55], and ResNet-50 [29]). Their best base model (VGG-19 [30]) ultimately achieved an F1-score of 81 percent.

Tabik et al. [56] created a dataset dubbed the "COVID-GR-1.0" dataset which was used in training their "COVID-SDNet" model in diagnosing COVID-19. Their dataset was divided in a novel fashion whereby COVID-19 positive patients were subdivided into four risk categories (normal-PCR+, mild, moderate, and severe). The authors created this dataset to see how many of weak COVID-19 cases would be analyzed by a prospective classifier correctly. More often than not, in COVID-19 datasets, there is an unequal number of severe COVID-19 patients. Typically, patients who end up undergoing a radiological examination end up being patients experiencing increased complications. COVID-GR-1.0 is a small but well-curated dataset that has utility in that it can be employed to determine a classifier's efficacy on weak COVID-19 images. Tabik et al.'s [56] pipeline consisted of a segmentation module and a classification module that performs "inference based on the fusion of CNN twins." [56] The authors used a U-Net [46] segmentation module and trained it on the Montgomery County X-ray dataset [57], the Shenzhen Hospital X-ray datasets [57] and the RSNA Pneumonia CXR challenge dataset [58]. They calculated the smallest rectangle

around each segmented image and added a border containing 2.5% of the pixels around each rectangle to obtain their final masked images. The X-rays they segmented were, therefore, never fully masked. The authors did not want to exclude relevant information in these images that could contain useful diagnostic information. After performing binary classification on their segmented COVID-GR-1.0 dataset, Tabik et al.'s [56] classifier obtained a COVID-19 sensitivity of 72.59%.

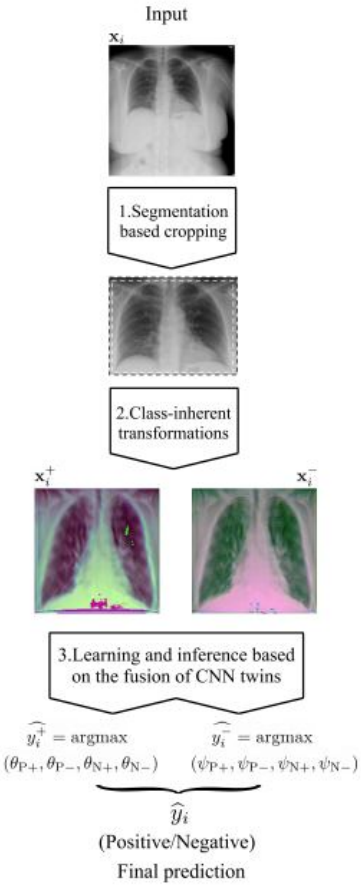


Figure 2.15: The 'segmentation – classification' system developed by Tabik et al. [56]

Teixeira et al. [59] designed a segmentation–classification pipeline used to diagnose COVID-19 that consisted of a U-Net [46] and InceptionV3 [34] CNN. Their U-Net [46] segmentation module was trained on images and masks that were hand-picked from a mixture of public datasets ([57], [60], [61]). The number of images and mask pairings they chose

in the Darwin V7 labs [60] segmentation dataset (489) was significantly lower than the total number of pairings available in that dataset (6504). This approach looks as though it allowed them to train their U-Net [46] to have a higher dice similarity coefficient (0.982) than other segmentation units we have seen in the literature for this task. For classification they otherwise used the RYDLS-20 dataset [62]. They had developed this dataset in a previous work and further added images to it to create a new "RYDLS-20-v2" dataset. They attempted to use several classifiers but ultimately found that using an InceptionV3 [34] CNN resulted in giving them their best overall multiclass performance metrics.

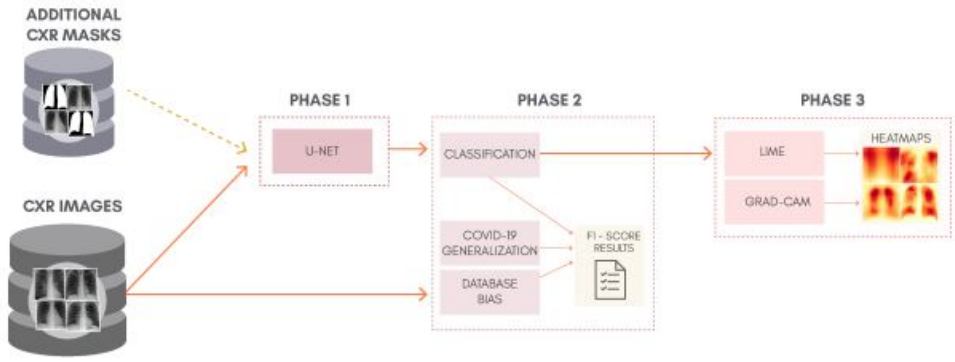


Figure 2.16: The 'segmentation – classification' system developed by Teixeira et al. [59]

Abdulah et al. [63] implemented a segmentation – classification pipeline that used a unique segmentation unit and ensemble model for classification. Their segmentation unit, the Res-CR-Net, is a new kind of segmentation model the authors introduced in a previous study [64] that does not contain the same encoder-decoder structure that the popular U-Net [46] contains. According to the authors, the Res-CR-Net "combines residual blocks based on separable, atrous convolutions [65, 66] with residual blocks based on recurrent NNs [67]." [64] The authors trained their Res-CR-Net [64] on several open-source sets of masks and images [57, 60, 61]. They acquired their classification dataset from the Henry Ford Health System (HFHS) hospital in Detroit. This private dataset contained 1417 COVID-negative patients and 848 COVID-positive patients. The authors used this dataset to train a unique hybrid convnet called the "CXR-Net" that contains a Wavelet Scattering Transform (WST) block

[68, 69], an attention block containing two MultiHeadAttention layers [70, 71], and several convolutional residual blocks. This segmentation-classification pipeline ultimately achieved an accuracy of 79.3% and an F1 Score of 72.3% on their test set.

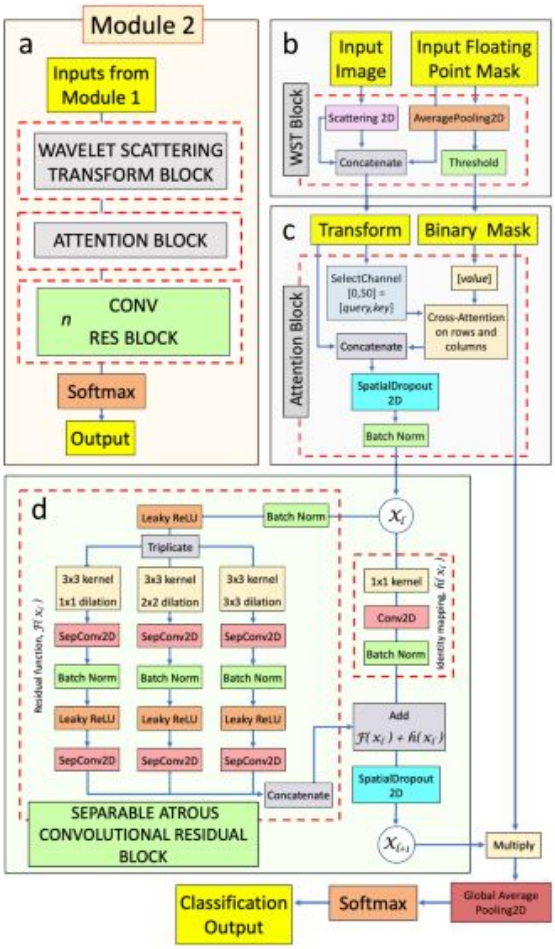


Figure 2.17: The 'segmentation – classification' system developed by Abdulah et al. [63]

2.5.2 2D CT Studies

There are only a few true 2D CT COVID-19 studies that have been reviewed here. If the only criteria for diagnosing a person with COVID-19 in a study involves checking whether a single slice in a volume contains COVID-19, that study ends up in this 2D CT section. If there are machine learning techniques that analyze a set of slices after passing

through a 2D classifier in a study, that study ends up in the 2.5D CT section of this report. The studies reviewed in this section below attempt to diagnose COVID-19 based on using single slices.

Amyar et al. [72] created a multi-task system that classifies patients to be COVID-19 positive vs. COVID-19 negative. The team built a 2D U-Net encoder-decoder network for processing the 2D CT slices of suspected COVID-19 patients. The latent space in the encoder-decoder network was utilized for three separate parallel tasks. The first task was to reconstruct the original images of the CT slices input into the system. The second and third tasks were for creating a robust segmentation network and classifier. For the third task, the output of the encoder had a convolutional layer added to it, followed by a max-pooling and flattening layer. The flattened layer had an MLP with three dense layers added to it, where the last layer was a single neuron with a sigmoid output (COVID vs non-COVID). The authors of this paper created this model to leverage information in the three related tasks to improve the segmentation and classification models. As the tasks were all trained in parallel, they felt this could eventually lead to the creation of better systems for each task. They combined three datasets to obtain a total dataset of 449 COVID-19 patients and 595 non-COVID-19 patients. The non-COVID-19 patients were either normal patients or had some other lung pathology. This study suffers from a few deficiencies. No mention of data augmentation is found in the study. Interpretable saliency maps have not been used, making it difficult to know if their system results can be trusted with such a small amount of data. The method the used for training three parallel tasks however is unique in the COVID-19 literature. For the segmentation task, they report achieving a dice coefficient of 78.52 percent. The classifier they have built results in an accuracy of 0.86, a sensitivity of 0.94, a specificity of 0.79 and an AUC 0f 0.93.

Polsinelli et al. [73] proposed a model for COVID-19 detection based on the SqueezeNet architecture [32]. Polsinelli et al. [73] was an Italian research team looking to make a fast contribution to the task of detecting COVID-19 when Italy was being the hardest hit in April 2020. The authors attempted to make an original research contribution of their own by changing SqueezeNet's architecture to help make it more lightweight while retaining or

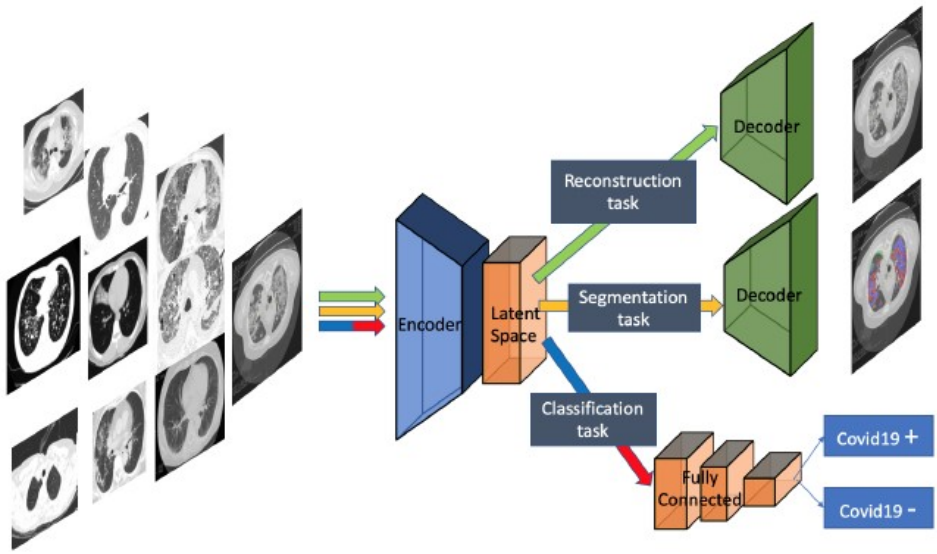


Figure 2.18: Amyar et al.'s [72] multi-task model for training segmentation and classification tasks simultaneously.

improving its accuracy. The SqueezeNet architecture uses "Fire Modules" and the authors of this paper suggest making changes to the classic modules used in the original architecture. They have replaced all ReLU layers with exponential linear unit (ELU) [74] layers because there is literature showing that ReLU networks with batch normalization can be outperformed by ELU networks without batch normalization [74]. The authors encourage their readers to keep in mind that each batch normalization added increases computational overhead by approximately 30 percent. Their architecture removes the original skip connection of the SqueezeNet architecture and adds a transpose convolutional layer to the last custom-designed fire module in the network. The dataset they used consisted of 460 COVID-19 images and a batch of 397 CT scans from patients who either had other lung illnesses or were healthy. They decided to use a Bayesian optimization [75] approach for the tuning of their learning rate, momentum and L2-regularization hyperparameters. The achieved results were 83 percent accuracy, 85 percent specificity, 81 percent specificity, and an F1-score of 0.8333. The authors created a metric to measure their performance in terms of efficiency (sensitivity/number of parameters) to compare their work with other papers. This is a sensible step since the team was attempting to show the lighter SqueeeNet architecture could be used in place of the very deep networks used in other COVID-19 papers. The system is

trained on a high-end workstation (Intel Xeon Processor E5-1620, CPU RAM 16GB, GPU Nvidia Quadro M4000 8GB) but ultimately tested on an i5 laptop (8 GB Ram without a dedicated GPU). The aforementioned evaluation metric for determining the model's efficiency is important here in showing that the system can be implemented on a device with less computational capacity. The performance of the model in this paper was evaluated using Grad-CAMs. While the system has noted areas of infection correctly it also has mistakenly found other areas not relevant to a positive COVID-19 diagnosis.

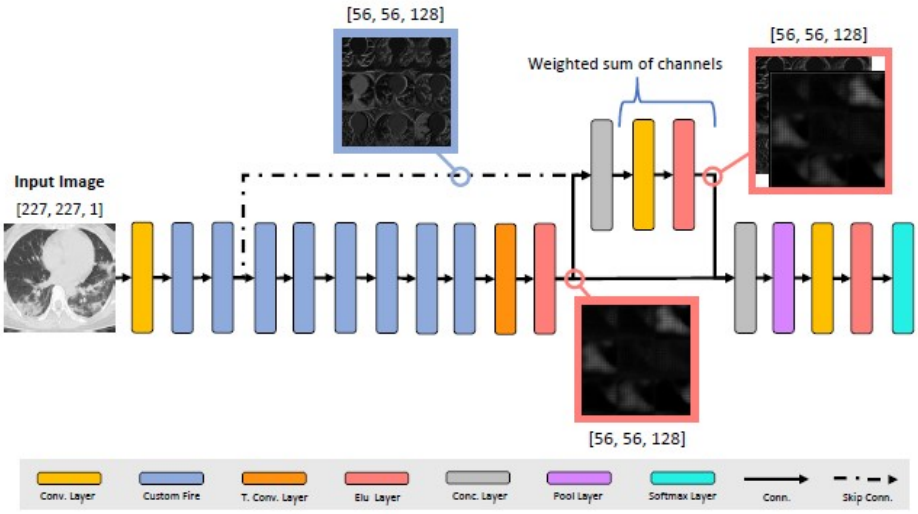


Figure 2.19: Polsinelli et al.'s [73] SqueezeNet model CT model to diagnose COVID-19.

Ko et al. [76] developed the fast-track COVID-19 classification network (FCONet) to perform diagnose COVID-19. Their dataset has 264 images from 264 COVID-19 patients. It also has 1357 CT scans from 100 CAP patients and 1442 CT scans from 126 normal patients. Their image sizes were 256x256 and they used image rotation and zooming as data augmentation methods. They used five-fold cross-validation and used a holdout test set. They compared four very popular pre-trained 2D models for classification. Ultimately out of the four models they tested, they found the ResNet50 was the best, having a sensitivity of 99.58 percent, a specificity of 100 percent and an accuracy of 99.75 percent. To improve the interpretability of the model the authors used Grad-CAMs. The heatmaps they generated for COVID-19 lung images strongly indicated the suspected infection regions. The framework is

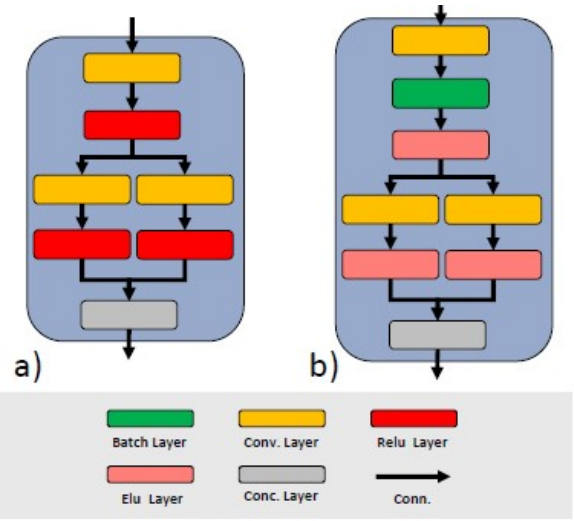


Figure 2.20: (a) Original fire module, (b) Polsinelli et al.'s [73] custom fire module.

useful in that it is more easily generalizable to other CT sets because it works on a slice-by-slice basis. CT scanners depending on their settings can output different numbers of scans. The thickness of these scans often varies as well. One major limitation is the authors did not make use of lung segmentation. Another limitation is that the "testing data set was obtained from the same source as the training data set" [76] so it is difficult to tell how well the system generalizes.

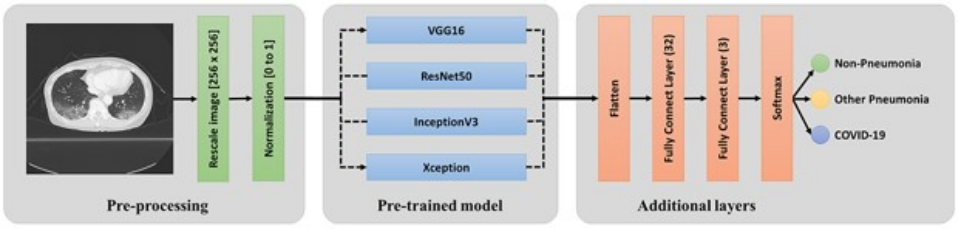


Figure 2.21: Ko et al.'s [76] various 2D CT models that were attempted.

2.5.3 2D X-ray and CT Studies

There are very few machine learning models that deal with both X-ray and CT modalities in the COVID-19 imaging literature. This means there is a lot of room in this space

for exploring models that leverage both imaging modalities. Below are the only two models that can be found in online databases at present.

Maghdid et al.[77] is the first available paper published that diagnoses COVID-19 using a combination of X-ray images and single CT scan slices. The authors of this paper designed a very simple CNN that is built using a single CNN layer, two fully connected layers and a softmax classification layer. For comparison, the authors used an ImageNet pretrained 2D version of AlexNet. The system did not use a segmentation unit although during the preprocessing stage the chest and lung areas were cropped to remove unnecessary information surrounding regions of interest. The authors ensured that their images were gathered from various facilities and devices so that the system could generalize to new examples. Their dataset had 170 X-rays and 361 CT images. The model on X-ray images had an accuracy of 94 percent, a sensitivity of 100 percent and a specificity of 88 percent. The performance changed when inputting CT images into their CNN. CT images with their model produced a 94.1 percent accuracy, 90 percent sensitivity, and 100 percent specificity. Although the results seem acceptable, no posthoc analysis was performed using Grad-CAMs to ensure the classifier was operating correctly on such a small dataset. The model is included in this review only because it was the first model that worked on the principle of using a single classifier using both major radiological modalities simultaneously.

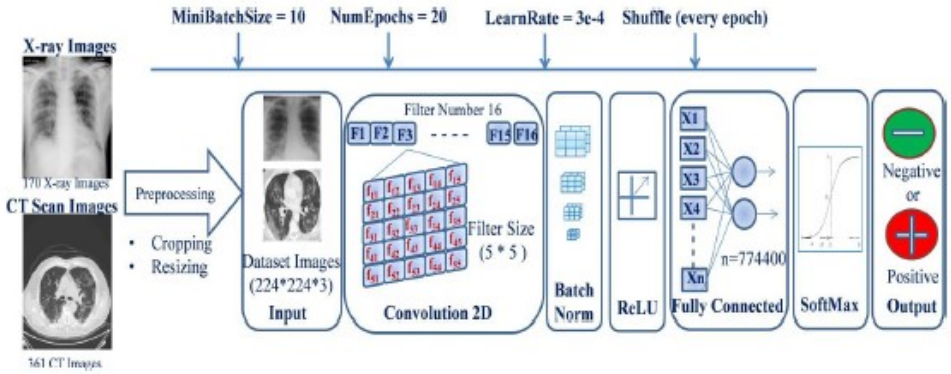


Figure 2.22: Maghdid et al. [77] built this single model to work on either single X-rays or CT slices.

In another related study, Alom et al. [78] used a 2D system to work on both X-ray and CT images. Unlike Maghdid et al. [77], Alom et al. [78] did not use the same single system to train both X-rays and CT scans simultaneously. To be clear, the authors performed the training and testing on each modality separately (two separate systems that were the same). The system has been named COVID-MTNet and is mentioned because the authors felt that if they were working on a 2D system for X-rays they should additionally use the same system for CT slices as well. The CT dataset used in this study has 420 samples collected from normal patients and 178 samples from COVID-19 patients. They used an Inception Recurrent Residual Neural Network (IRRCNN) [79] for the task of detecting COVID-19. The system they have designed additionally uses a segmentation module they have dubbed the "NABLA-N" model. Their CT lung segmentation module is trained on 267 samples with corresponding masks and labels for a task not COVID-19 related. They train their X-ray segmentation module similarly. Their CT and X-Ray systems use the exact same architectures and they are both initially trained for another task (normal vs. pneumonia) for which more X-ray images and CT scans are available. Once they have two good working systems for the normal vs. pneumonia task, transfer learning is used, and the IRRCNN-model gets trained on the COVID-19 classification task. The authors have created Grad-CAM heatmaps to show the performances of their X-ray and CT algorithms. This gives more confidence in their system's reported performance metrics. Their system for CT slices achieves a testing accuracy of 98.78 percent and their system for X-ray images achieves a testing accuracy of 84.67 percent. The authors believed the X-ray classifier struggled more because it needs to be trained on more samples. The model on X-ray images had an accuracy of 94 percent, a sensitivity of 100 percent and a specificity of 88 percent. The performance changed when inputting CT images into their CNN. CT images with their model produced a 94.1 percent accuracy, 90 percent sensitivity, and 100 percent specificity.

2.5.4 3D CT Studies

3D CT studies tend to involve a greater amount of hardware than is available to many academic research teams. Some teams at good universities or medical institutions have

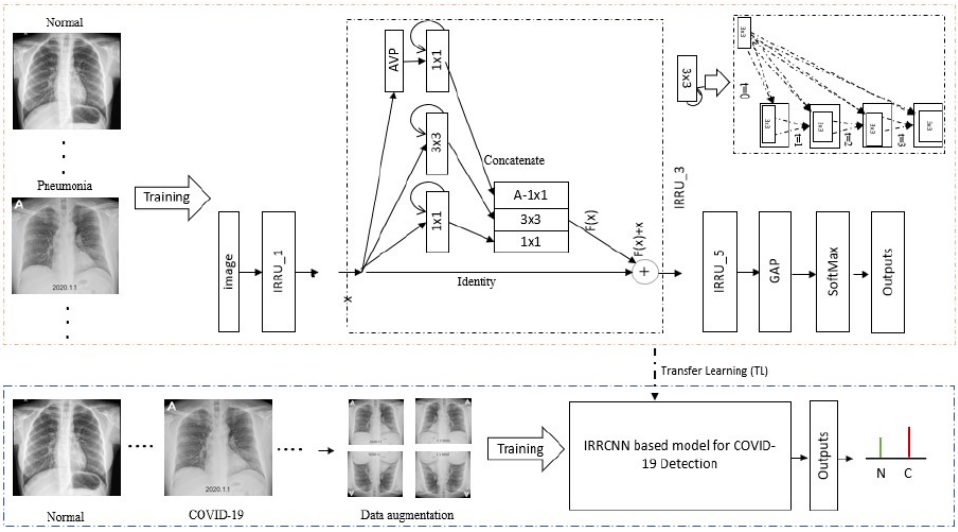


Figure 2.23: Alom et al. [78] built twin versions of this model for working on X-rays and CT slices.

the necessary hardware (GPU clusters with lots of VRAM) to perform 3D segmentation and classification on a COVID-19 patient's lungs. Sometimes a 2D segmentation unit is used by a 3D classifier in a manner that concatenates segmented CT slices into whole CT volumes before classifying them. Often the segmentation techniques in the literature require a substantial amount of annotated data. Annotated data can be difficult to obtain for researchers who are not working within or alongside a medical institution. Within medical institutions, investing time in manually annotating CT slices for building deep learning models is expensive. The models below often use annotated data not commonly available. They also tend to perform better than their 2D counterparts but may suffer from lower interpretability in some cases.

Shan et al. [80] published a paper where they set out to develop a deep-learning-based system for the "automatic segmentation and quantification of infection regions" within COVID-19 CT scans. Their system uses a 3D VB-Net segmentation network [81] that is trained on 249 COVID-19 patients and tested on 300 different COVID-19 patients. Their fast auto-contouring tool is constructed using a human-in-the-loop strategy. The training data is segmented into multiple batches. The smallest batch is manually contoured by trained radiologists first. After the first batch is used to train the network its output is

manually corrected by radiologists. This initial batch gets added to the next batch and this iterative process continues. The authors have discovered that the human-in-the-loop method converges after 3 to 4 iterations. The system they constructed results in a dice coefficient of 91.6 percent. The time it takes for radiologists to segment out COVID-19 CT images ranges between 1 to 4 hours, whereas it only takes their final system 4 minutes to perform the same task. After segmentation, the authors of this study developed quantitative metrics for measuring the volumes of infection in each region of interest and calculating the percentage of infection. The system they have developed will only work on volumes generated with their CT machine and their particular set of CT imaging protocols. The system is not generalizable for all volumes generated at different medical institutions.

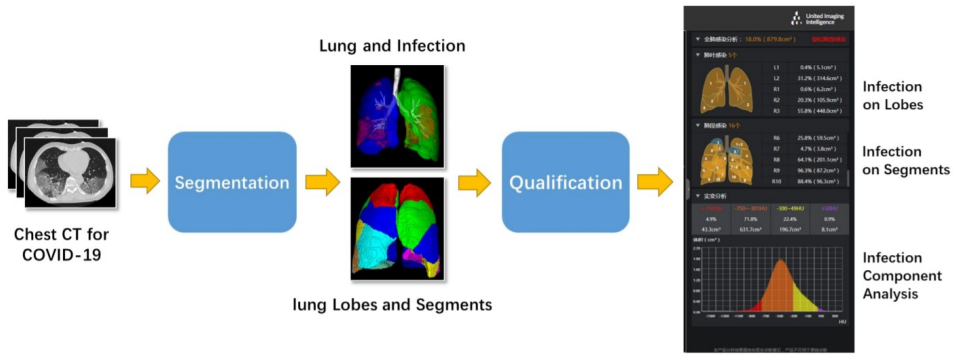


Figure 2.24: Shan et al. [80] built a segmentation model for segmenting regions of interest in quantifying COVID-19 infections.

Shi et al. [82] developed a 3D VB-Net segmentation module that is used prior to a series of random forest classifiers that perform COVID-19 screenings using location-aware feature extraction. The dataset for this study contains 1658 COVID-19 patients and 1027 CAP patients. Four handcrafted features have been created to analyze the data: the volume of lesions, the infected lesion number, a histogram of pixel intensity values for infected regions, and "the distance of each infection surface vertex to the nearest lung boundary surface."[82] The first random forest classifier in the model separates infection regions based on their measurements into four groups. Several random forest classifiers are then designed to operate on those four groups. During training, the authors employed the LASSO [83]

method to discover the most effective features that will provide clarity for a diagnosis. The selected features were input into logistic regression classifiers, support vector machines and neural network models for comparison with the proposed SARF (Size Aware Random Forest) model in this paper. The final trained SARF model used five-fold cross-validation and the method obtained an AUC of 0.942 on the test set. The method additionally obtained a sensitivity of 0.907, a specificity of 0.833 and an accuracy of 0.879. The authors found it difficult to screen for COVID-19 patients in the early stages of its progression, but as the lesion sizes grew the patients could be reasonably diagnosed using the SARF method.

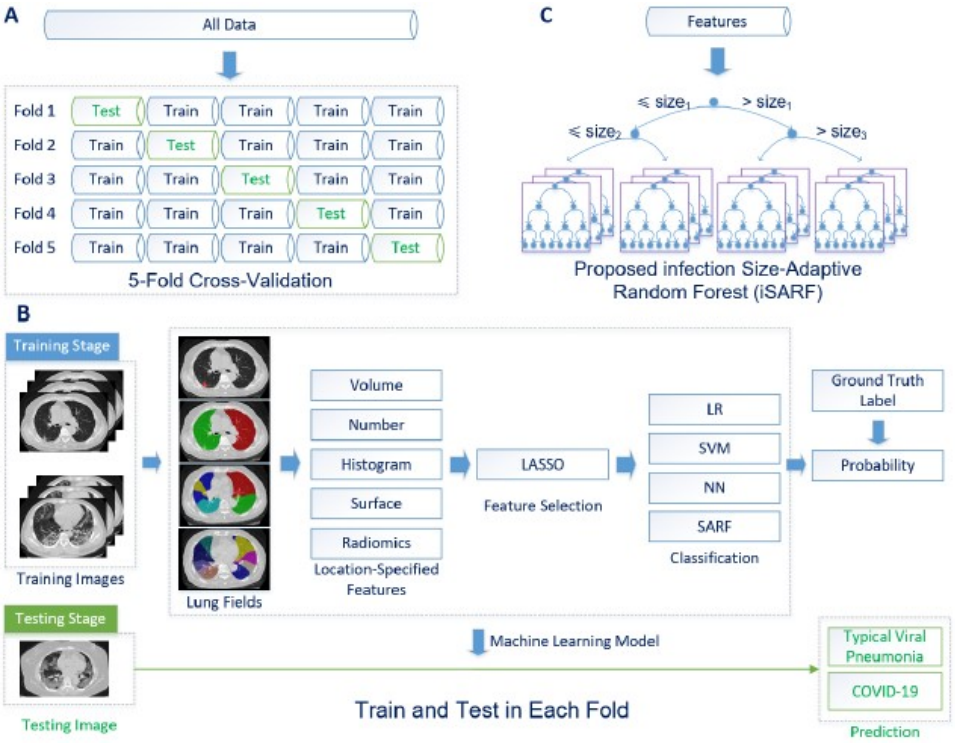


Figure 2.25: Shi et al. [82] built their SARF model to predict a COVID-19 diagnosis using 4 handcrafted features.

Tang et al. [84] designed a system for measuring the severity of a COVID-19 patient's illness that uses random forest classifiers. This study used a professional tool for lung segmentation developed by Shanghai United Imaging Intelligence Co. Ltd. that is based on a VB-Net architecture. The 3D segmentation module is important as it was used to

extract the quantitative features from the whole lung, right/left lung, 5 lung lobes, and 18 lung segments. The infection volume and ratio of all these segments are calculated. The volumes and ratios of these areas are used to create a total of 63 final features. These features are extracted from a dataset containing the chest CT scans of 176 patients. The authors created separate random forest models using 63, 50, 40, 30, 20, and 10 features. Their model with ten features obtained the best performance and so it was chosen as their final severity assessment model. This model obtained an accuracy of 0.875 and an AUC of 0.91. A limitation of this study is that the authors only designed the system to perform binary classification (non-severe and severe) rather than design for more nuanced categories (mild, common, severe, and critical for example). There are not many COVID-19 deep learning severity assessment models in the literature, so this is an important paper in that regard. One interesting finding of the authors was that the right lung lobes were more relevant to the severity of COVID-19 than the left lung lobes. Another important finding was that the infection volume of the entire lung is highly correlated to the severity of a patient's illness. This kind of imaging severity assessment requires good quality lung segmentation and a large number of annotations from experienced radiologists.

Wang et al. [18] designed COVID-19Net for the diagnostic and prognostic analysis of COVID-19 patients. The system starts with a segmentation module that has a DenseNet121-FPN as a backbone and is trained on the VESSEL12 dataset [85]. During segmentation module training, the authors fine-tuned their ImageNet pretrained DenseNet121-FPN backbone using 3 adjacent CT slices from the VESSEL12 dataset. During testing, however, "lung segmentation was performed slice-by-slice." [18] The segmentation process is performed on a two-dimensional basis whereby the segmented slices form concatenated volumes that are fed into the team's classifier. In addition to lung segmentation, the team preprocessed the data to suppress non-lung areas that still existed after segmentation by reducing those areas' pixel intensities. This is due to their finding that deep learning systems tend to focus on areas with high intensity. COVID-19Net is a custom-built DenseNet-like classifier with a series of carefully chosen convolution layers, dense blocks, and pooling layers. At the last convolutional layer, it uses "global average pooling to generate 64-dimensional deep learning features." [18] A sigmoid function at the tail end of the classifier results in a positive or

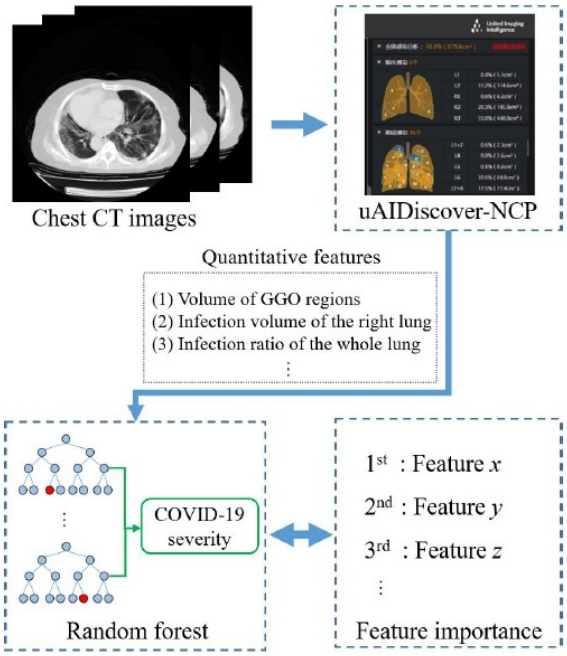


Figure 2.26: Tang et al. [84] built a segmentation model for segmenting regions of interest in quantifying COVID-19 infections.

negative diagnosis for COVID-19. This model is initially designed for the task of making predictions on the CT-EGFR dataset to predict lung cancer in people with EGFR gene mutations. This modality-specific pretraining is an important part of designing the 3D classifier. The dataset for this step has 4106 patients and is much larger than current COVID-19 CT datasets. Following this auxiliary training step, COVID-19Net is trained on a COVID-19 dataset. The COVID-19 dataset contains 1266 patients (471 with CT follow-ups, 924 COVID-19 patients, and 342 with other pneumonia). COVID-19Net's final 64-dimensional feature vector was later combined with metadata (age, sex, and comorbidity) to predict a patient's prognosis using a multivariate Cox proportional hazard model [86]. The authors in this study have used Grad-CAMs and visualization tools to ensure their system is localizing COVID-19 features correctly. In diagnostic classification, the model on their 2nd validation set achieves an AUC of 0.88, an accuracy of 80.12 percent, a sensitivity of 79.35 percent, a specificity of 81.16 percent and an F1-score of 82.02 percent. Their prognostic Kaplan-Meier analysis [87] shown in Fig. 2.28, succeeded in demonstrating that "patients in high- and low-risk groups had a significant difference in hospital stay time." [18]

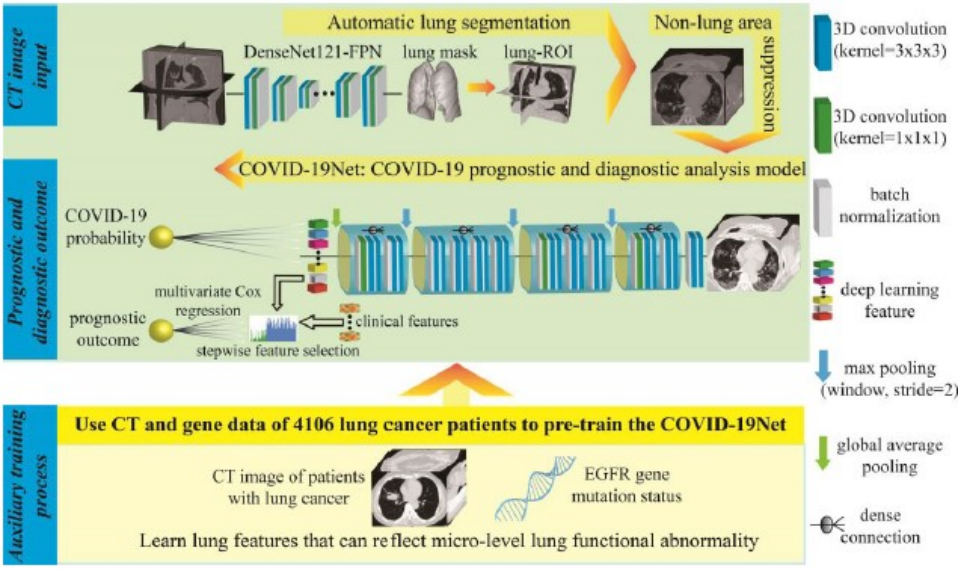


Figure 2.27: The COVID-Net19 system model built by Wang et al. [18]

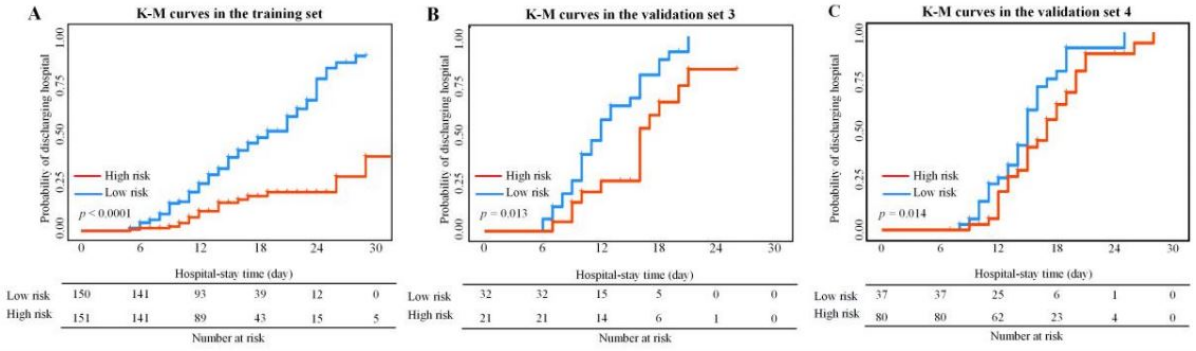


Figure 2.28: Prognostic Kaplan-Meier analysis for high and low risk patients by Wang et al. [18]

Wang et al. [88] have published a study on a "weakly supervised deep learning framework" that uses 3D volumes for classification/diagnosis. The purpose of this study is to automatically annotate lesions to speed up the work of radiologists. Their classifier is a lightweight 3D CNN model they have named 'DeCoVNet'. Their dataset consists of 540 patients, 313 of which have been diagnosed with COVID-19. The dataset also has 229 normal patients. All these patients initially came into the hospital with possible symptoms for COVID-19 (fever, cough, fatigue and diarrhea). The slices of each 3D volume are slice-by-slice sent through a U-Net segmentation module. The authors trained a 2D U-Net while

using masks that were generated using the unsupervised 3D connected component method [89]. The authors here hoped to require less large-scale annotation than would otherwise be required using supervised segmentation. The team made use of trained professionals who were able to manually remove poorly annotated segmented images output by the unsupervised U-Net. All of the slices were processed for segmentation and thereafter concatenated to form a 3D CT volume of each patient. These volumes were fed into the 3D DeCovNet classifier. The main classifier they used consisted of a 5x7x7 3D stem, two 3D ResNet blocks, an adaptive max-pooling block and several fully connected blocks leading to two outputs. They used localization algorithms to find the 2D and 3D locations where DecoVNet was identifying infection regions. The author's algorithm results in a ROC AUC of 0.959, a PR AUC of 0.976, an accuracy of .901, a ppv of 0.840 and an npv of 0.982. One drawback of this approach is that all the images sent through the system are from the same CT scanner. This means the system may not generalize well to other CT scans from different facilities. The drawback of using a 2D segmentation approach is that segmentation is often performed by radiologists in both a 2D and 3D fashion. If a radiologist thinks a slice should be annotated a certain way, he/she will look at several other nearby slices to inform his/her annotations. The 2D U-Net will miss some 3D annotated information but takes considerably fewer resources to train.

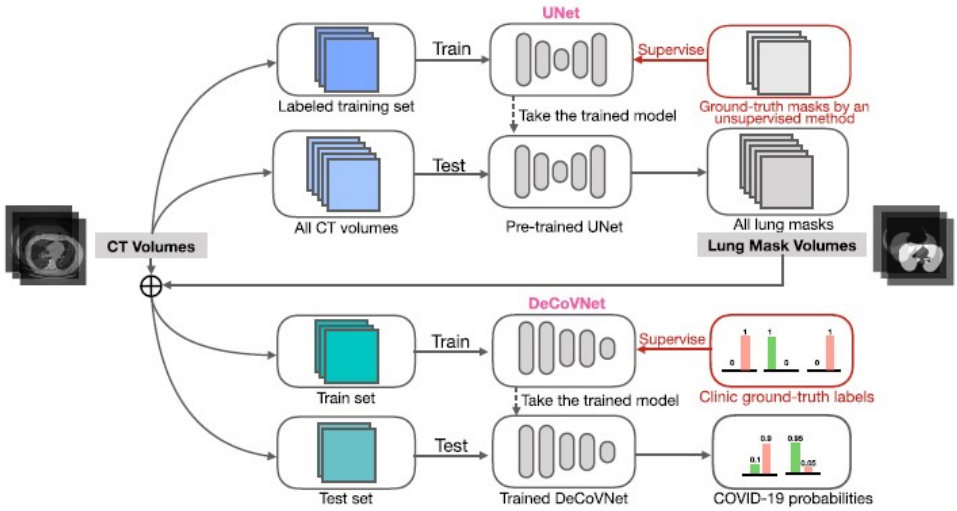


Figure 2.29: The 2D U-Net and 3D DeCovNet classifier system built by Wang et al. [88]

Jin et al. [90] built a system to detect COVID-19 and distinguish it from other pulmonary diseases. Their model has deployed in 16 hospitals in China and "is performing over 1300 screenings/day." [90] The authors in this study used 1136 cases for training (723 were positive) and 282 for testing (154 were positive). Their data-gathering stage was well planned and they obtained CT scans from five Chinese hospitals. Those hospitals used eleven different CT scanner models. This data-gathering stage ensured the system could generalize across a host of CT scanners. The authors designed a "segmentation – classification" system. Particular regions of interest were extracted using a 3D U-Net++ [91] and those regions were thereafter forwarded to a 3D classifier for diagnosis. The system segments out the lungs first before segmenting out regions of interest. The supervised segmentation units built by the authors of this study required the large-scale annotation of CT scans to design a robust system. The authors used lesions that were annotated from COVID-19 images as well as lesions taken from subjects suffering from other pulmonary diseases. They had a large team of trained radiology experts performing these annotations. After segmenting out lesions, a 3D ResNet-50 classifier was used by the authors to classify whether the lesions in the patches presented to the classifier were caused by COVID-19. The authors tried various segmentation and classification modules and determined that a "U-Net++ - ResNet-50" combined model obtained the best results. Their model achieved an AUC of 0.991, a sensitivity of 0.974 and a specificity of 0.922. The system had extensive hardware requirements. It required "an Intel Xeon E5-2680 CPU, an Intel I210 NIC, two TITAN X GPUs and 64 GB RAM" [90] at deployment. The system additionally required a server with 8 TITAN X GPUs during training.

2.5.5 2.5D CT Studies

There are more 2.5D CT studies in the COVID-19 deep learning literature than there are 2D or 3D CT studies. The individual slices that are processed in 2D and 2.5D CT systems can be input into models that were pretrained on millions of images, whereas an ImageNet for 3D volumes currently does not exist. 2D CT systems are rare in the literature in comparison to 2.5D systems because most researchers feel that it is important to leverage the

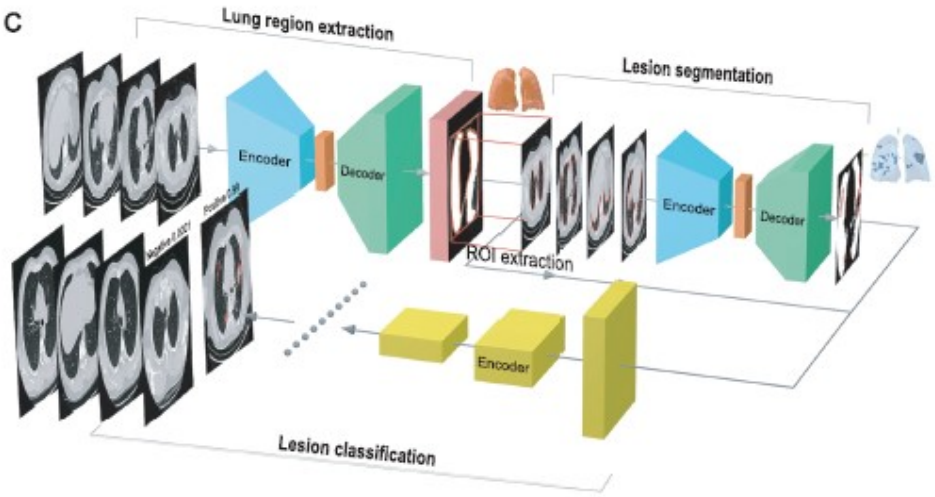


Figure 2.30: The 3D 'segmentation – classification' system developed by Jin et al. [90]

3D information in a CT volume. 2.5D CT systems, therefore, offer a compromise where they are easier to train/implement than 3D CT systems, but can also leverage 3D information. This compromise is often motivated additionally by hardware/cost considerations. Many 3D CT systems require a server with many CPUs/GPUs. 2.5D systems however may only require an expensive workstation with one or two high-quality GPUs. The lower cost and option of leveraging transfer learning in a 2D system leads many researchers to prefer some of the approaches provided below.

A study published in Nature Medicine by Mei et al. [92] used two CNNs and an MLP for diagnosing COVID-19. Their system used 2D images and 12 clinical features to classify patients as COVID-19 positive or COVID-19 negative. Their dataset of 905 patients consisted of 419 COVID-19 patients and 486 COVID-19 normal patients. The authors chose to separate this data into a 60 percent training set / 10 percent tuning set / 30 percent test set division. The image preprocessing of the system initially uses a standard lung window to normalize the pixel intensities into 256 bits. There is a form of segmentation in the system that uses some thresholds in the intensities of the pixels throughout the image to allow the system to focus only on the lungs. A slice selection CNN (Inception-ResNet-v2) has been pretrained on a related task and is used to identify abnormal CT images [93]. The 10 most abnormal slices get sent on to another CNN that performs an image-level diagnosis of a

patient. The diagnosis CNN's output has a global averaging layer that is used to create a 512-dimensional feature vector. This vector is combined with a patient's clinical features in a three-layer MLP. The MLP and CNN are trained together. The CNNs weights were initialized using a weakly supervised method whereby the CNN is initially trained to classify small image patches chosen from randomly selected training set images. The system was trained afterward on normal and augmented data for "40 epochs with a batch size of 16 samples." [92] The final system achieves an AUC of 0.92 and a sensitivity of 84.3 percent. This AUC and sensitivity outperformed a senior radiologist the hospital had on staff. The model has some limitations. Despite its promising initial results, it is not generalizable to other patient populations. Choosing slices out of an entire volume helps reduce the computational complexity of working with 3D CT scans, but can result in missing some key information from the other slices. One last weakness in this approach is that it has a bias towards COVID-19 patients and has difficulty detecting other forms of respiratory illnesses.

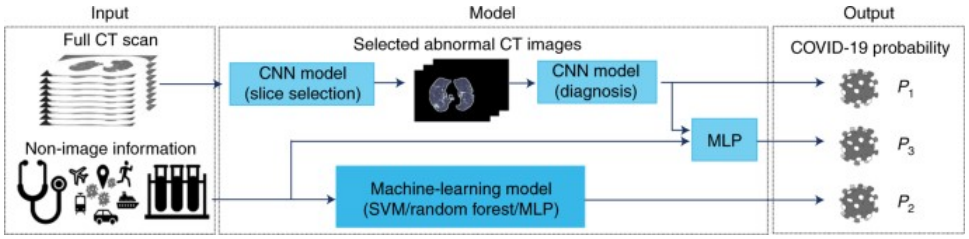


Figure 2.31: Mei et al.'s [92] model for combining the 10 highest 2D slice probabilities with metadata for COVID-19 diagnosis.

Song et al. [94] have developed a model called "DRE-Net" for COVID-19 diagnosis. It was developed right at the beginning of the pandemic (February) in China. It uses a pretrained ResNet-50 and a feature pyramid network is added to it to extract the top details of CT images. They have coupled an attention module to their network to focus their system on the most important details of the network. The authors have a dataset of 88 COVID-19 patients (777 CT images) and 101 patients infected with bacterial pneumonia (505 CT slices) and 86 normal patients (708 CT slices). The 3D volumes of the patients have been preprocessed into 15 slices and the slices with incomplete lungs were removed.

Rahimzadeh et al. [95] who copied Song et al. [94] in terms of using a ResNet feature pyramid did the same thing. Although Rahimzadeh et al.'s [95] paper will not be reviewed here, in Fig. 2.33 they published an instructive figure showing an example of incomplete lung that requires preprocessing. Song et al.'s [94] dataset has been randomly split into 60/30/10 percent training, validation, and test sets respectively. After the images pass through their DRE-Net model the slices are aggregated and pooling is used to calculate a result for each person. For two pneumonia patients, the team validated their model results using visualization techniques on the patient's top 3 predicted slices. They did so to see if the model was making the correct predictions in locating GGO abnormalities. The model succeeds in detecting areas with GGOs in both patients. It achieves an AUC of 0.97 at the image level and 0.99 at the patient level. At the patient-level, the system additionally achieves a recall of 0.93, precision of 0.96, F1-score of 0.94 and accuracy of 0.94.

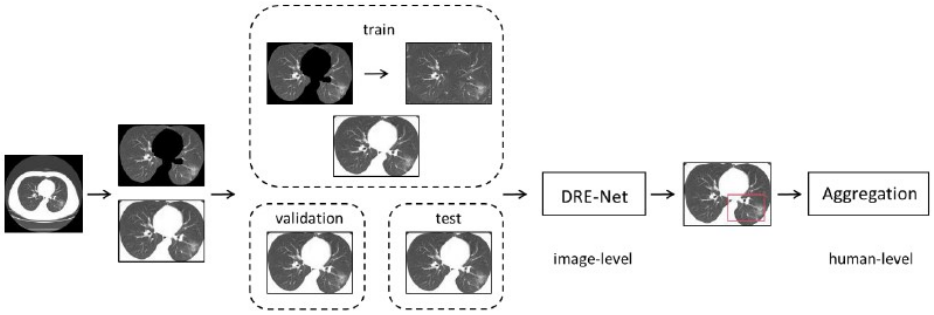


Figure 2.32: Song et al.'s [94] system for diagnosing COVID-19.

Bai et al. [96] have created a system that inputs all the slices in a CT scan through a series of parallel 2D classifiers. They use parallel EfficientNet-B4 CNNs on each slice (each slice is stacked to 3 channels of the CNN) of a patient and each network outputs a prediction. The predictions of the network are then pooled and a two-layer neural network is thereafter used to output a prediction as to whether a patient suffers from COVID-19 pneumonia or has some other form of pneumonia. The authors chose to use EfficientNets because this type of architecture possesses fewer parameters and still achieves good performance in comparison with other kinds of CNNs. Each slice of the network was segmented based on attenuation

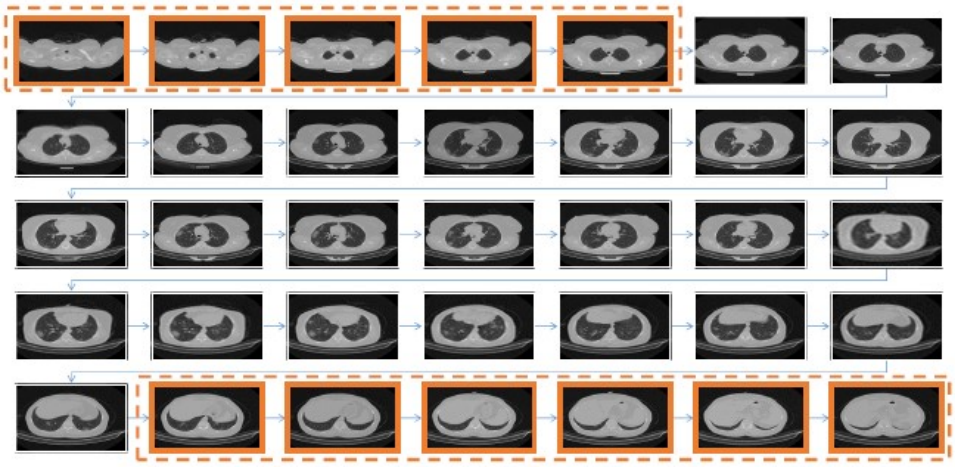


Figure 2.33: Rahimzadeh et al.'s [95] lung preprocessing step in their 2D CT CNN models cutting out uninformative slices.

(-320 HU used as a threshold value) to exclude non-pulmonary regions of the CT before being input through the classifier. The authors used a lung window with a window width of 1500HU and a window level of -400HU. This lung window was applied during preprocessing to generate 8-bit images that eventually were normalized from a 0-255 value to a 0-1 value. There was an additional step mentioned where the authors needed to normalize to the ImageNet mean and standard deviation values. The authors used data augmentation methods dynamically during training and this "included flips, scaling, rotations, random brightness and contrast manipulations, random noise, and blurring." [96] Heatmaps were additionally generated to ensure the classifiers were finding locations with COVID-19 correctly in the lungs. The model achieved a test accuracy of 96 percent, a sensitivity of 95 percent, a specificity of 96 percent and an AUC of 0.95. The model was later shown to help actual radiologists improve their diagnoses.

Lin et al. [97] have created a system that has both 2D and 3D features. Their dataset consists of 4356 volumetric CT scans from 3322 patients. 1296 patients are confirmed to have COVID-19, 1735 patients have CAP and 1325 patients have no form of pneumonia whatsoever. Their COVID-19 positive patients all have had confirmed RT-PCR tests. They first preprocess the images and extract the lungs using a U-net segmentation method. The images then are input through the team's 'COVNet' classifier for generating predictions.

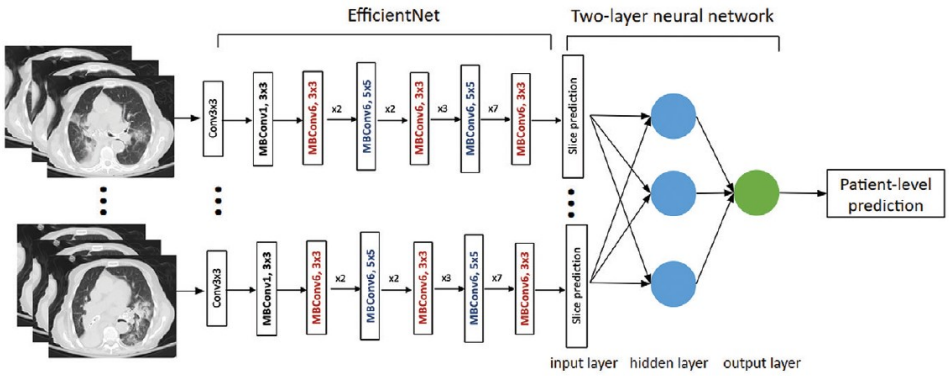


Figure 2.34: Bai et al.'s [96] model of using parallel EfficientNet CNNs followed by a two layer neural net.

COVNet is designed with several parallel 2D ResNet-50s whose outputs are all fed into a max-pooling layer. Following the max-pooling layer, "the final feature map is fed to a fully connected layer and softmax activation function." [97] The purpose of this processing step is to extract features from local 2D slices while also extracting global 3D features. Their system detects COVID-19 with a sensitivity of 90 percent and specificity of 96 percent. The AUC for COVID-19 detection is 0.96. For detecting CAP their system achieves a sensitivity of 87 percent, a specificity of 92 percent and an AUC of 0.95. Their system uses Grad-CAMs to ensure it localizes areas of infection correctly. The study accurately detects COVID-19 and has differentiated it from other sources of pneumonia (bacterial and viral). It may suffer from an issue of generalizability in that all the CT scans they used came from the same hospital. They suggest including a history of exposure that could further increase their model accuracy and reduce misclassifications for more challenging imaging cases.

Gozes et al. [98] designed a system that concatenates a 3D system (designed by RADLogics Inc) for detecting nodules / focal opacities and a 2D system for detecting and localizing larger-sized diffuse opacities. These diffuse opacities are indicative of coronavirus infection. Existing lung pathology detection solutions that are commercially available often focus on nodule detection and cannot be relied on for detecting global GGOs. This is the motivation the authors have cited for building their 2D ResNet-50 classifier. The classifier depends on a lung segmentation module with a U-net architecture. It has been trained on 6150 CT slices with lung abnormalities that have corresponding masks. Their ResNet-50

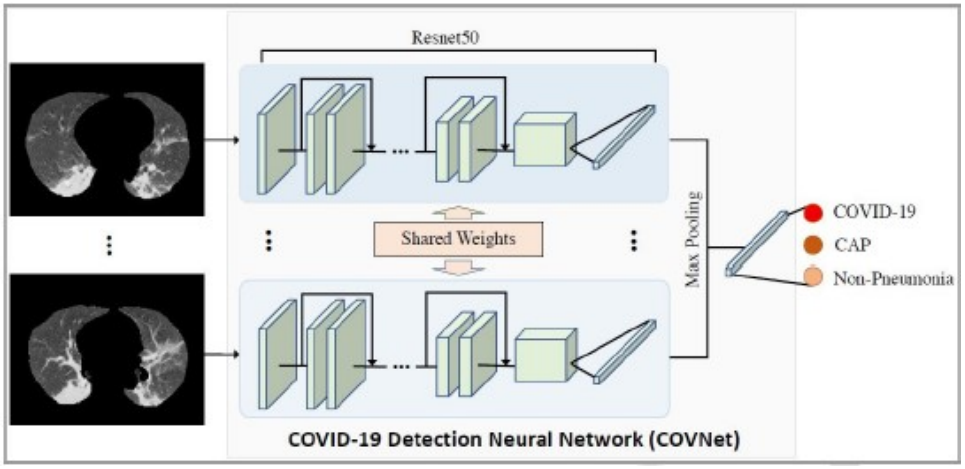


Figure 2.35: Li et al. [97] created this model for extracting 3D features from 2D ResNet-50s.

was pretrained on ImageNet. They mention that they employ data augmentation techniques. They calculate the number of positive slices that are output by the classifier for each patient out of the total number of slices and use a threshold to determine whether the patient should be diagnosed with COVID-19. The dataset used by the authors has 1036 normal slices and 829 COVID slices. They have used in-house radiologists to annotate these slices. To validate their system the authors used Grad-CAMs to localize the areas contributing the most to the classifier's decisions. Their total system uses 2D and 3D localizations throughout the lungs to generate a corona score "that measures the progression of the disease over time." [98] This quantization of a patient's disease burden could be used for the management of patients in the future. Their classification of COVID-19 vs. Non-COVID-19 patients results in an AUC of 0.996, a 98.2 percent sensitivity and a 92.2 percent specificity.

Gozes et al. [99] released another paper with a related but different system that again utilizes the idea of a "Corona Score." This score is derived from the degree of localized corona infected patches in a patient's lungs. This "Corona Score" is used for disease detection and is additionally used as a measure for categorizing the severity of the patient's illness. The model proposed by the team operates exactly as the previous model except it does not use a separate subsystem for 3D nodule detection like before. The model still uses the 2D ResNet-50s of the previously mentioned system and still volumetrically combines slices as before. The difference between Gozes et al.'s previous work [98] and this paper is this paper's use

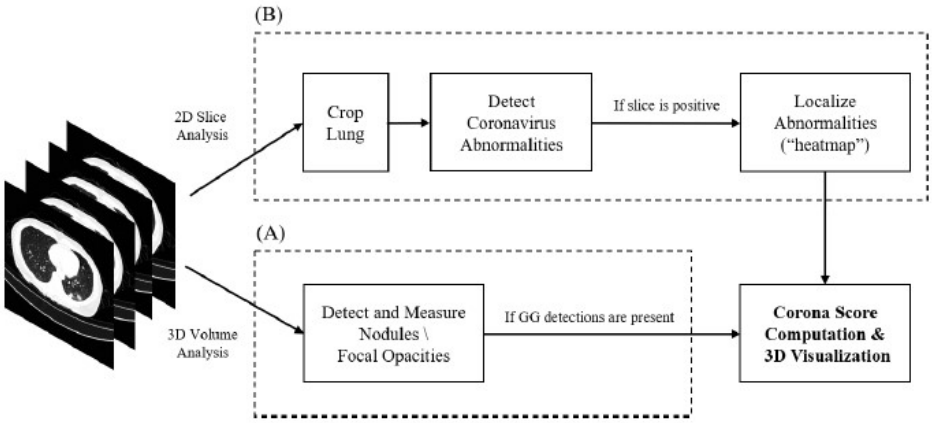


Figure 2.36: Gozes et al.'s [98] model combines a commercial 3D nodule detector with a 2D slices abnormality detector.

of k-means clustering. They use K-means clustering and the elbow method and eventually find the optimal number of clusters to be 3. The 3 classes that the authors found using unsupervised learning corresponded to normal, focal, and diffuse disease manifestations. For positive slices from coronavirus patients (n = 1592 from 110 patients) and negative slices from patients (n = 701 from 81 patients), the final convolutional layer of each 2D input is flattened into a 2048-dimensional feature vector. PCA further reduces the dimensions from 2048 to 2. The space that results can be visualized in Fig. 2.38. This feature map shows that COVID-19 positive and negative patients can be clearly differentiated. The system is attractive because the classifier is not a black box and allows for 3D visualization by a radiologist. The overall system achieves an AUC of 0.948.

Jin et al. [100] developed a 2.5D CT model for detecting COVID-19 using a large dataset with 10,250 CT scans. These scans came from COVID-19, non-pneumonia, CAP, and influenza-A/B patients. The model leverages the advantages of using a pretrained ResNet-152 for slice diagnosis. Lung segmentation is performed prior to classification using a 2D U-Net. After segmentation, the masked slices are input into the slice diagnosis network. Following the slice diagnosis module, a fusion block is used to perform an analysis on the entire 3D volume of a subject. The top 3 slice scores are averaged over the volume and used for diagnosis. Grad-CAMs were used to obtain attentional regions and generate heatmaps of infected regions. An additional COVID-infectious slice diagnosis ResNet-152 module was built

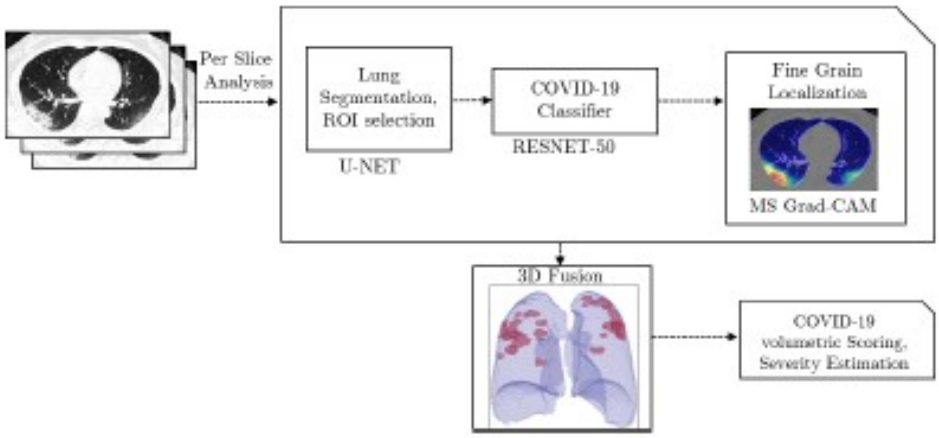


Figure 2.37: Gozes et al.'s [99] model combines a commercial 3D nodule detector with a 2D slices abnormality detector.

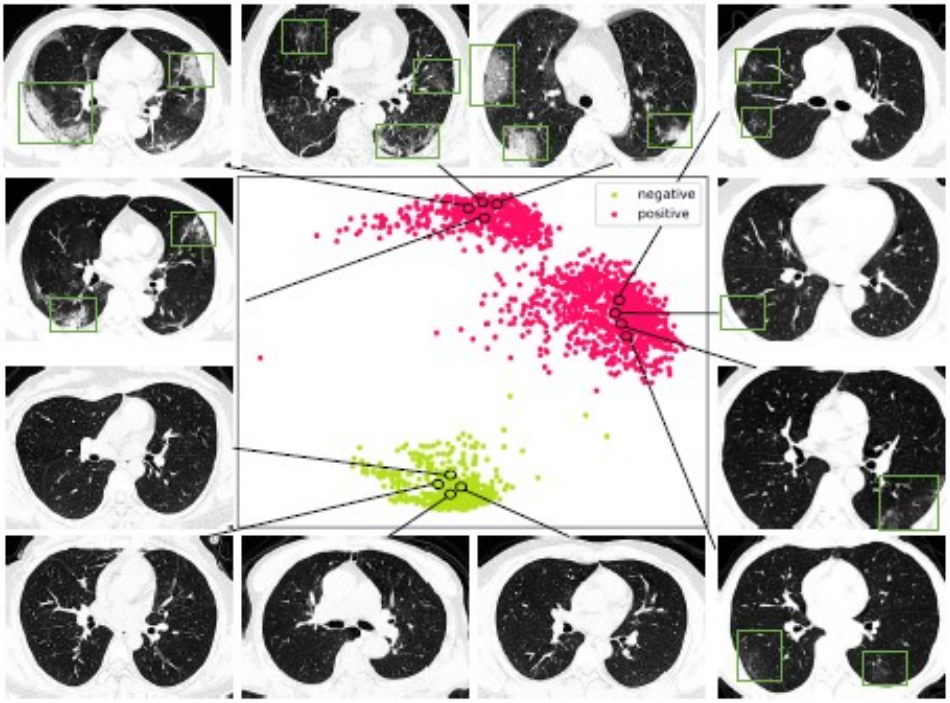


Figure 2.38: Gozes et al.'s [99] Principal Component Analysis (2048 Dimensions to 2 Dimensions).

by the team. It was trained on COVID-19 slices where normal slices or abnormal slices had been manually marked. The COVID-infectious slice diagnosis module had a better ability to locate COVID-19 infected slices than the first classifier (the slice diagnosis module). The

slice diagnosis module was trained on COVID-19, influenza-A/B, CAP and non-pneumonia subjects. Finally, t-SNE [101] was used on features from the slice diagnosis module. The last fully-connected layer of the slice diagnosis module was stripped away and the last layer's features were max pooled with all the other slices. The CT volumes were all mapped to a "2048-d latent which was used to perform t-SNE." [100] The system obtained an AUC of 97.17 percent, a sensitivity of 90.19 percent and a specificity of 95.76 percent. The authors mentioned that they first used a 3D classifier, but eventually abandoned the project due to the 3D classifier memory requirements and a lack of accuracy. The main importance of the methodology in this work lies in how the authors have been able to achieve high accuracy when combining 2D slices for a 3D representation of the data while not creating so many parameters that the system cannot be run on a GPU with approximately 11GB of RAM.

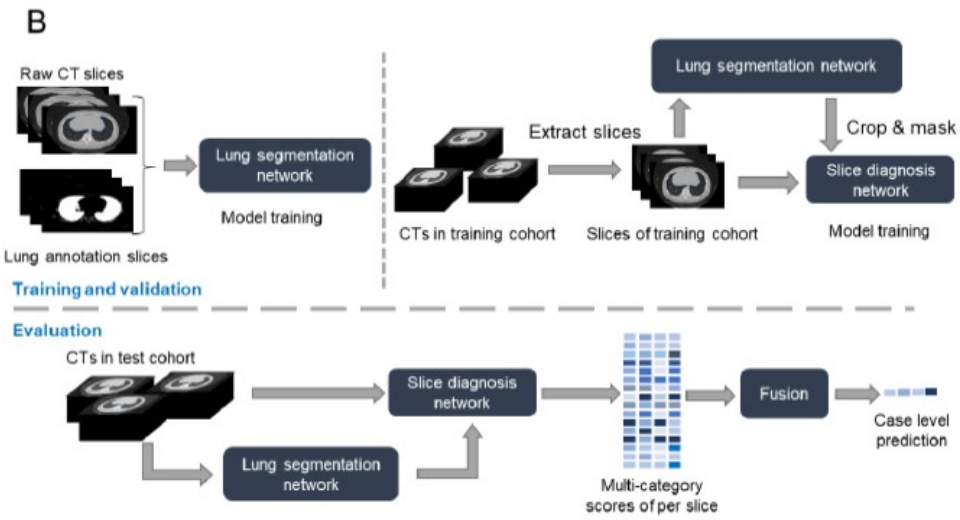


Figure 2.39: Jin et al.'s [100] model where the top 3 slice scores are averaged per volume.

Zhou et al. [102] have developed a 2.5D model that is designed for estimating the disease burden of COVID-19 patients and representing different states of the illness in time. This estimation is broken down into three categories. Early, progressive, and serious states of the disease are tracked. 2D state-of-the-art lung segmentation methods for CT scans often rely only on a single axis direction (z-axis) on which to obtain the slices (on the x-y plane) that will be processed for segmentation. This is not the only fashion in which

real radiologists perform segmentation. True segmentation requires information from all three axes and 3D segmentation units can capture this behavior at a very large and often unreasonable computational expense. There is a need in the medical imaging community to find methods that perform the necessary 3D segmentation on a pathology while reducing a system's parameters, convergence rate, and memory requirements. This paper offers a unique solution that uses a series of 2D slices from the x-y, x-z, and y-z planes. During the manual annotation process, radiologists may not get enough information along one axis and need to use slices along other axes when they run into problems segmenting an area. For every voxel in Zhou et al.'s [102] system, there are three images (Pxy, Pyz, and Pxz) where the probability of each voxel being an infection point is calculated. Intermediate models composing the equation of a voxel being an infection point are individually trained first. An aggregation function over these separate models brings the system to a final prediction. The authors note that there is no current method they are aware of that can normalize a 3D CT signal intensities and dimensions simultaneously. They propose a new preprocessing method that performs spatial and signal normalization steps and makes different 3D volumes fit into a standard volume. Data augmentation techniques help to improve the system given the limited 3D volumes available. The authors have used visualization techniques to ensure that infected regions are being picked up by the team's deep learning model. Their model achieves state-of-the-art segmentation results and outperforms many 3D segmentation units. They used a dataset of 201 CT scans from 140 COVID-19 patients (using 6 different CT scanners) for testing. Segmentation on the largest tested dataset resulted in a dice-coefficient of 0.783. Their disease quantification results were accurate with their model achieving an average error rate of 2.5 percent.

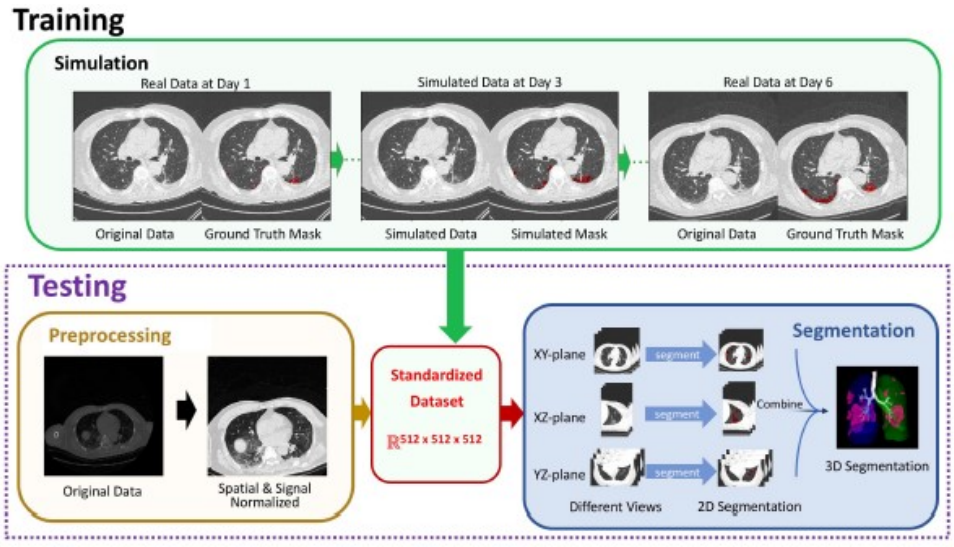


Figure 2.40: Zhou et al.'s [102] model estimating the disease burden of COVID-19 patients.

Table 2.1: Summary of Papers Reviewed

Paper	Dimension	Purpose	M.L. Methods	Datasets	Evaluation
Zhang et al. [38]	2D X-ray	Diagnosis	EfficientNet Feat. Ext. MLP Anomaly Detector	37393 Non-Vir. Pne. Images 5977 Pne. Images	AUC: 83.61% Sens.: 71.7%
Hemdan et al. [13]	2D X-ray	Diagnosis	DenseNet-201 (Best)	25 COVID Images 25 non-COVID Images	F1: 91%
Aposto. et al. [15]	2D X-ray	Diagnosis	VGG-19 (Best)	224 COVID Images 714 Pne. Images 504 Normal Images	2-Class Acc: 98.75% 3-Class Acc: 93.48%
Ozturk et al. [40]	2D X-ray	Diagnosis	DarkNet-19		2-Class Acc: 98.08% 3-Class Acc: 87.02%
Haghanifar et al. [43]	2D X-ray	Diagnosis	ChexNet (DenseNet) Seg: U-Net	780 COVID Images 4600 CAP Images 5000 Normal Images	2-Class F1: 94% 3-Class F1: 85%
Mangel et al. [44]	2D X-ray	Diagnosis	ChexNet (DenseNet)	155 COVID Images 1493 Viral Pne. Images 2780 Bac Pne. Images 1583 Normal Images	3-Class Acc: 90.5% 4-Class Acc: 87.2%
Al-Waisy et al. [47]	2D X-ray	Diagnosis	ChexNet (DenseNet-121)	400 COVID Images 400 Normal Images	2-Class Acc: 99.99% F1: 99.99% COVID Sens: 99.98%
Khalifa et al. [48]	2D X-ray	Diagnosis	GAN ResNet-18	6240 Images	2-Class Acc: 99%
Waheed et al. [49]	2D X-ray	Diagnosis	AC-GAN	403 COVID Images 721 Normal Images	2-Class Acc: 95%
Oh et al. [50]	2D X-ray	Diagnosis Triage	Parallel ResNet-18s Seg: FC-DenseNet-103	200 Viral pne. Images 54 Bacterial pne. Images 57 Tuberculosis Images 191 Normal images	4-Class Acc: 91.9%
Wang et al. [51]	2D X-ray	Diagnosis	Custom ResNet	358 COVID Images 8066 Normal Images 5538 Other Pne. Images	3-Class Acc: 93.3%
Rajaraman et al. [14]	2D X-ray	Diagnosis	VGG-16 VGG-19 Inception-V3 Ensemble: Weight Avg Seg: U-Net	313 COVID-19 Images 2780 CAP Images 7595 Normal Images	3-Class Acc: 99.01% F1: 99%
Wehbe et al. [19]	2D X-ray	Diagnosis	Ensemble: Weight Avg Seg: Cropped Lung Box	4253 COVID-19 Images 10,535 Normal Images	2-Class Acc: 82% F1: 83% AUC: 90.0% COVID Sens: 75%
Yeh et al. [20]	2D X-ray	Diagnosis	DenseNet-121	510 COVID-19 Images 45030 Normal Images 17906 Pne. Images	3-Class Acc: 82% F1: 83% AUC: 90.0% COVID Sens: 75%
Horry et al. [54]	2D X-ray	Diagnosis	VGG-19 Seg: OpenCV GrabCut	100 COVID-19 Images 100 Pne. Images	F1: 81%
Teixeira et al. [59]	2D X-ray	Diagnosis	VGG-16 Seg: U-Net	503 COVID-19 Images 1016 Normal Images 1159 Opac. Images	F1: 94%
Tabik et al. [56]	2D X-ray	Diagnosis	ResNet-50 Seg: Cropped Lung Box	426 COVID-19 Images 426 Normal Images	2-Class Acc: 76.18% COVID Sens: 72.59%
Abdulah et al. [63]	2D X-ray	Diagnosis	Hybrid Convnet Seg: Res-CR-Net	848 COVID Images 1417 non-COVID Images	2-Class Acc: 79.3% F1: 72.3%
Aymer et al. [72]	2D CT	Diagnosis	MLP Seg: U-Net	449 COVID Images 595 Non-COVID Images	2-Class Acc: 86% Dice-coeff: 78.52%
Polsinelli et al. [73]	2D CT	Diagnosis	SqueezeNet	460 COVID Images 397 Non-COVID Images	2-Class Acc: 83% Sens.: 81% Spec.: 85% F1: 83.3%
Ko et al. [76]	2D CT	Diagnosis	ResNet-50	264 COVID Images 1357 Pne. Images 1442 Non-COVID Images	3-Class Acc: 99.75 Sens.: 99.58% Spec.: 100%
Maghdid et al [77]	2D X-ray/CT	Diagnosis	Custom small CNN	85 X-ray COVID Images 203 CT COVID Images 85 X-ray Non-COVID Images 153 CT Non-COVID Images	X-ray Acc: 94% CT Acc: 94.1% X-ray Sens.: 100% CT Sens.: 90%
Alom et al. [78]	2D X-ray/CT	Diagnosis	IRRCNN Seg: NABLA-N	178 CT COVID Images 247 CT Normal Images 1341 X-ray Normal Images 3875 X-ray Pne. Images 67 X-ray COVID Images	CT Acc: 94.1% X-ray Acc: 84.67%
Shan et al. [80]	3D CT	Quantifaction	Seg: 3D VB-Net	549 COVID Scans	Dice-coeff: 91.6%.
Shi et al. [82]	3D CT	Diagnosis	Random Forest Seg: 3D VB-Net	1658 COVID Scans 1027 CAP Scans	AUC: 94.2% 2-Class Acc: 87.9% Sens.: 90.7% Spec.: 83.3%
Tang et al. [84]	3D CT	Quantification	Random Forest Seg: 3D VB-Net	176 COVID Non-severe Pat. 55 COVID Severe Pat.	2-Class Acc: 87.5% AUC: 91%
Wang et al. [18]	3D CT	Diagnosis Prognosis	Custom DenseNet Seg: DenseNet121-FPN	924 COVID Patients 342 Pneumonia Patients Metadata	AUC: 88% 2-Class Acc: 80.12% Sens.: 79.35% Spec.: 81.61% F1: 82.02%
Wang et al. [88]	3D CT	Diagnosis Unsuperised.. Annotation	Custom CNN ResNet Seg: 2D U-Net	313 COVID Patients 229 Non-COVID Patients	AUC : 95.9% 2-Class Acc: 87.9%

Paper	Dimension	Purpose	M.L. Methods	Datasets	Evaluation
Jin et al. [90]	3D CT	Diagnosis	3D ResNet-50 Seg: 3D U-Net++	877 COVID 413 Other Pulm. Disease	AUC: 99.1% Sens.:97.4% Spec.: 92.2%
Mei et al. [92]	2.5D CT	Diagnosis	Inception-ResNet-v2 ResNet-18 MLP Seg: Pixel Thresh.	419 COVID 486 Non-COVID Metadata	AUC: 92% Sens.: 84.3%.
Song et al. [94]	2.5D CT	Diagnosis	ResNet50 Feat Pyr.	777 COVID Images 505 CAP Images 708 Normal Images	AUC image-lvl: 97% AUC Patient-lvl: 99% F1 Patient-lvl: 94% Acc. Patient-lvl: 94%
Bai et al. [96]	2.5D CT	Diagnosis	EfficientNet-B4 Seg: Pixel Thresh. MLP	521 COVID Patients 665 Pneumonia Patients 132583 Slices Total	2-Class Acc: 96% Sens.: 95% Spec.: 96%, AUC: 95%
Li et al. [97]	2.5D CT	Diagnosis	ResNet-50 Seg: 2D U-Net MLP	1296 COVID Scans 1735 CAP Scans 1325 Normal Scans	COVID AUC: 96% COVID Sens.: 96% COVID Spec.: 96%
Gozes et al. [98]	2.5D CT	Diagnosis Quantifaction	ResNet-50 RADLogics Nodule Det Seg: 2D U-Net	829 COVID Images 1036 Normal Images	AUC: 99.6%
Gozes et al. [99]	2.5D CT	Quantifaction	ResNet-50 PCA K-means Clustering Seg: 2D U-Net	1592 COVID Images 701 Normal Images	AUC: 94.8%
Jin et al. [100]	2.5D CT	Diagnosis	ResNet-152 Seg: 2D U-Net t-SNE	2228 COVID Scans 2298 CAP Scans 83 Influenza Scans 3338 Non-Pne. Scans	AUC: 97.17% Sens.: 90.19% Spec.: 95.76%
Zhou et al. [102]	2.5D CT	Quantifaction	Seg: 3 2D U-Nets	160 COVID Scans	Dice-coeff: 78.3%

2.6 Discussion About the Approaches Reviewed

2.6.1 Choice of Dataset

An important aspect of designing a deep learning system for finding the diagnosis or prognosis of a COVID-19 patient starts with the data. What data is available to build a system in terms of images/volumes and metadata is important. It will determine the entire direction of how the project is implemented. All of the studies reviewed here started with a data-gathering stage. Some of the datasets cited in these articles are publicly available. It is clear however that many institutions are keeping information to themselves and not making that information public. The information that is released to a public dataset is often missing some of the additional information researchers would require to begin a project. This is especially true with metadata (age, sex ICU stay, hospital stay, survival) which is necessary for discovering the prognosis of a COVID-19 patient. There is only one study [18] that has been able find the prognosis of COVID-19 patients using metadata and the authors of the study had access to private information that was collected from several regional hospitals. The vast majority of studies [18, 102, 98, 80, 84] attempting to perform prognosis had insufficient metadata and instead relied mostly on imaging quantification methods. While this is important work, extra metadata would have greatly assisted these authors. More often than not most researchers have been focused on diagnosis alone due to this lack of available information. There is currently only a small number of COVID-19 X-ray images and CT scans that are publicly available, but this should change in the coming months as more datasets are released.

Following the data-gathering stage, a plan needs to be developed around structuring the data into a format that can allow a deep learning algorithm to glean meaningful insights from the data. There could be hundreds of gigabytes of data that need to be organized. One consideration among the studies reviewed here that often goes unmentioned is concerning whether the authors have ensured no cross-contamination has crept into the training set and test set. Sometimes public datasets are composed of other public datasets. The datasets different authors have produced and distributed can often be combined and thereafter redis-

tributed with a new name. When this happens, information from the original datasets gets removed. Many datasets for instance do not include patient number in their composition. With no way to determine whether multiple images from the same patient have been included in a dataset data leakage can occur. Many papers do not mention how they ensured their data somehow did not become compromised. If multiple duplicates and/or different images from the same patient are mixed between a learning algorithm's training and test sets, the overall statistics a paper reports cannot be fully trusted. Many papers additionally do not discuss the variance in their image datasets given the different imaging systems and protocols that were responsible for generating the images in their datasets.

One extremely concerning problem regarding some X-ray datasets in wide distribution is their use of Kermany et al.'s dataset that is constructed from "5,232 chest X-ray images from children" [16]. This dataset started incorrectly being used by researchers to have normal and non-COVID-19 pneumonia X-rays to compare COVID-19 X-rays against. The children in this dataset are between one and five years old and the dimensions and features of their lungs should not be used in a learning algorithm against adult lungs. There are so many COVID-19 datasets on Kaggle and other dataset platforms that use this dataset that caution is urged when accessing public datasets. It is good practice to ensure that datasets compiled by other individuals mention the sources of all of their original images.

2.6.2 Purpose: Diagnosis and Prognosis

After gathering the data, a decision needs to be made about whether the system can be built for diagnosis, prognosis, or both. From a diagnosis perspective the classes a model predicts will have consequences on how the deep learning system can be used. Some papers focus solely on binary classification [38, 13, 48, 49, 72, 73, 77, 78, 92, 96, 98]. Binary classification in some of the reviewed studies was viewed as acceptable so long as the model was trained on a sufficient amount of non-COVID-19 illnesses that closely resemble COVID-19 (viral pneumonia, bacterial pneumonia, etc.). If a system has been constructed to diagnose COVID-19, there needs to be a way for that system to differentiate the scans of COVID-19 patients from other closely related illnesses. This would be important in a clinical setting

if a suspected COVID-19 patient obtained a negative RT-PCR test result. Other studies [44, 50, 51, 14, 76, 80, 18, 88, 90] believed that multiclass classification was important as well. If a patient could have the flu, bacterial pneumonia or COVID-19, it would be nice to have a model that could inform healthcare professionals what the patient is suffering from. That way the healthcare professional could develop a personalized approach to dealing with each patient. Some studies [15, 40, 43] employed both binary and trinary diagnosis models and compared the two approaches. All of these approaches to diagnosing COVID-19 could be acceptable depending on the end application and data availability in the data-gathering stage.

To determine the prognosis of a COVID-19 patient, it clearly must first be determined whether a patient has COVID-19. A prognostic system might proceed based on a patient having undergone an RT-PCR test. A few of the quantification and prognosis systems in the AI medical imaging literature worked by diagnosing a patient with COVID-19 using imaging technology first. This step adds extra clarity to a patient's original diagnosis if the image classification system is sufficiently sensitive at differentiating COVID-19 from other illnesses. The metadata associated with COVID-19 radiological images will be of the utmost importance for determining the future course of a patient's illness. There is currently only one CT study so far that in a very limited way was able to glean insights by combining radiological images and metadata [18] to classify patients as high-risk or not. The study used a Kaplan-Meier analysis to show "high- and low-risk groups had a significant difference in hospital stay time." [18]

2.6.3 Hardware Considerations

There are hardware considerations to consider when implementing a deep learning imaging system. If a project moves ahead with using CT scans, a purely 3D CNN system is hard to fit on high-end workstations. Even workstations with a GPU larger than 8GB RAM will struggle to fit a 3D model. Jin et al. [90] for instance, mentioned that their 3D system required 8 TITAN X GPUs for training. VRAM (Video RAM) is often the largest bottleneck in terms of whether an engineer will be able to implement an experimental

architecture. If the GPU cannot hold the model, the experimental model may never get off the ground. An example of this in our survey was in a 2.5D CT study conducted by Jin et al. [100]. They initially tried to implement a 3D model but eventually found that they did not have the necessary hardware and were forced to move to designing a 2.5D model. Their workstation used an 11GB GPU and this limited the team's options. Waheed et al. [49] were implementing a GAN as a part of their system and they as well found that their hardware limited them to using images that were 112 x 112.

If a team's workstation uses less VRAM, this will result in more data transfers between the CPU and GPU. This is likely going to matter more than the number of tensor cores an engineer has available in a GPU. The hardware requirements of computer vision, therefore, prioritize purchasing GPUs with a higher amount of VRAM. Extra space is also required in terms of a system's CPU/RAM requirements. When originally fitting a model on a CPU's RAM, it is best to use only 80 percent of the available RAM or the system may start to implement paging and slow down the performance of the entire system. Resource management from a computer architecture perspective is extremely important for implementing deep learning computer vision systems. The system's resource availability as well its accuracy requirements will ultimately determine which choice of model is designed.

2.6.4 Resolving the Class Imbalance

Resolving the class imbalance of COVID-19 vs non-COVID-19 datasets was an important consideration when designing deep learning systems in many of the articles reviewed here. There are far fewer COVID-19 scans in many datasets than the X-ray and CT-scans that are taken from other sources. In binary and multiclass classification this can result in an algorithm that will see mostly non-COVID-19 examples. A deep learning algorithm usually has equal amounts of images in each class. One solution to this that is mentioned in the literature is to weight the loss function in a manner whereby the smaller number of COVID-19 patients in the dataset has more weight than the normal examples. An example where X-rays with COVID-19 are underrepresented in a dataset is shown in Fig. 2.41. This procedure was mentioned by Hagnifar et al. [43], Mangal et al. [44], and Rajaraman et al.

[14] in dealing with their studies' class imbalances of COVID-19 X-ray images. Another commonly used method in tackling the class imbalance problem is resampling. Alom et al. [78] used resampling in their paper where they "applied class specific data augmentation" [78] to boost the amount of COVID-19 images in their dataset. Resampling involves shuffling the examples in a dataset to create an even distribution of classes to analyze. Resampling can involve using fewer images from the class with more examples. It can also involve using more images from the class with fewer examples multiple times. Ultimately an even distribution is input into a CNN. An example of a system which shuffles images to achieve a normal distribution can be seen in Fig. 2.42. An alternative to the aforementioned strategies is mentioned by Zhang et al. [38]. The authors there used anomaly detection and their system was trained on a larger dataset with no exposure to viral pneumonia. The authors then used a small viral pneumonia dataset and tested their model to see if it could be picked up by the anomaly detection unit.

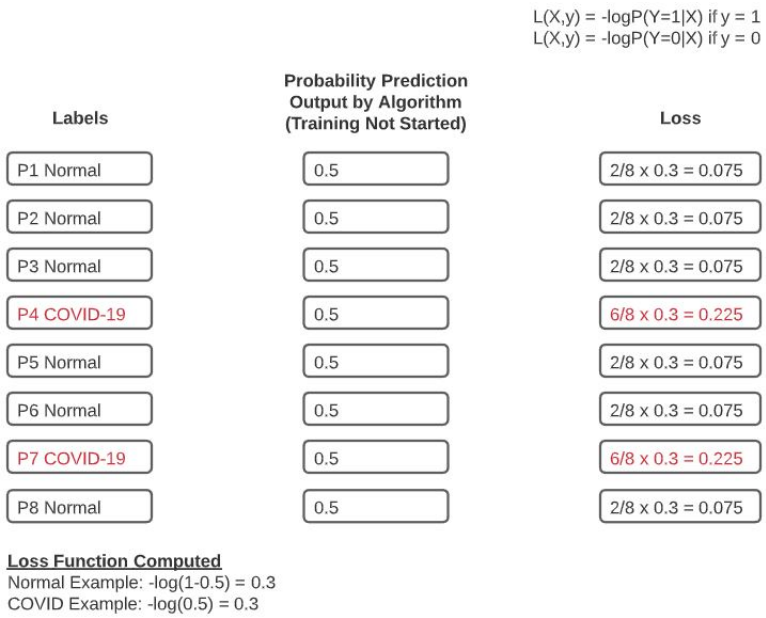


Figure 2.41: Adjusting the weight of the loss function to correct for class imbalance.

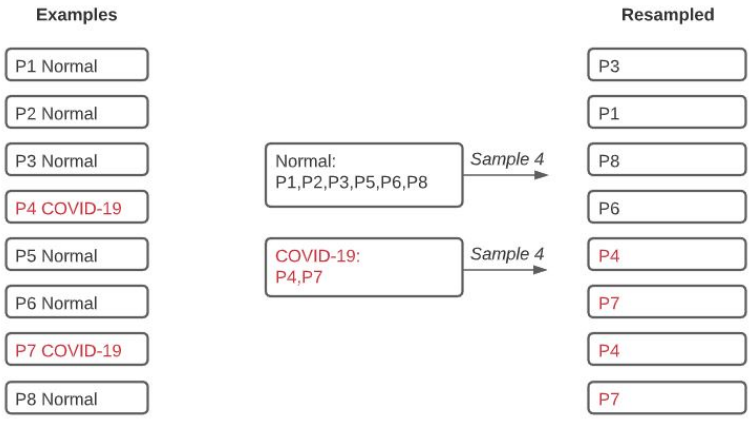


Figure 2.42: Example of resampling to correct for class imbalance.

2.6.5 Preprocessing and Segmentation

Segmentation generally always improves the accuracy of a system. The systems that achieved the best performances usually included image segmentation. Some systems segment out the lungs alone, while others segment out specific lung areas. Depending on the annotated data made available, a designer must choose the kind of segmentation to be used. There are 2D and 3D segmentation systems that are publicly available to be adapted to a project. 3D segmentation units preserve original spatial relations between the slices of a CT scan. Using a 2D segmentation unit on all the slices of a CT scan is common but some spatial information is lost. The training of a 3D segmentation module however may require multiple GPUs and medical staff to annotate entire volumes. Segmenting an entire 3D volume is ideal, but sometimes to do so within existing hardware constraints the spatial resolution (256 x 256 for instance) along a 3D volume needs to be reduced (perhaps to 100x100 as an example). This also depends on the batch sizes being used. A higher batch size could likely be used on 2D slices. A 3D system may require a stochastic batch size of 1 for a 3D volume. Simple thresholding for lung segmentation can also be effective in that lung tissue exists within certain intensities in CT scans. Bai et al. [96] for instance segmented each slice using attenuation (-320 HU used as a threshold value) to exclude non-pulmonary regions in CT scans. Mei et al. [92] followed a similar procedure. Mei et al. [92] normalized pixel intensities to exist within a 0-255 window and afterward the lung region was defined

to exist underneath a threshold intensity of 175. They also found that small regions of the lungs with pixel intensities less than 64 needed to be removed due to random noise. Fitting a segmented organ within threshold regions therefore is quite common. Normalizing pixel intensities and the scale of images/volumes is standard in virtually all papers. Song et al. [94] performed an operation in their 2.5D CT model to remove slices of incomplete lung as a preprocessing step. Many deep learning computer vision systems perform better with a preprocessing step that removes less helpful images before training.

2.6.6 Transfer Learning

CNNs tend to be the most popular machine learning methods used in most of the studies reviewed, which is unsurprising given their recent success in imaging competitions. The majority of studies favour using transfer learning. Many deep learning networks require millions of training samples and computer vision problems often have only hundreds or thousands of samples. This is why transfer learning using image datasets like ImageNet is so popular in biomedical imaging. The medical industry often makes it difficult for researchers to obtain datasets due to privacy concerns. Transfer learning using X-rays for diagnosing COVID-19 is easier for two reasons. The first reason is that X-rays are 2D images and utilizing 2D image datasets for transfer learning is easier. There is no ImageNet for 3D volumes in medical imaging. The 2nd reason is that there are larger high-quality datasets with hundreds of thousands of chest X-rays specifically made for diagnosing pulmonary diseases. This allows for modality-specific transfer learning whereby a CNN that was designed for 14 pulmonary diseases like ChexNet [45] can be fine-tuned on another chest X-ray problem in diagnosing COVID-19. Modality-specific transfer learning has been shown to increase the efficacy of transfer learning on new problems. ImageNet is a fine resource, but in application-specific circumstances, modality-specific transfer learning is more effective than using ImageNet alone. CT scans do not have the same availability of resources for performing modality-specific transfer learning. There are many more X-ray pneumonia samples online than there are for CT scans. ResNet, DenseNet, VGG, Inception and SqueezeNet architectures are some of the most popular off-the-shelf architectures found in the literature.

In the literature reviewed here, custom 3D CNNs are often designed for 3D systems. The 3D systems used in the literature use shallower CNNs than the CNNs often used in 2D studies. 2D Multimodal diagnosis systems in the literature review were mentioned briefly that use both X-rays and CT scans. The primary multimodal system mentioned here [77] could diagnose COVID-19 using either X-rays or CT scans. It would have been interesting to see if X-rays and CT scans on the same patient could have been combined in training, but no such study has ever been conducted. Current datasets do not contain many examples where both the X-rays and CTs of individual patients are available.

Unsurprisingly, a majority of the studies reviewed here used transfer learning. This was especially true for most 2D systems. Leveraging transfer learning is more effective than training a CNN from scratch with very little data. A challenge for many deep learning architectures is that they are very data-hungry algorithms. Biomedical datasets are often smaller than a million images and it is common to only have thousands of images when diagnosing pulmonary pathologies. This problem is compounded by how recent the pandemic is. Many medical institutions are mainly trying to deal with a flood of patients rather than concerning themselves with organizing and releasing data for developing deep learning algorithms. With the small datasets currently available, transfer learning is one of the main tools available to imaging specialists in the AI community. In transfer learning, it is generally understood, that the earliest layers in a CNN capture low-level image features like the edges of an object. The later layers of a CNN learn to identify entire portions of an image. When training with ImageNet, the early layers are generally the most useful and the later layers (which identify whole random objects in photographs) are less useful. In medical imaging, a CNN may first be trained on ImageNet, but afterward, be fine-tuned on the medical dataset. There are a couple of ways to do this. One way is to freeze all of the layers in a CNN except for the last layer or last couple of layers. The last layer(s) can then be trained on the medical dataset. A software engineer may otherwise consider not freezing any of the layers. In doing so, the engineer might initialize a model with ImageNet's weights and then train the CNN in an end-to-end fashion.

2.6.7 Optomizers and Hyperparameter Optomization

Many of the studies reviewed here use the Adam optimization algorithm [43, 76, 78, 96] in a CNN to update their network weights in iterative training. The reasons cited are that the method is computationally efficient, easy to use, does not require as much memory as other optimizers, and leads to high accuracy when implemented in deep learning models. The Adam optimizer is widely used and can quickly train CNNs with reasonably accurate results. As seen in Fig. 2.43 and Fig. 2.44, it outcompetes many other optimizers on popular image datasets. Stochastic gradient descent, however, is also used in some papers [13, 14] and with a well-chosen learning rate converges better than the Adam optimizer in certain cases. The learning rate is typically a hyperparameter that receives a lot of attention in papers. It is the hyperparameter that determines how much a model responds to the model's generated error every time the model's weights are updated. It is often the most important hyperparameter when adjusting a model for optimal performance. Using a learning rate scheduler is another option. If the learning rate is too large, the system might not converge (unstable training) or a designer could end up with sub-optimal final model weights. Other hyperparameters such as batch size however are also closely paid attention to in the studies reviewed here. There a several techniques researchers have available to help them settle on an optimal set of hyperparameters. Bayesian optimization is a hyperparameter optimization technique Polsinelli et al. [73] discussed using for optimizing their model.

2.6.8 System Generalizability

One of the major barriers to deep learning technologies being applied in clinical situations is system generalizability. Achieving system generalizability in one country does not mean that the system will generalize to another population in another country. We first need to use an external validation set formed from a group of patients in the other country. If the system cannot generalize to the population in another country, the deep learning system may be later fine-tuned on a population of patients in the new country so that it functions appropriately. Another consideration when thinking about system generality is related to the type of device used to take an X-ray or CT scan. An X-ray machine or CT scanner may

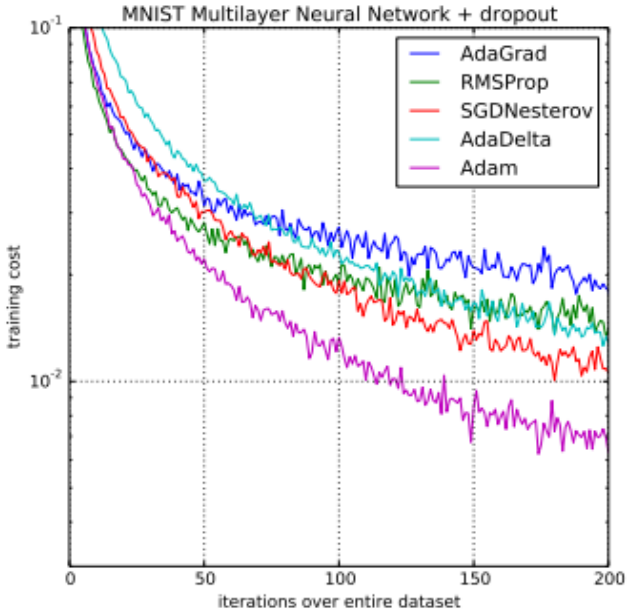


Figure 2.43: Adam optimizer compared with other optomizers on the MNIST dataset. [103]

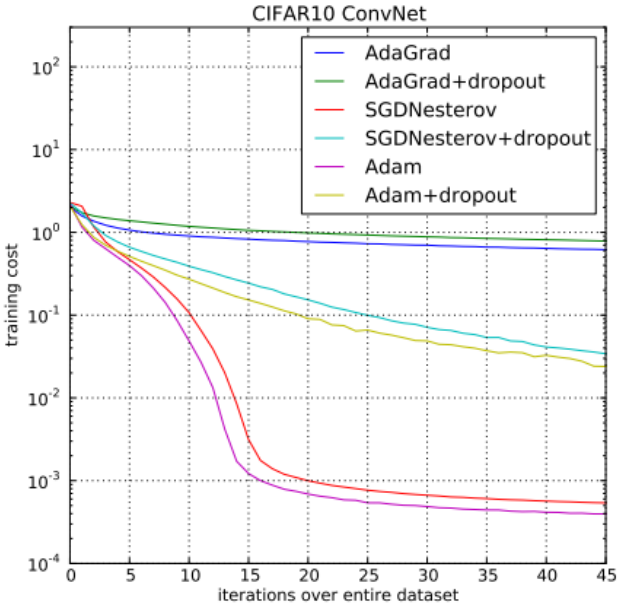


Figure 2.44: Adam optimizer compared with other optomizers on the CIFAR dataset. [103]

have different settings and different protocols in terms of how a patient is positioned. A liquid contrast agent is often injected into patients to get improved images. Different contrast agents may lead to different results. Contrast materials can greatly assist a radiologist to

distinguish abnormal patients from normal patients. The position of a person can change in radiological imaging. Patients undergoing an X-ray commonly have both lateral (patients positioned on their side) and frontal X-rays taken. A deep learning model constructed with both of them either needs to filter out lateral X-rays or needs to be fine-tuned to work on both of them.

Figure 2.45: A major barrier to applying deep learning technologies in medical imaging is system generalizability

2.6.9 Saliency Maps

A key point that can be derived from the COVID-19 deep learning literature is that a designer often cannot evaluate a CNN on the basis of its numerical metrics alone. Using the F1-score, sensitivity, specificity and accuracy of a model is important, but deep learning algorithms can often focus on incorrect details and achieve great results on small datasets. Increasing the size of a dataset will help train a CNN to focus on more relevant details. The way to figure out if the current system is correctly identifying features in an image is to use a saliency map. Many good papers in the literature made extensive use of them when determining if a system is localizing COVID-19 correctly [43, 50, 51, 14, 73, 76, 78, 18, 98, 100]. It takes a trained expert to determine whether the visual features of an image were correctly identified by a CNN. Saliency maps are informative in that they can show if a system is being deceived by features that have nothing to do with COVID-19. The system could be focusing on an area outside of the lungs for instance and that would indicate an

issue with the algorithm. An example showing some saliency maps from a chest X-ray can be seen in Fig. 2.46. Segmentation always helps generate better saliency maps and allows a learning algorithm to focus on the most relevant areas.

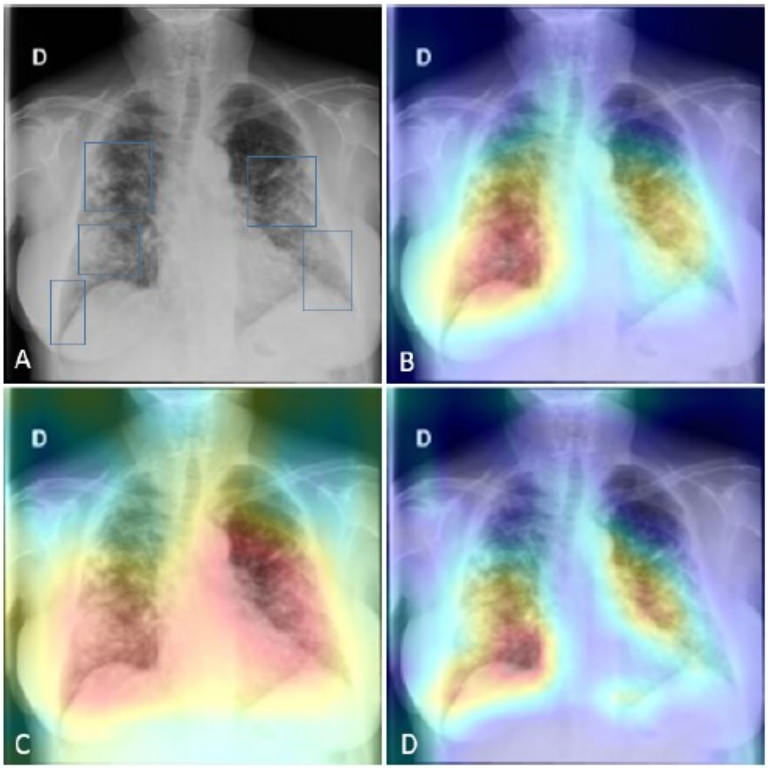


Figure 2.46: Inspecting saliency maps for infection localization performance [14]

2.7 Choosing a Modality and Narrowing our Scope

At the end of this literature review, we determined that it was best to proceed with building an X-ray-based deep learning diagnostic model. While a project based on CT scans may have been rewarding, there were significant hardware restraints for following through with such a project. A 2.5D or 3D CT project would, unfortunately, require several high-performance GPUs. The hardware requirements for segmenting CT scans are even more severe. Our research budget only allowed room for using a CPU and GPU hosted on a Google Colab Pro server. In the absolute best-case scenario, Google Colab Pro allows a user

to use an NVIDIA Tesla K80 GPU for 24 hours. There are, however, interruptions to the service and this 24 period is not guaranteed. Over the time that we have used Google Colab Pro, we have found that there are a sufficient number of interruptions throughout any given day that one can never assume that 24 hours of service will be available.

There are significantly fewer COVID-19 CT scans available in public datasets than COVID-19 X-ray scans. We came to eventually realize over time that a greater number of X-ray datasets were coming online throughout the pandemic. This increased availability of images in conjunction with our restricted computer resources caused X-ray scans to be the most practicable modality to be used in our project. As we discussed earlier, there are several advantages that X-ray scanners have over CT scanners in terms of how they are deployed in a medical environment. We decided, therefore, to proceed with X-ray scans for the duration of this project. While it would have been an appealing option to incorporate metadata over the course of this project, sufficient amounts of metadata accompanying X-ray images never became available. We, therefore, removed COVID-19 prognosis from our project's scope and instead focused on COVID-19 diagnosis alone.

Chapter 3

Neural Networks, Classification, and Segmentation

3.1 The Basics of Neural Networks

Neural Networks (also referred to as ANNs) are algorithms modeled originally on neurons contained in the brains and nervous systems of most animal species. Understanding the biological motivation of ANNs can be helpful in initially understanding how ANNs function. Originally modeling small biological neurons proved difficult. Neuroscientists required relatively large naturally occurring neurons from different species to visually observe and physically measure these systems. Eventually, by the 1950s, Frank Rosenblatt developed the first ANN. A typical neuron in an animal species is comprised of many dendrites, a cell nucleus, and an axon. The dendrites of a biological neuron input the signals of other cells into the cell nucleus. The nucleus contains a threshold function determining whether the inputs into the neuron have a high enough potential to trigger an output. If this threshold is passed, an output signal is carried away from the cell body via the cell's axon. The output signal is eventually sent to other cells residing alongside the axon's terminals. Fig. 3.1 shows an illustration describing the picture more clearly. This simple illustration helps build an understanding of how some of the simplest neural networks like logistic regression function.

Logistic regression is an algorithm that is useful when working on a binary classification problem. It is typically the first neural network presented to new students being introduced to the subject. When logistic regression is given a feature vector (x), the algorithm will

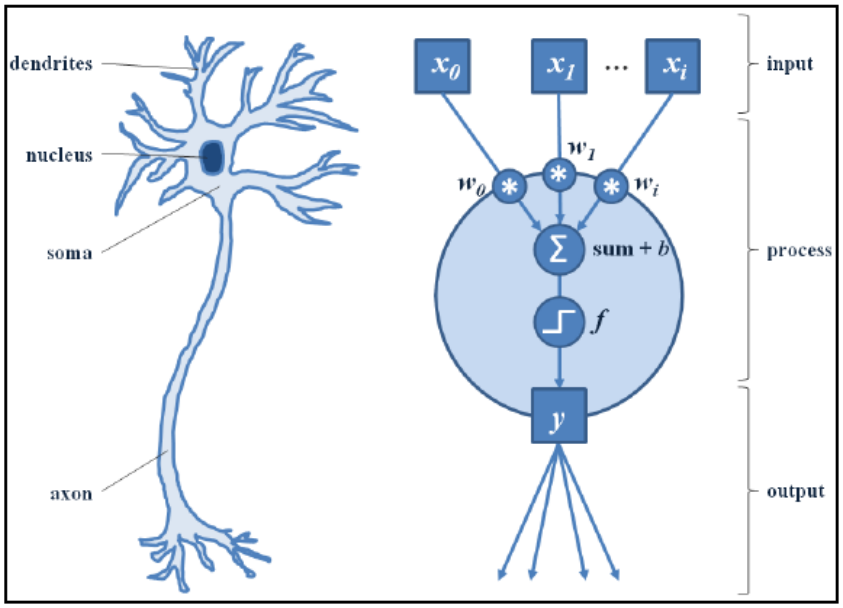


Figure 3.1: The biological comparison of a neuron with a neural network. [104]

output a prediction ($\hat{y}$) which is an estimate of the output (y). The training set to the algorithm will have multiple training samples ($x^{(i)}$) that get fed into the algorithm. In an ANN, each input into a neuron has a weight associated with it (w) that is multiplied with a part of the feature vector (x). These weights are summed within the neuron and an offset (b) is added as well. This summation is passed through an activation function (sigmoid function for binary classification) and if the predesigned threshold is reached, the neuron outputs a positive prediction. The part of logistic regression we so far have described is called forward propagation. A figure describing this system can be seen next to the biological neuron in Fig. 3.1. The equations describing forward propagation in logistic regression can be seen in equations 3.1 and 3.2:

$$z = w^{\mathrm{T}} x^{(i)} + b \tag{3.1}$$

$$\sigma(z) = \frac{1}{1 + e^{\ z}} \tag{3.2}$$

Logistic regression needs a way of determining if the outputs generated during forward propagation are correct. It is classified as a supervised training algorithm because during training it is fed the correct answers (y) for each feature vector (x). This allows for a determination regarding whether the algorithm is outputting correct predictions. The resulting error information is fed back into the algorithm's weights in a process known as backpropagation. To train the parameters w and b, the model requires both a cost and a loss function. The cost function is modeled by summing the loss functions of the training examples and dividing that sum by the number of training examples. The loss and cost functions in logistic regression are defined in equations 3.3 and 3.4 respectively as follows:

$$L(\hat{y}_i, y_i) = -y_i log(\hat{y}_i) – (1 – y_i) log(1 - \hat{y}_i) \tag{3.3}$$

$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}_i, y_i) \tag{3.4}$$

Gradient Descent as a part of logistic regression can now be implemented on the cost function we have derived. The goal of gradient descent is to minimize the cost function and discover the best parameters that optimize for that function. An illustration showing a simple version of gradient descent can be seen in Fig. 3.2. The plot ignores b for now and only shows a simple one-dimensional plot that takes w into account during updates. Gradient descent performs the following operation when considering both w and b in equation 3.5:

$$\begin{aligned} &Repeat[\\ &\quad w := w - \alpha \frac{\partial J(w, b)}{\partial w} \\ &\quad b := b - \alpha \frac{\partial J(w, b)}{\partial b} \\ &\quad] \end{aligned} \tag{3.5}$$

In Fig. 3.2, the slope of the tangent (taking the derivative) where w starts is negative. When we multiply alpha (the learning rate) by a negative number, we end up increasing w

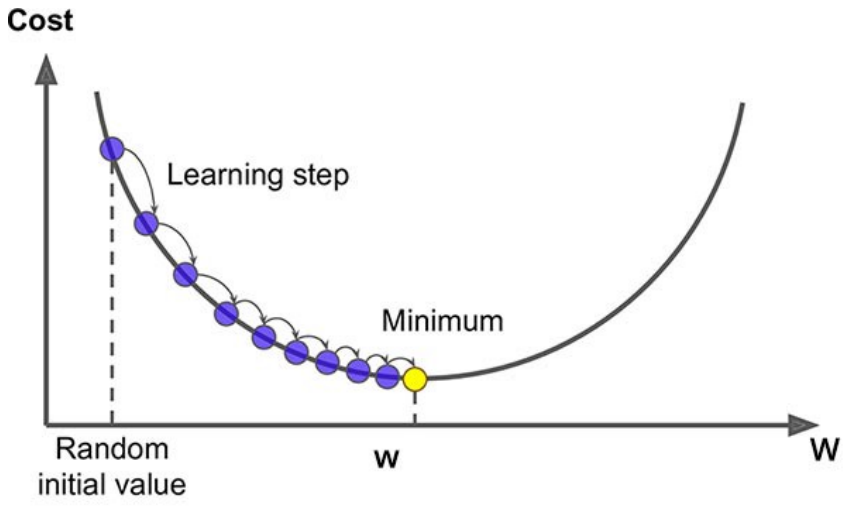


Figure 3.2: Gradient descent along one dimension. [105]

as we move to the local minimum point (towards the right). The inverse can be shown if w start on the other side of Fig. 3.2. In that case, gradient descent slowly decreases w to move towards the local minimum. As the w parameters are updated, the error decreases, and the algorithm eventually converges on the minimum point. There are different forms of gradient descent. The batch size for gradient descent is the hyperparameter that determines how many samples will be used before updating the weights over an iteration. The batch size for gradient descent is a useful hyperparameter that can sometimes assist a learning algorithm in achieving better results. Batch gradient descent uses all of the samples in a training set during each iteration of training. This can result in an extremely slow learning algorithm for problems with a large training set. Stochastic gradient descent uses only a single sample over each iteration of training. Mini-batch gradient is probably the most common and allows for different batch sizes. This allows for faster training than batch gradient descent and also allows for a higher accuracy than stochastic gradient descent. Batch gradient descent converges on global minima over a smooth function. Mini-batch and stochastic gradient descent are noisier and may not directly converge on a global minimum but hover around it.

We have briefly summarized the basics of logistic regression to show the simple case where we are only using one neuron. ANNs, however, were developed to work over a series of layers. If we were to add another layer of neurons in front of the logistic regression algorithm

we were discussing, we would not just be computing the derivatives for w and b in a single layer. The w and b parameters of all the neurons in each layer would need to be computed. Introducing more layers often leads to longer training times. Better results, however, can be achieved. More input data is often required when constructing deeper networks. There is a wide variety of layers that we have used in our research studies, and in the following sections, we introduce their basic structures. While doing so, we highlight how these layers are trained and initialized.

3.2 The Training and Construction of Neural Networks

3.2.1 Weight Initialization

When training a neural network, it is important to initialize its weights to nonzero values. Initializing the bias terms to zero values is acceptable but initializing the weight terms to zero values is a mistake. If each row of weights is initialized to zero, the derivative for every column of weights (w) remains the same. During weight updates, new features can then not be learned on successive iterations. The topic of weight initialization, however, is deeper than that. There are many weight initialization schemes that have been put forward by the AI research community that are useful. Initializing the weights of your neural network to the weights of a network trained on a similar task can greatly increase the accuracy and training speed of your network. On training simple fully-connected networks, we should also decide on more parameters than the weights (w) and biases (b) alone. In deep learning textbooks, authors refer to the weights and biases of a network as parameters. Other designer choices like a network's number of layers, learning rate, number of hidden units, and activation function are referred to as hyperparameters.

3.2.2 Dataset Division

One of the main considerations a designer must initially make when designing a neural network is the division of the project's data. Splitting a project's data into a training set,

validation set, and test set is a common division in the literature. In computer vision projects, a designer usually has a limited number of images. The division of the data in computer vision projects, therefore, often favors larger validation and test sets. In computer vision problems with small datasets, it is common to see 60/20/20 splits and 80/10/10 splits. In an ideal big data scenario, however, the validation and test sets may only be 1 or 2 percent of the overall data available. When choosing a validation and test set, it is best to consider from what sources the original data has been derived. It is often best to not use the same distribution of images in the training set, validation, and test set. As a rule of thumb, it is common for the validation and test set to come from the same distribution and for the training set to come from a larger but different distribution. A designer often spends a lot of time choosing hyperparameters to optimize a neural network's performance on the validation set. It is appropriate, therefore, for the test set to come from the same distribution as the validation set. The network should train, however, on a completely different set of images to avoid biasing the network via cross-contamination. If the final trained model works well on both the validation and test set, a designer can have confidence that the model generalizes well.

3.2.3 Underfitting Versus Overfitting

A common scenario most machine learning practitioners find themselves in involves deciding on whether a network is either underfitting or overfitting. A designer ultimately wants to make the training error small and the gap between the training error and the test error small. According to Goodfellow et al., "Underfitting occurs when a model is not able to obtain a sufficiently low error value on the training set. Overfitting occurs when the gap between the training error and the test error is too large" [106]. A couple of good illustrations showing both cases of underfitting and overfitting can be seen in Figs. 3.3 and 3.4. Both situations present problems. Finding a balanced mix with neither underfitting or overfitting results in an optimal model that generalizes accurately. Large neural networks tend to be prone to overfitting, while small neural networks tend to be prone to underfitting. Adjusting the number of layers in a neural network can have an effect in either direction. Getting more

training examples can help with fixing high variance (overfitting). Adding more features into a network can help with fixing high bias (underfitting). Adding regularization to a model is common and often helps to reduce overfitting. In the case of logistic regression, this involves adding an additional expression to the cost function of logistic regression shown here in equation 3.6:

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}_i, y_i) + \frac{\lambda}{2m}\sum_{i=1}^{m}(w_i^2) \tag{3.6}$$

The additional regularization lambda hyperparameter when increased can fix high bias. When it is decreased, it can fix high variance. This extra term penalizes a neural network's weight matrices for being too large. Setting the regularization parameter to be large incentivizes the weight matrices to end up closer to zero. This will reduce the effect of many neurons in the network, creating a sparser network.

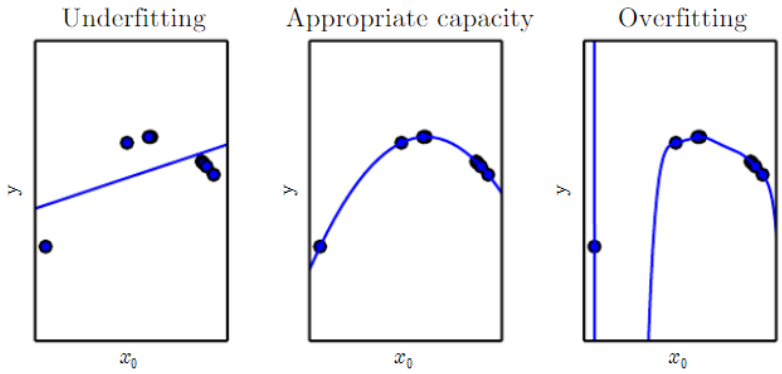


Figure 3.3: Underfitting vs. overfitting. [106]

3.2.4 Dropout

Another common technique for regularization in computer vision is dropout. This technique over every iteration assigns a probability to each node in a layer and determines which nodes are eliminated. It allows a neural network to not concentrate its weights only

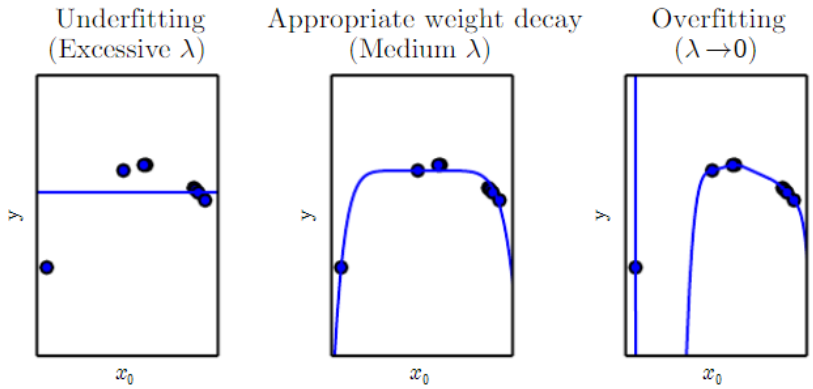


Figure 3.4: Underfitting vs. overfitting when choosing lambda. [106]

on certain nodes and allows for learning from other nearby nodes. An illustration of this concept can be seen in Fig. 3.5. In this illustration, each node in both layers is assigned a probability of being dropped over the course of every iteration. Dropout in computer vision systems is commonly used after multiple fully connected layers towards the tail end of a CNN. Some computer vision researchers use dropout in all their learning algorithms, although preferences vary.

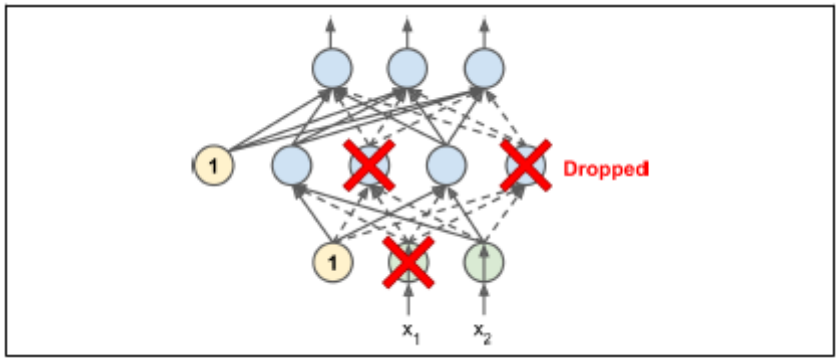


Figure 3.5: Dropout. [105]

3.2.5 Input Normalization

Normalizing the data that is input into a neural network is common across all branches of deep learning. After normalization, the cost function of a neural network looks more

symmetric, and gradient descent is allowed to learn more quickly. It helps the parameters of the network to be updated in equal proportions, allowing for a higher learning rate. Without normalizing inputs, larger parameters tend to dominate in the network, and gradient descent may require many steps to eventually reach the global minimum. This feature scaling is illustrated in Fig. 3.6. The left part of Fig. 3.6, shows a picture of gradient descent that has an easier path towards the global minimum. On the right side of the figure, the first feature is larger than the second, causing descent towards the global minimum to be constricted. In this scenario, a small learning rate is required. The first common step in normalizing inputs is to subtract every sample by the mean of the training set. The second common step is to normalize the variance in the training set. The mathematical expressions seen in equations 3.7 and 3.8 are commonly implemented in the code of neural networks that use input normalization:

$$mean:$$

$$\mu := \frac{1}{m}\sum_{i=1}^{m} x_i \tag{3.7}$$

$$x := x - \mu$$

$$variance:$$

$$\sigma(z) := \frac{1}{m}\sum_{i=1}^{m} x_i^2 \tag{3.8}$$

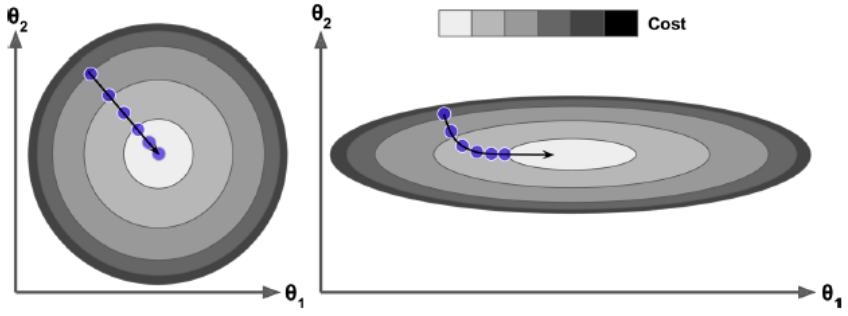


Figure 3.6: Gradient descent with (left) and without (right) feature scaling. [105]

3.2.6 Batch Normalization

While input normalization has been around for a long time, in a paper in 2015, Sergey Ioffe and Christian Szegedy invented an extremely useful idea called batch normalization [107]. It is used in many famous CNNs that have been published over the past five years. For framing the motivation behind developing this technique, Goodfellow et al. in their popular textbook on deep learning wrote in 2016: "Very deep models involve the composition of several functions or layers. The gradient tells how to update each parameter, under the assumption that the other layers do not change. In practice, we update all of the layers simultaneously" [106]. What the authors are saying here is that while weights are being updated during backpropagation, the algorithm assumes all layers are fixed. That is not in reality what is happening. The only layer that a designer can count on being fixed is the network's input layer, which is dealt with using input normalization. The layers behind the layer being updated are in flux. With all layers changing during an update, the weight updates in the network are constantly chasing a moving target. This phenomenon in the industry is called internal covariate shift. In deeper networks, the problem is especially pronounced, and early layers can cause great disruptions to the weight updates of downstream layers where this effect is amplified. Batch normalization was designed to account for changing parameter values in the shallow layers of deep neural networks. If we were trying to update layer three of a neural network, we would need to know what to expect from the output of layer two. What batch normalization does is normalize the outputs of the previous layer (layer two) so as to train the parameters of the next layer more effectively (layer three). In the field of computer vision, this process is also called 'whitening.' In their paper introducing the world to batch normalization, Sergey Ioffe and Christian Szegedy state: "By whitening the inputs to each layer, we would take a step towards achieving the fixed distributions of inputs that would remove the ill effects of the internal covariate shift" [107]. By standardizing the activations of previous layers, batch normalization ensures there are not any large swings in the value of the inputs into the subsequent layers. This helps to speed up and stabilize the training of the network.

3.2.7 Activation Functions

Before describing other layers in a CNN, a couple of common activation functions should be discussed. In Fig. 3.7, some of the more common choices of activation functions are represented. The sigmoid function was previously shown in equation 3.2. As previously discussed, this activation function is used in predicting an output between zero and one. This function will be shown again below for the sake of comparison with the other activation functions. The tanh activation function is used to predict an output between negative one and one. The ReLU activation function accounts for some limitations of the sigmoid and tanh activation functions. The tanh function was favored over the sigmoid function during the late 1990s and early 2000s until the ReLU function gained in popularity. The function has been noted as "[another] major algorithmic change that has greatly improved the performance of feedforward networks"[106]. The simpler ReLU function allows deep networks to train several times faster than networks trained with tanh and sigmoid functions. The sigmoid, tanh, and ReLU activation functions are respectively listed below in equations 3.9, 3.10, 3.10 and 3.11:

$$\sigma(z) = \frac{1}{1 + e^{\ z}} \tag{3.9}$$

$$tanh(z) := \frac{e^z - e^{\ z}}{e^z + e^{\ z}} \tag{3.10}$$

$$ReLU(z) = max(0, z) = \begin{cases} 0, & \text{if } z < 0 \\ 1, & \text{if } z >= 0 \end{cases} \tag{3.11}$$

ReLU activation functions are presently the default activation unit used throughout most networks. That is generally true except on the last layer of a network where sigmoid, tanh, and softmax functions are used. The softmax activation function is useful for multi-class problems and can replace sigmoid and tanh activation functions in such scenarios. A

Figure 3.7: Common activation functions. [104]

representation showing such a scenario on a small MLP network is shown in Fig. 3.8. The softmax function is shown below in equation 3.12:

$$\sigma(\overrightarrow{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{(z_j)}} \tag{3.12}$$

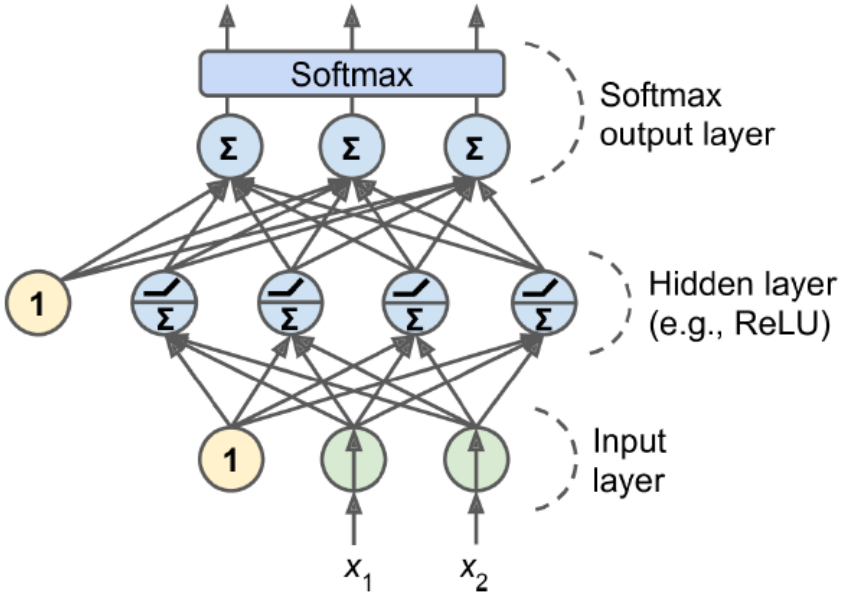


Figure 3.8: MLP with ReLU activations in hidden layer and softmax layer. [105]

3.3 Construction of Convolutional Neural Networks

Having described many of the layers common to neural networks, it is now time to move on to discussing several layers that are specific to CNNs. CNNs typically have two main

portions: A feature extractor portion and a classification portion. An illustration depicting this division is shown in Fig. 3.9. These two portions can be treated in isolation. Let us cover the feature extraction portion first. There are two main layers used in the feature extraction portion of a CNN, although there is considerable architectural diversity that can be added. These two layers are convolutional layers and pooling layers.

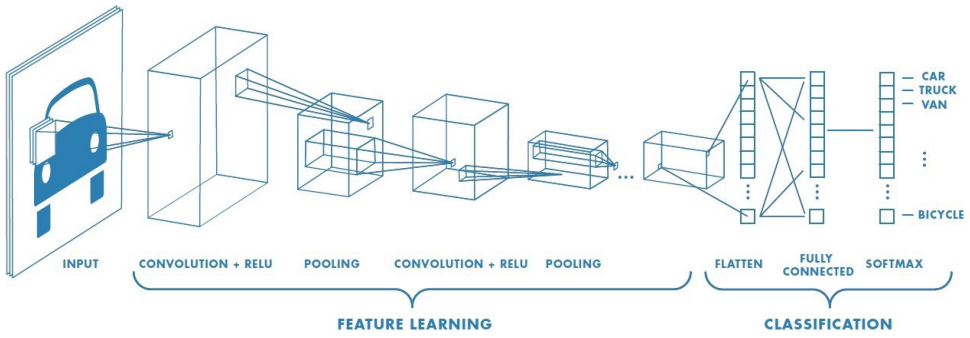


Figure 3.9: A CNN with its feature extraction and classification portions. [108]

3.3.1 Feature Extraction - Convolutional Layers

The purpose of a convolutional layer is to extract the high-level features of complex input images. Convolutional layers help to reduce input image matrices into forms that are easier to process. They do this while attempting to retain features that are relevant to classifying an image. A convolutional layer takes a 2D image and runs a kernel overtop of it. This kernel traverses the entire surface area of the image and produces a feature map. The depth of the output of a convolutional layer depends on its number of kernels/filters (k). If there are three kernels, for example, the output will have three corresponding feature maps. To calculate the final feature maps output by a convolutional layer, a dot product of the original image matrix and the kernels is calculated as is illustrated in Fig. 3.10.

The stride (s) of a convolutional layer is the number of pixels that a kernel moves over top of while traversing the input image matrix. A higher stride setting has the effect of reducing the height and width of a convolutional layer's final feature maps. Padding (p) is

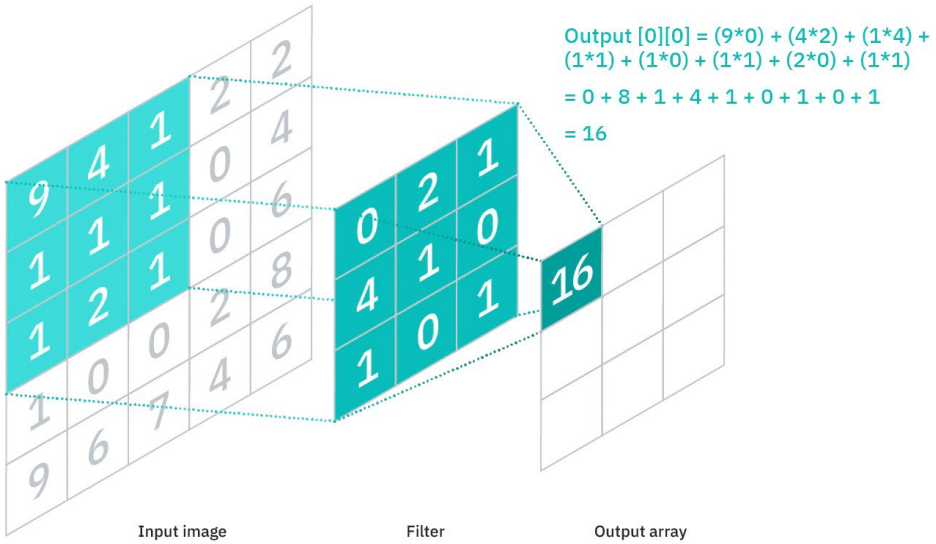


Figure 3.10: A kernel traversing a convolutional layer with dot product calculation shown. [109]

occasionally used in CNNs to account for instances where a convolutional layer's kernels do not spatially fit the original input image. There are two main kinds of padding available on deep learning platforms: valid padding and same padding. Valid padding is the no padding option. Same padding forces a convolutional layer's output feature maps to be the same size as its input image if there is a stride of one. The kernel window using this option spills over the edges of the input image while traversing it. An example showing this is illustrated in Fig. 3.11.

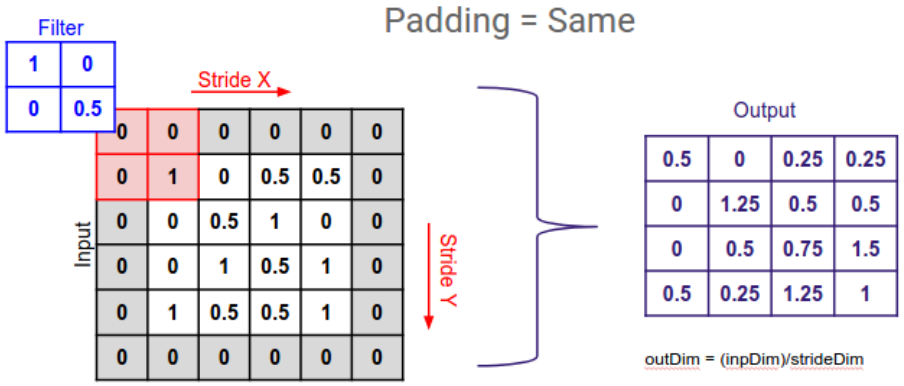


Figure 3.11: A kernel traversing a convolutional layer with 'same' padding. [110]

3.3.2 Feature Extraction - Pooling Layers

In CNNs, pooling layers are almost always used in combination with convolutional layers to conduct dimensionality reduction. Pooling also uses a filter operation that sweeps across an entire input matrix. Unlike with convolutional layers, however, this filter operation does not have any weights associated with it. Instead, this filter uses an aggregation function across a specified receptive field. There are two main kinds of pooling: max pooling and average pooling. Max pooling moves a filter across an input matrix, selects the pixel with the highest intensity, and forwards it to the output matrix. Average pooling, on the other hand, takes the average value of the pixels across the field that is encompassed by the filter and outputs that value to the output matrix. Both forms of pooling are illustrated in Fig. 3.12. While some information in pooling is lost, it has significant advantages. Max pooling is generally more popular as it helps to extract the sharpest features in an image. It also helps to reduce complexity and protects against overfitting. Importantly, pooling helps to add translational invariance into a CNN. This means that small horizontal or vertical translations in an input image do not have as severe of an effect on a CNN's final interpretation.

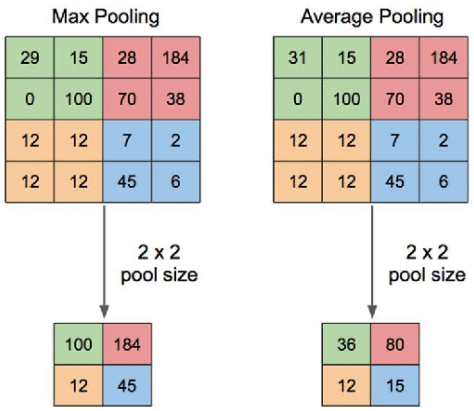


Figure 3.12: max pooling and average pooling. [111]

3.3.3 CNN Classification Layers

The feature extraction portion of a CNN helps to reduce the complexity of the features that eventually need to be classified by the classification portion of a CNN. While convolutional layers and pooling layers are the most common parts of a feature extractor, before we move on, it should be mentioned that batchnormalization layers are commonly found in the feature extraction portions of CNNs as well. After a 2D image passes through the feature extraction portion of a CNN, the classification layers of a CNN work to classify the processed image. The classification portion of a CNN takes 2D reduced images from a CNN's feature extractor and flattens them into a 1D fully connected MLP network. This MLP network takes the high-level features represented at the output of the feature extractor and classifies an image. The classification portion of a CNN is typically composed of a flatten layer, several fully connected layers, occasionally a dropout layer, and a softmax/sigmoid classification layer. For binary classification, a sigmoid activation function is generally used in the final layer. For multiclass classification, a softmax activation is almost always used instead. In more recent implementations of CNNs, it is also common to see global max pooling layers or global average pooling layers used as alternatives to flatten layers.

3.4 Segmentation Networks

Segmentation is often used in computer vision problems to help improve the performance of a classifier. A deep learning classifier like a CNN is excellent at classifying the patterns of original images that are presented to it. Sometimes, however, it is better to restrict the original pixel information that is input into a CNN. Removing unnecessary imaging details can assist a CNN to focus on more relevant portions of an image. CNNs, unfortunately, often get deceived by non-relevant segments of an image. It is not uncommon for a CNN to obtain the correct classification for the wrong reasons. A CNN, for instance, may focus on the text in an image that is related to the object being classified in the picture. Rather than focusing on the actual object, a CNN might focus on text because it occurs more often in one of the categories being classified. Quite often, if a CNN has had issues

during training, it is looking at unexpected parts of an image like the background. This can be discovered by creating and analyzing the saliency maps of images classified by the CNN.

There are two main kinds of image segmentation: semantic segmentation and instance segmentation. Semantic segmentation gives each pixel in an image a class label. This process causes objects to be grouped into defined categories with predetermined colors. On the other hand, instance segmentation assigns a new color to every object in the same group. Fig. 3.13 shows an example where a ground truth image of two dogs is processed with both semantic segmentation and instance segmentation. The instance segmentation in Fig. 3.13 causes both dogs to receive a different color. Using semantic segmentation, however, both dogs receive the same color and the background is removed. Both forms of segmentation are used in medical imaging, but only an understanding of semantic segmentation is necessary for the work presented in later chapters.

Figure 3.13: Ground truth Vs semantic segmentation Vs. instance segmentation. [112]

3.4.1 U-Net Segmentation Layers

The medical imaging industry in recent years has made wide use of the U-Net [46] architecture in performing semantic segmentation. The basic architecture of the U-Net was originally designed by Ronneberger et al. [46] in 2015. The U-Net [46] is designed with many of the deep learning and CNN layers already highlighted in this chapter, with one notable exception. The U-Net [46] has also been designed to operate with up-convolution layers, which we will discuss shortly. Let us take a look at the overall original structure of Ronneberger et al.'s [46] U-Net shown in Fig. 3.14 and discuss the design methodologies employed in its architecture.

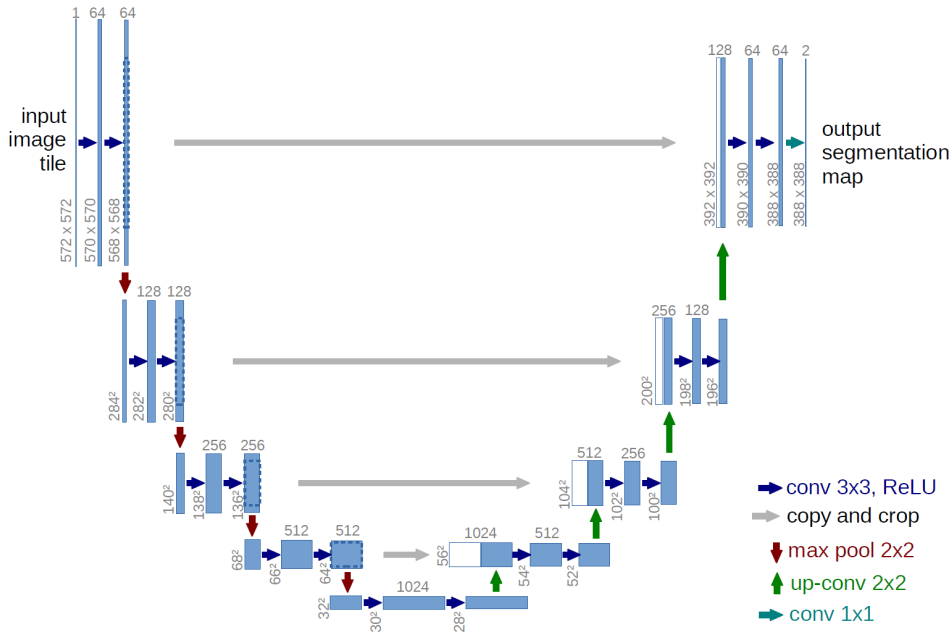
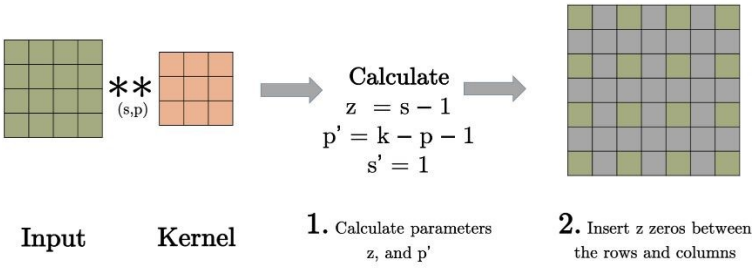


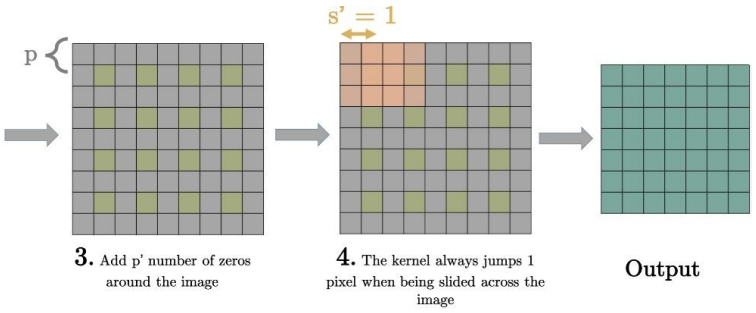
Figure 3.14: U-Net architecture. [46]

On initial inspection, an observer might first notice that a U-Net's [46] output image is the same size as its input. This is due to every pixel in the original image getting classified as belonging to a certain class. The architecture is shaped like a U and has a very nice symmetric design. The first half of a U-Net architecture [46] is the 'encoder' or 'contraction' path. The encoder path processes an image using two convolutional layers. These are each followed by a ReLU activation function and batchnormalization layer. Following each set of these layers along the 'encoder' path, downsampling using max pooling occurs at progressive intervals. Eventually, the spatial dimensions of the images become reduced in size. These image matrices, thereafter, get pushed through the bottom middle part of the U-Net called the 'bottleneck.' The bottleneck is composed of another two convolutional layers that can further extract features. The bottleneck, unlike the encoder, however, does not contain any form of max pooling.

After the bottleneck layer, the 'decoder' or 'expansive' path eventually upsizes a condensed image back up to its original size. Directly after the bottleneck, an upsampling layer is used to move the condensed image back up to the fourth level of the U-shape. This upsampling layer is typically a transposed convolutional layer. A transposed convolutional



(a) Inital Steps



(b) Final Steps

Figure 3.15: Transposed convolution. [113]

layer in effect performs the reverse operation of a convolutional layer. As shown in Fig. 3.15, a transposed convolutional layer inserts zeros (z) between all the rows and columns of the input image matrix. It also adds a layer of padding (p) around the image before employing kernels that move along the image matrix with a predetermined stride (s). The result is an upsampled output. This upsampling procedure moves the image matrix up to the fourth layer of the U-Net [46]. The filters in this portion of the encoder get concatenated with a set of filters from the decoder via a skip connection. This leaves us with a total of 1024 filters on the fourth layer. The concatenated set of 1024 filters gets passed through two convolutional layers and another transposed convolutional layer. This pattern continues until the end of the decoder. At the final stage, an output segmentation map is created after performing a 1 x 1 convolution.

Chapter 4

COV-SNET: A Deep Learning Model for X-Ray-Based COVID-19 Classification

4.1 Introduction

Our first research study was focused on the development of a new deep learning model that was trained to classify patients suspected of suffering from COVID-19. Our objective was to obtain the highest COVID-19 sensitivity possible in order to ensure COVID-19 patients receive a positive diagnosis. The contributions of our work are three-fold:

1. The proposed COV-SNET models we present are capable of diagnosing COVID-19 with accuracies above those reported by practicing radiologists in a related work [19]
2. The dataset we use does not incorporate several sources of bias contained in related works
3. Our work presents a comprehensive study that benchmarks our new COV-SNET models with other existing COVID-19 deep learning models

Our work commences in section 2 with a discussion of other studies that have used transfer learning for diagnosing COVID-19. In section 3, we then move on to discuss our proposed network architecture and the deep learning methods we have employed for processing the X-ray scans of COVID-19 patients. After explaining these methods, in section 4 we present the experimental results of our system. We thereafter compare the performance of

our models with other existing systems and discuss the advantages of our approach. Lastly, in section 5 we conclude our discussion with possible future directions for this research.

4.2 Related Works

There are a number of papers that have been published on using deep learning methods on X-ray images for diagnosing COVID-19. There is a variety of approaches that have been researched on the subject and a large assortment of public COVID-19 X-ray datasets in circulation. Below are some of the findings of the most important papers that have been published on the subject.

The designers of COVIDX-Net [13] compared seven 2D off-the-shelf architectures. Hemdan et al. [13] intended to compare these architectures using the same training and test methods. Apostolopoulos and Mpesiana [15] took the same approach as Hemdan et al. [13] and compared several architectures that were pretrained on ImageNet weights. Hemdan et al. [13] reported the best architecture's results came from using the VGG-19 [30] and DenseNet-201 architectures [31]. Apostolopoulos and Mpesiana [15]'s approach differed from Hemdan et al. [13] in that they reported 2-class and 3-class (COVID vs. pneumonia vs. normal) results. They found a VGG-19 obtained the highest results. There were a couple of major deficiencies in these reported studies. These studies' datasets (especially Hemdan et al. [13]) were both too small to achieve trustworthy results. They also only used ImageNet and neglected using a form of modality-specific transfer learning. Apostolopoulos and Mpesiana [15] made the mistake of using Kermany et al.'s [16] pneumonia dataset of children between the ages of one to five years old. We noticed that papers that have used this dataset tend to report unrealistic evaluation metrics.

Khalifa et al. [48] first proposed using a generative adversarial network (GAN) [35] to further augment the images input into their classifier and increase its accuracy in diagnosing patients with pneumonia. The authors increased the size of their dataset by a factor of ten. They believe this helped their classifier to avoid overfitting. They attempted to use several deep learning classifiers in their model and ultimately decided to use a ResNet-18 [29].

Waheed et al. [49] also designed their model incorporating a GAN and later released a work similar to Khalifa et al. [48]. Their model differed in that they used an auxiliary classifier generative adversarial network (AC-GAN) [114]. Their AC-GAN generated synthetic images that were input into a VGG-16 classifier [30]. Khalifa et al. [48] made the mistake of using Kermany et al.'s [16] pneumonia dataset. Waheed et al. [49] look to have made the same mistake by using the COVID-19 Radiography Database [115].

Wang et al. [51] designed "COVID-Net" for the purpose of diagnosing COVID-19. The dataset used to train this custom-designed CNN was made public and eventually used in several other research papers. This dataset is one of the largest datasets publicly available and the dataset does not contain many of the errors found in several other public datasets. Their model demonstrated promising results and achieved an accuracy of 93.3 percent. Their model was constructed using a "machine-driven design exploration strategy" [51] that uses generative syntheses [52]. This particular strategy was the subject of some of the authors' previous research prior to the COVID-19 pandemic. Their approach is capable of generating efficient deep neural networks automatically and designs these networks using a ResNet architecture [29]. The authors of this paper also used an explainability method called GSInquire [53] to validate their work.

Rajaraman et al. [14] created a model of iteratively pruned deep learning ensembles to diagnose COVID-19. The authors carried out their work by first training several popular CNN models (VGG-16/VGG-19 [30], Inception-V3 [34], Xception [55], DenseNet-201 [31], etc.) on a separate lung X-ray task (a modality-specific task). To use fewer model parameters and help improve the model's accuracy, the authors iteratively pruned their CNNs. They combined these iteratively pruned CNNs using several ensemble strategies. They found weighted averaging to be the most effective ensemble strategy. Like many other studies, they made the mistake of using Kermany et al.'s [16] pneumonia dataset.

Another study that deserves consideration is Wehbe et al.'s [19] publication that attempted to diagnose COVID-19 using a large private dataset from a US medical institution. This paper was similar to Rajaraman et al.'s paper [14] as the authors constructed an ensemble of many CNNs to detect COVID-19. Their dataset, however, didn't suffer from the

same deficiencies in size as other datasets. They also did not use Kermany et al.'s [16] dataset. The paper is noteworthy in that the authors assembled a team of five radiologists to determine the diagnosis of COVID-19 patients. They thereafter compared the predictions of the radiologists with their ensemble model. They found that the consensus of five radiologists was only able to detect COVID-19 with 81 percent accuracy. These results give a reasonable estimate of Bayes error for the task of determining the diagnosis of suspected COVID-19 patients. The author's ensemble model produced predictions with 82 percent accuracy, which is reasonable given the experts' consensus accuracy of 81 percent. Previous studies were unable to perform comparisons of their models against the predictions of working radiologists. The evaluation metrics mentioned in many of the previous papers were also liable to be skewed by the size of their datasets. Smaller datasets can sometimes lead to overly promising results.

Yeh et al. [20] used private datasets from several medical institutions and added them to Wang et al.'s dataset [51] when training their DenseNet-121 model [31]. They trained and tested their deep learning model initially using images from the same sources as Wang's COVIDx Dataset. They also used pneumonia, COVID-19, and normal X-ray images from two medical institutions. They obtained very promising results and achieved COVID-19 sensitivities between 95-100 percent. They held out a third much larger private dataset from a medical institution to see how their results would change with extra data. This larger dataset caused their accuracy to drop and they achieved an 81.82 percent COVID-19 sensitivity on their test set. This is evidence that using a small COVID-19 X-ray dataset leads to unrealistic evaluation metrics. The third private dataset only included 306 extra COVID-19 patients, but these added images caused a drastic change to the results of their deep learning model.

Mangal et al. [44] have created a computer-aided detection (CAD) system for diagnosing COVID-19 based on a ChexNet model [45]. ChexNet first gained the attention of the research community because of its ability to diagnose 14 pulmonary pathologies. The model is designed using a DenseNet-121 architecture [31] and has been trained on over 100,000 X-rays. They created 3-class and 4-class models. Mangal et al. [44] validated their model using

Gradient-weighted Class Activation Mappings (Grad-CAMs) [39]. A deficiency in this model was that it used a dataset from Kermany et al. [16] when making use of Paul Mooney's Chest X-ray dataset on Kaggle [116]. The dimensions of the lungs in these X-rays that were taken from children likely caused their final classifier to produce unpredictable results. Haghanifar et al. [43] improved on Mangal et al.'s [44] original design by including a segmentation unit with their ChexNet model. They constructed a different dataset than Mangal et al. [44] for training their ChexNet model. Hagnifar et al. [43] made the same mistake as Mangal et al. [44] in including Kermany et al.'s [16] dataset. Al-Waisy et al. [47] likewise published a paper using a ChexNet model that made the same mistake. The authors obtained an even more exaggerated set of performance metrics than the previous two models mentioned. Unfortunately, the use of Kermany et al.'s [16] dataset is widespread and this has created a major flaw in all of these ChexNet models.

Islam et al. [117] developed a novel CNN-LSTM model for diagnosing COVID-19 with chest X-rays. Their model was unique in terms of its architecture in the literature. During validation, they obtained accuracies, specificities, sensitivities, and F1-scores between 98-100 percent for all classes in their results. Their model seemed to report what looked like overly optimistic performance metrics. This suspicion was confirmed when it was noticed that their model reported using Kermany et al.'s dataset [16] (also referred to as the Kaggle chest X-ray repository in their article).

Rahimzadeh et al. [95] developed a deep learning model that combined the Xception [55] and ResNet-50 [29] models together. Two '$10 \times 10 \times 2048$ feature maps' [95] forming the last feature extractor layers of both models were concatenated to improve on the final results of each classifier. This novel architecture worked quite well and the authors additionally performed five-fold cross-validation to improve the robustness of their results. Overall the authors of this article achieved reasonable success with their model as they achieved an overall accuracy of 91.4 percent and sensitivity of 80.5 percent.

Panwar [118] et al. constructed and optimized a VGG-19 model with ImageNet weights to detect COVID-19 in suspected patients. Their model was trained both on x-ray and CT scans. Their models were all binary models and these models compared COVID-19 patients

vs. normal patients, COVID-19 vs. pneumonia patients, and COVID-19 patients vs. non-COVID-19 patients. The authors also focused on generating Grad-CAM heatmaps to make sure they were picking up the features of COVID-19 in X-rays and CT scans. While their CT classifier's dataset is likely adequate, their dataset for comparing COVID-19 vs pneumonia patients had a source of bias as their X-ray pneumonia images were derived from Kermany et al.'s [16] dataset.

Afshar et al. [119] published a paper that utilized a unique deep learning approach to diagnosing COVID-19. While the vast majority of models in the literature use CNNs to detect COVID-19, Afshar et al.'s [119] model utilized Capsule Networks (CapsNets). CapsNets are alternative models that can better utilize the spatial information in images by using "routing by agreement" [119]. The capsules in these networks are thereby capable of reaching "a mutual agreement on the existence of the objects" [119] in an X-ray. Like previous teams mentioned before, the authors pretrained their COVID-CAPS model on 94,323 X-rays before fine-tuning the model to a smaller COVID-19 dataset. A deficiency we found in this work is that the authors included Kermany et al.'s dataset [16] when making use the Paul Mooney's Chest X-ray dataset [116] on Kaggle.

Karthik et al. [120] presented a unique CNN in their work, which used a Channel-Shuffled Dual-Branched (CSDB) CNN that is augmented with Distinctive Filter Learning (DFL). This unique architecture learns "custom filters within a single convolutional layer for identifying specific pneumonia classes." [120] They compared their model with a variety of standard CNNs and promisingly outperformed those CNNs after training them on the same dataset. Their dataset, unfortunately, contained a deficiency whereby the authors used bacterial pneumonia and pneumonia X-rays derived from Kermany et al.'s dataset [16].

4.3 Proposed Network Architecture

4.3.1 Dataset

An important aspect of developing a deep learning model in medical imaging begins with the data. The availability of X-ray images and metadata is important when considering the research directions for such a project. In our data-gathering stage, we found it difficult to find metadata accompanying X-ray images. There was an insufficient amount of metadata to assist with developing a practical COVID-19 diagnosis system. There were many publicly available datasets available, but in analyzing these datasets we found that many of them were incorrectly assembled. Many datasets on Kaggle and in various research papers used Kermany et al.'s [16] dataset. As previously mentioned, this dataset consists of chest X-rays from children between the ages of one and five years old. A child's lungs have different features than an adult's lungs and hence these datasets were taken out of consideration. We also found that the vast majority of publicly available datasets made no mention as to whether they divided their training and test sets by patient number. Most datasets incorporated COVID-19 X-rays harvested from medical research papers. In many of these datasets, multiple images from the same patient could be found. Wang et al.'s [51] 'COVIDx' dataset does not suffer from the same disadvantages. Wang et al. [51] split their training and test sets by patient number. Their COVIDx dataset is large in comparison with other datasets and is "comprised of a total of 13,975 CXR images across 13,870 patient cases" [51]. This dataset contains 358 COVID-19 images, 8066 normal images, and 5541 pneumonia images. The COVIDx dataset has been used by many other research teams and is currently a good benchmark for testing a new model's results with other papers. For these reasons, we decided to use the COVIDx dataset in our study.

We divided the COVIDx dataset into a 90 percent training set and 10 percent test set ratio. This allowed for a suitable number of COVID-19 examples in the training set given the extreme class imbalance in the COVIDx dataset. The multi-class training set, therefore, consisted of 258 COVID-19 patients, 7966 normal patients, and 5441 pneumonia patients. Ten percent of the dataset was leftover for validation, but within the test set, there

was again a class imbalance. We, therefore, reduced the number of normal and pneumonia examples in the test set to match the number of COVID-19 examples. In doing so, we obtained a balanced test set for evaluating our model's performance. This three-class test set ultimately consisted of 100 COVID-19 examples, 100 normal examples, and 100 pneumonia examples. A binary classifier was also designed in this study which grouped pneumonia and normal images into a single category. Our two-class COVID-19 vs. non-COVID-19 X-ray classifier was constructed to compare our approach with other two-class studies. Our binary training set consisted therefore of 258 COVID-19 images and 13407 non-COVID-19 images. The binary classifier's test set consisted of 100 COVID-19 X-rays and 100 non-COVID-19 X-rays.

We first trained and tested our deep learning model on the aforementioned datasets but later went on to create another set of larger training sets. Given the small number of COVID-19 images available in the COVIDx dataset, we expanded the number of COVID-19 images in this dataset to examine possible overfitting. Previous studies [20, 19] mention this specifically as a reason for reduced COVID-19 sensitivity in their work. We wanted to investigate if more COVID-19 images would create a significant correction to our classifier's COVID-19 sensitivity. This second training set we created started out with 517 COVID-19 images from the COVIDx5 [51] training set. This second training set also included 922 images from the MIDRC-RICORD-1C database [121] and 2474 images from the BIMCV dataset [122]. Our second training set, therefore, consisted of 3913 COVID-19 images, 7966 normal images, and 5441 pneumonia images. For binary classification, we also examined how well our model works with a training set of 3913 COVID-19 images and 13417 non-COVID-19 images. We kept the original test sets as a benchmark to test our system against our previously trained classifiers and Wang et al.'s published model [51]. Tables 4.1 - 4.2 shows the COVIDx training set dataset alongside our expanded training set as well as our shared test set.

Table 4.1: Datasets - Number of Images in the Multiclass Training and Test Sets

	COVID-19	**Normal**	**Pneumonia**
COVIDx Multiclass Training Set	258	7966	5451
Our Expanded Multiclass Training Set	3913	7966	5451
Multiclass COVIDx Test Set	100	100	100

Table 4.2: Datasets - Number of Images in the Binary Training and Test Sets

	COVID-19	**Non-COVID-19**
COVIDx Binary Training Set	258	13417
Our Expanded Binary Training Set	3913	13417
Binary COVIDx Test Set	100	100

4.3.2 System Design

Both models in our study are designed with a DenseNet-121 [31] base feature extractor and trained on the ChestX-ray14 dataset [123]. The ChestX-ray14 dataset contains "112,120 frontal-view X-ray images of 30805 patients" [45]. This form of modality-specific transfer learning increases our model's ability to capture COVID-19 features. The DenseNet-121's earliest layers contain feature maps that have already been trained to pick up many of the tissues and patterns seen in chest X-ray images. Many architectural design options were investigated before finalizing a new architecture model based on a DenseNet-121 network. The proposed system architecture, COV-SNET network, has the following features. After loading our pretrained weights into the DenseNet-121 network we have added a dense layer with 128 units, a dropout layer with a dropout rate of 10%, and a 3-class softmax layer for multiclass classification. An illustration of our model can be observed in Fig. 4.1. For our binary classifier, we replaced the softmax layer with a dense layer containing a single sigmoid activation function. Table 4.3 shows a detailed layer by layer description of our model.

Prior to training our models, we noticed that a class imbalance existed that required correction. This mainly was due to the lack of COVID-19 X-rays publicly available. A weighted loss function was used during training to correct for this class imbalance. The class_weights parameter in Kera's model.fit method was used to balance our classes. This

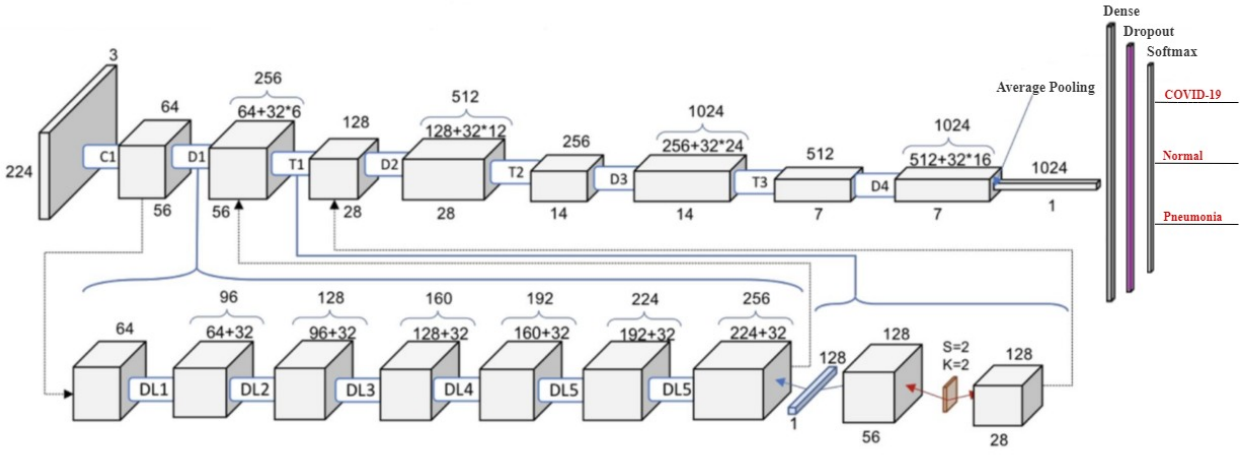


Figure 4.1: Proposed network architecture.

Table 4.3: Proposed Network Architecture for COVID-19 Classification

Layers	Output Size	Model
Convolution	112x112	7x7 conv, stride 2
Pooling	56x56	3x3 max pool, stride 2
Dense Block (1)	56x56	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} x6$
Transition Layer (1)	56x56 28x28	1x1 conv 2x2 average pool, stride 2
Dense Block (2)	28x28	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} x12$
Transition Layer (2)	28x28 14x14	1x1 conv 2x2 average pool, stride 2
Dense Block (3)	14x14	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} x24$
Transition Layer (3)	14x14 7x7	1x1 conv 2x2 average pool, stride 2
Dense Block (4)	7x7	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} x16$
Average Pooling	1x1	7x7 global average pool
DNN	-	128 units, relu
Dropout	-	10 percent
Classification	-	3 category softmax

function took a while to find. At first, we attempted to use a custom weighted loss function. We found, however, that this algorithm took too long to compute on successive iterations and we thereafter used Kera's inbuilt function. An equation describing the function cannot currently be found Kera's documentation. For finding weights, early in our work, we attempted to use the inverse of the number of samples in each category. We did so to find a reasonable ratio between the classes. When setting our weights, we ensured that the minority class was set to be a higher weight to compensate for its underrepresentation. We later found that the performance of our COVID-19 class needed improvement, so we later finetuned its weight to be more highly represented. In doing so, we achieved a higher COVID-19 sensitivity. In addition to correcting for the class imbalance, we also used data augmentation methods during training to increase our model's capacity to generalize on new examples. All final models used image rotations, vertical/horizontal translations, horizontal flips, shearing, and random zooms to augment the training datasets. Each category of augmentation was set to 15 percent for the multiclass models and 20 percent for the binary models. In addition to correcting for the class imbalance, our training required some necessary preprocessing steps. We used data augmentation methods during training to increase our model's capacity to generalize on new examples. For our multiclass models, we set image rotations to 15%, vertical/horizontal translations to 15%, image shearing to 15%, and random zooms to 15% when augmenting our training dataset. For our binary models, each of the aforementioned augmentation categories was set to 20%. In all of our models, we additionally used horizontal flips in our augmentation process. During training and testing, our batch size was set to 32. Using Kera's ImageDataGenerator class, we additionally normalized our training data so that the values in each batch had a mean of 0 and a standard deviation of 1.

The first step in training our COV-SNET models involved initially training the final layer alone. The last layer of each network was trained in TensorFlow 2.0 for 9 epochs. The Adam optimizer was used during this training. To increase the performance of our networks we unfroze all of the layers in our models for further training. For 6 epochs we left the Adam optimizer at its default learning rate. After 6 epochs we fixed the learning rate to $1x10^{-5}$ and trained each model until their peak sensitivities were reached. For the models trained on the COVIDx dataset alone this required 10 epochs. For the models trained on our larger

training set, this took 13-14 epochs. Before unfreezing the layers in our model, we fixed the moving mean and moving variance of the batches in our model's batchnormalization layers. These batchnormalization parameters were fixed to the weights generated from training our model on the ChestX-ray14 dataset.

4.4 Experimental Results

4.4.1 Performance Evaluation

The results reported in the COVID-19 deep learning literature are typically based on a variety of evaluation metrics. Accuracy, specificity, sensitivity, precision, recall, negative predictive value (NPV), positive predictive value (PPV), F1-Score, and area under the ROC curve (AUC) are all evaluation metrics used in the literature and included in our final results.

After training the last layer of each model for 9 epochs, the overall validation accuracy for each model was between 75 to 80 percent. While this was close to the performance of practicing radiologists in a previous study [19], we knew this result could be further improved upon by unfreezing layers in each model. After the models were unfrozen, all of the models achieved COVID-19 sensitivities of at least 95 percent. The entire set of class-wise performance statistics that were calculated for each classifier can be seen in Tables 4.4 - 4.7. Their corresponding confusion matrices can also be seen in Figs. 5.4 - 5.7. Our three-class model trained on the original COVIDx training set ultimately hit a final validation accuracy of 84.3 percent. Our 3-class model trained on our expanded training set obtained a validation accuracy of 86 percent. The final accuracy of the two-class model trained on the original COVIDx training set was 88.5 percent. The two-class model trained on our expanded training set obtained a validation accuracy of 87.5 percent. The AUC curves of all four of our models generated comparable results as can be seen in Figs. 5.8 - 4.7.

Table 4.4: Three-Class Model Performance Metrics After Training on the COVIDx Multiclass Training Set

	TP	TN	FP	FN	Acc.	Sens.	Spec.	PPV	NPV	F1
COVID-19	95	166	34	5	0.870	0.95	0.830	0.736	0.971	0.84
Normal	86	192	8	14	0.926	0.86	0.960	0.915	0.977	0.88
Pneumonia	72	195	5	28	0.890	0.72	0.975	0.935	0.874	0.82

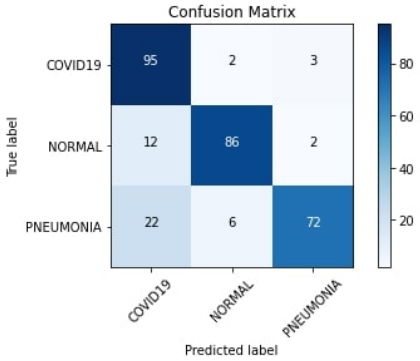


Figure 4.2: Confusion matrix generated by three-class model with COVIDx training set.

Table 4.5: Two-Class Model Performance Metrics After Training on the COVIDx Binary Training Set

	TP	TN	FP	FN	Acc.	Sens.	Spec.	PPV	NPV	F1
COVID-19	96	81	19	4	0.885	0.96	0.81	0.835	0.959	0.89
Non-COVID-19	81	96	4	19	0.885	0.81	0.96	0.953	0.835	0.876

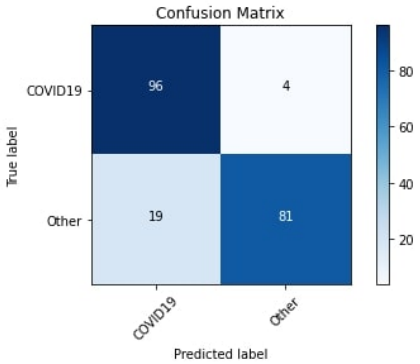


Figure 4.3: Confusion matrix generated by two-class model with COVIDx training set.

Table 4.6: Three-Class Model Performance Metrics After Training on Our Expanded Multiclass Training Set

	TP	TN	FP	FN	Acc.	Sens.	Spec.	PPV	NPV	F1
COVID-19	95	170	30	5	0.833	0.95	0.850	0.760	0.971	0.86
Normal	93	189	11	7	0.940	0.93	0.945	0.894	0.964	0.91
Pneumonia	70	199	1	30	0.897	0.70	0.995	0.989	0.869	0.82

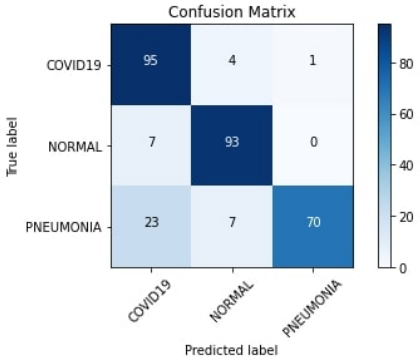


Figure 4.4: Confusion matrix generated by three-class model with expanded training set.

Table 4.7: Two-Class Model Performance Metrics After Training on Our Expanded Binary Training Set

	TP	TN	FP	FN	Acc.	Sens.	Spec.	PPV	NPV	F1
COVID-19	95	80	20	5	0.875	0.95	0.80	0.826	0.941	0.883
Non-COVID-19	80	95	5	20	0.875	0.80	0.95	0.941	0.826	0.865

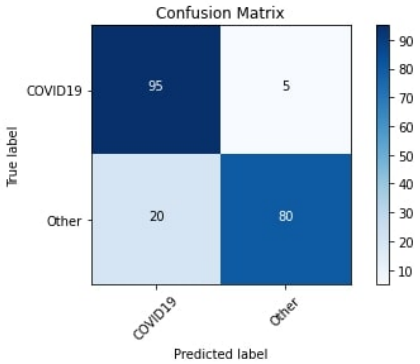
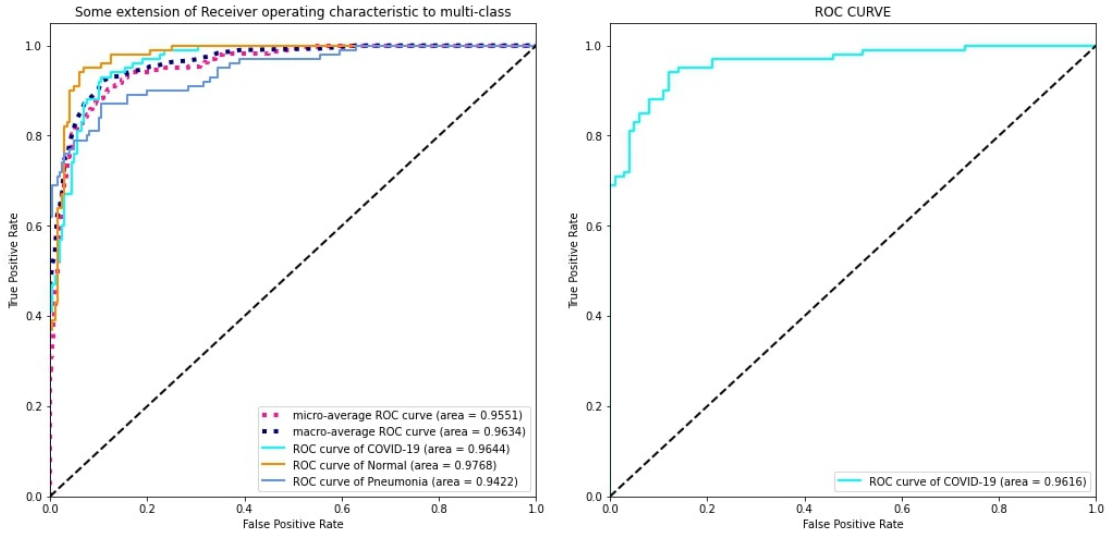
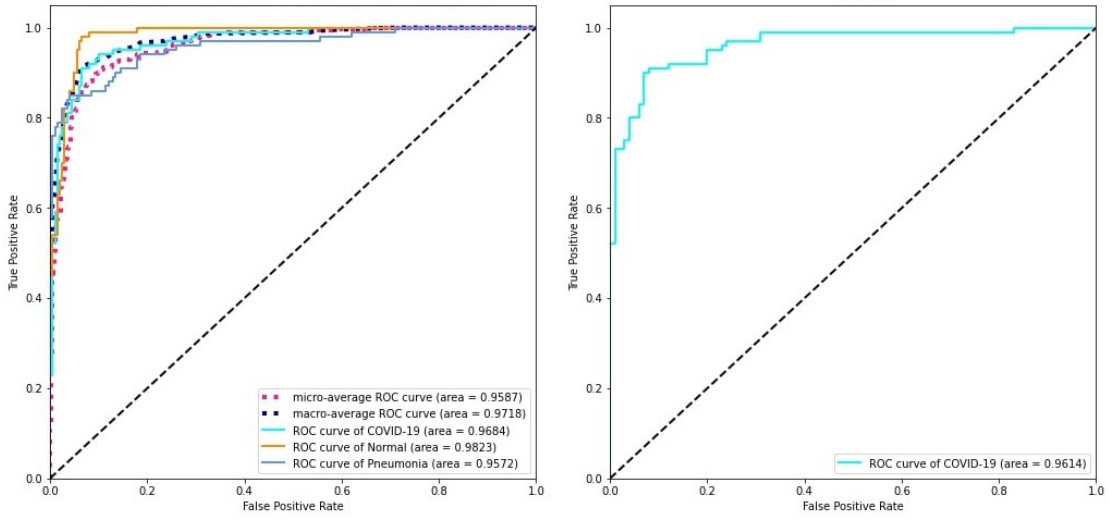


Figure 4.5: Confusion matrix generated by two-class model with expanded training set.



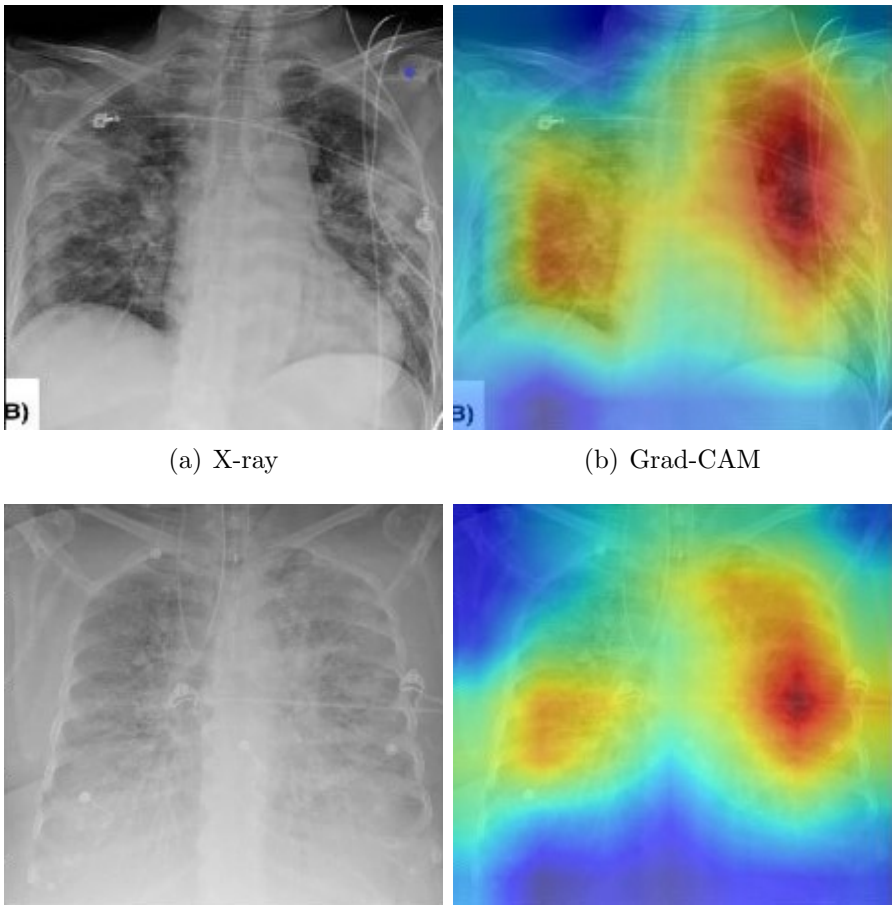
(a) (b)

Figure 4.6: ROC AUC graphs for COVIDx on (a) Three-class model and (b) Two-class model.



(a) (b)

Figure 4.7: ROC AUC graphs for Expanded Set on (a) Three-class model and (b) Two-class model.



(a) X-ray (b) Grad-CAM

(c) X-ray (d) Grad-CAM

Figure 4.8: Two different COVID-19 patients showing their original X-rays alongside their Grad-CAM produced heatmaps.

The evaluation metrics of a deep learning model should never alone be relied upon while validating the model's performance. Small datasets may only contain hundreds of images of the particular pathology under investigation. They tend to be prone to generating unrealistic evaluation metrics. To ensure a deep learning model is picking up correct features, saliency maps are widely employed in medical imaging. Saliency maps are important in that they can inform a designer whether a deep learning algorithm is being deceived by image characteristics that are unrelated to the pathology being imaged. Deep learning algorithms often incorrectly lock onto necklaces, medical devices, and text appearing in X-ray images. In our study, a Grad-CAM [39] was used to determine whether our COV-SNET model is fixing onto the correct features of COVID-19 in frontal chest X-rays. The heatmaps produced by a Grad-CAM contain color encoded information that highlights the features of an image that are the most relevant to a CNN's final classification. Fig. 5.9 shows the performance of our model on COVID-19 patients using Grad-CAM generated heatmaps. The red and orange regions of these Grad-CAM heatmaps are the most relevant parts of each image that contributed to a COVID-19 diagnosis in both patients. These colors transition into blue regions that are the least relevant portions of each image in contributing to our CNN's final classification. The Grad-CAM we employed uses the final feature maps in the last convolutional layers of our model to generate these regions of importance. As can be seen from our two examples, our Grad-CAM is locating the opacities in both images that would normally be picked by a radiologist when assessing these patients.

4.4.2 Discussion

All of our COV-SNET models achieved higher evaluation metrics than the consensus performance of the five radiologists in Wehbe et al.'s study [19] on a related dataset. While their dataset is not available publicly at this time, Wehbe et al.'s [19] study on the performance of five radiologists provides a good approximation for Bayes error. The best performing radiologist in Wehbe et al's [19] study only achieved an accuracy of 81 percent in diagnosing COVID-19 correctly. The best sensitivity among the radiologists was 76 percent. All of our models beat their best-performing radiologists by a substantial margin. Their

Table 4.8: Performance of Five Radiologists in Diagnosing COVID-19 with X-rays [19]

	Acc.	**Sens.**	**Spec.**
Consensus	81%	70%	89%
Best Radiologist	81%	76%	91%
Worst Radiologist	76%	60%	75%

work has been useful in that it provides designers with beneficial insights as to whether a deep learning model is providing reasonably grounded performance metrics. The consensus and best/worst performances of the five radiologists in Wehbe et al. [19] are provided in Table 4.8.

Many deep learning models in the literature report metrics that are superior to the performance of the radiologists in Wehbe et al.'s study [19]. Some papers report evaluation metrics that are superior to our own as well. What could be the reasons for this? Many papers have incorporated Kermany et al.'s [16] dataset. This dataset contains chest X-rays from children between the ages of one and five years old. The children in these chest X-rays are all suffering from various forms of bacterial and viral pneumonia. The extra categories in Kermany et al.'s [16] dataset were used as sources for comparison when diagnosing COVID-19 in other deep learning models. Many designers thought these extra categories would be useful in clinical situations for ruling out other possible sources of infection. It is incorrect however to train a deep learning algorithm with children's lungs if that same algorithm will ultimately be deployed on adult lungs. Apostolopoulos and Mpesiana [15], Khalifa et al. [48], Waheed et al. [49], Rajaraman et al. [14], Haghanifar et al. [43], Mangal el al. [44], Al-Waisy et al. [47], and Islam et al. [117] all used Kermany et al.'s [16] dataset in their models. Many of those models reported exceedingly high-performance metrics. To the best of our knowledge there is only one other deep learning model in the existing literature that uses a COVID-19 dataset as large as our own and at the same time does not make the mistake of using Kermany et al.'s [16] dataset. That model was published by Wehbe et al. [19] and they ultimately only achieved a COVID-19 sensitivity of 75 percent. There is still a need therefore to explore whether a deep learning model can achieve a higher COVID-19 sensitivity while using a larger training set than has commonly been available

Table 4.9: Performance of Past DenseNet-Based Models Versus Radiologists

Paper Reviewed	F1	ACC	COVID-19 Sens.
Yeh et al. [20]			
3-class	-	-	81.82%
Haghanifar et al. [43]			
2-class	94%	98.62%	-
3-class	85%	81.04%	-
Mangal et al. [44]			
3-class	92.3%	90.5%	100%
Al-Waisy et al. [47]			
2-class	99.99%	99.99%	99.98%
Rajaraman et al. [14]			
4-class	96.77%	96.83%	96.34%
Radiologists [19]			
2-class	-	81%	70%

Note: Haghanifar et al. [43], Mangal et al. [44], Al-Waisy et al. [47], and Rajaraman et al. [14] all improperly used Kermany et al's dataset [16].

to past authors. A correctly constructed dataset is required to perform this research. Prior to expanding Wang et al.'s [51] COVIDx dataset, we attempted to use public datasets that incorporated Kermany et al.'s dataset [16]. We trained a DenseNet-121, a DenseNet-201, and an Inception V3 architecture on these datasets. In doing so, we obtained suspiciously high-performance metrics and obtained accuracies between 98.0 and 99.6 percent on three-class and two-class models respectively. These performance metrics mirrored the performance metrics we have found in other studies that made the same mistake. Table 4.9 illustrates our point. It compares the performance of the radiologists in Wehbe et al.'s [19] study with other DenseNet-based models we have reviewed from the COVID-19 deep learning literature.

There are other possible reasons for the deep learning models in other studies to be generating unrealistic performance metrics. Many public datasets on Kaggle and various other platforms do not specifically state whether they have divided their training and test sets by patient number. If there has been cross-contamination between a deep learning

model's training and test sets, there is a high probability that the trained model will have a better knowledge of the features in the test set. This data leakage leads to unrealistic performance metrics. The X-ray files in public datasets are often renamed and their original source information in many instances is lost. Many papers have combined several public datasets. They often have done so without making any mention as to how they ensured the same images from different datasets were not duplicated in their own dataset. The datasets in some papers are also difficult to reconstruct and it is challenging to trace the chain of images that ended up being included in some datasets. These are all likely factors that are contributing to the high-performance metrics of some studies which are far outside of the performance range of practicing expert radiologists. We decided to use Wang et al.'s [51] 'COVIDx' dataset because the designers of that dataset took into account these issues being discussed. The dataset, therefore, is more conservative and grounded compared to other online public datasets.

It should now be clear that the composition of the datasets used to train deep learning COVID-19 models is one of the main contributing factors to the high evaluation metrics often being reported in the literature. There is however another crucial factor that is contributing to these unrealistic evaluation metrics. Many datasets in the COVID-19 X-ray imaging literature do not have a sufficient number of COVID-19 images. This lack of COVID-19 X-ray images in medical datasets can sometimes lead to unpredictable results. When more images are added there can be a correction in a system's evaluation metrics towards the performance reported by practicing experts in the field. This is precisely what happened in Yeh et al.'s [20] study. The work in [20] commenced with using an earlier version of the COVIDx dataset. The authors of the study also initially used the private X-ray images of two medical institutions. When the authors trained a DenseNet-121 classifier on these initial datasets alone they achieved a COVID-19 sensitivity of 96.8 percent. This did not last however and the inclusion of a third medical institution's COVID-19 X-rays in their model's training caused a correction in its evaluation metrics. This led their model to have a final COVID-19 sensitivity of 81.82 percent.

Yeh et al.'s [20] final dataset contained 510 COVID-19 images. The COVIDx dataset we used had 358 COVID-19 images. Our original three-class model, therefore, contained only 70 percent of the number COVID-19 images that Yeh et al.'s [20] model initially trained on. Our three-class model generated a COVID-19 sensitivity of 95 percent. Yeh-et al.'s [20] three class-model obtained a final COVID-19 sensitivity of 81.82 percent. Wang et. al.'s [51] three-class model used the same original dataset as ours and obtained a COVID-19 sensitivity of 91 percent. How do we know however that our 95 percent sensitivity would not correct if we trained on more COVID-19 images? After all, there are some in the research community [19] that have pointed out that overfitting is occurring in past models trained on small COVID-19 datasets. Recently a large number of COVID-19 images have become available that are independent of previous COVID-19 datasets. This led us to create an expanded dataset from the original COVIDx dataset that we used to check for overfitting. After further examination, we discovered that our evaluation metrics were not impacted by training our model on the expanded COVID-19 dataset. We were able to maintain the same COVID-19 sensitivity (95 percent) using this dataset on our three-class model.

We thereafter moved on to creating a two-class model with the same expanded dataset. Our original two-class model generated a COVID-19 sensitivity of 96 percent. After training this model on our expanded dataset we obtained a COVID-19 sensitivity of 95 percent. Wehbe et al.'s [19] two-class COVID-19 model obtained a COVID-19 sensitivity of 75 percent. Their ensemble model however was trained on a slightly larger dataset than ours. Their dataset contains 4253 COVID-19 images. They showed in their paper that their model's sensitivity (75 percent) was better than the consensus performance of the five radiologists in their study. They also argued that the high sensitivities of deep learning models presented in other studies were caused by a lack of COVID-19 images in publicly available datasets. We wrote earlier that this was indeed the case in Yeh et al.'s [20] study, but have been able to prove that it is not the case in our study. Expanding the COVIDx dataset did not significantly affect the performance of our classifier. Of all of the studies that do not improperly use Kermany et al.'s [16] dataset, our models achieve the highest sensitivities that we can find in the literature. Table 4.10 presents a comparison of the sensitivities among models that do not have any issues regarding dataset composition. Out of the papers in

Table 4.10: Performance of Papers Without Dataset Composition Issues

Research Paper	**COVID-19 Sens.**
Yeh et al. [20]	
3-class	81.82%
Wang et al. [51]	
3-class	91%
Wehbe et al. [19]	
2-class	75%
Rahimzadeh et al. [95]	
3-class	80.53%
Ours	
2-class	95%
3-class	95%

Note: These papers all do not include Kermany et al.'s dataset [16].

Table 4.10, we were able to only make a direct comparison of our work with Wang et al.'s [51] COVID-Net model. Our models ultimately required different augmentation settings than theirs in order to achieve optimal results. Unfortunately, we were unable to replicate the other datasets in Table 4.10. A couple of the papers in Table 4.10 mention that their datasets are private. Wehbe et al. [19] currently have the largest COVID-19 dataset that we have found in the literature, but unfortunately, it's entirely private. We have however been able to assemble a dataset that is now much closer in size to Wehbe et al.'s [19] private COVID-19 dataset. In doing so, we have been able to prove that deep learning models are capable of obtaining higher COVID-19 sensitivities than has previously been reported.

Chapter 5

A Deep Learning Segmentation-Classification Pipeline for X-Ray-Based COVID-19 Diagnosis

5.1 Introduction

Our second research study has been fully devoted to the task of constructing a segmentation-classification pipeline for diagnosing COVID-19. Many deep learning X-ray studies up until now have solely focused on classification in diagnosing COVID-19 in X-rays. While excellent research has occurred in this space, the number of articles dealing with COVID-19 X-ray segmentation has been quite limited. Segmentation is an important preprocessing technique that can shield a classifier from unnecessary pixel information when categorizing an image. Many authors from the studies published on various computer vision applications have found that proper segmentation increases the overall accuracy of a classifier [124, 125, 126]. It is vital, therefore, to explore the effect that segmentation has while training a COVID-19 classifier.

Our work begins in section 2 with an overview of various research studies that have constructed segmentation-classification deep learning pipelines to diagnose COVID-19. In section 3, we thereafter present our proposed deep learning pipeline's architecture, showing the internal details of our segmentation and classification modules. Following a discussion of our pipeline's architecture, in section 4 we present the experimental results of our overall system. In section 4, we additionally present a detailed comparative analysis of our pipeline

versus other well-constructed models in the literature. Concluding in section 5, we discuss potential future directions for this research.

5.2 Related Works

A large number of deep learning models have been designed that classify COVID-19 with and without segmentation. Here is a summary of some of the more important models that have been published in the field. We have expended considerable effort to include articles with a segmentation unit in order to see how our deep learning pipeline compares with other related studies. There are several public datasets available in circulation for segmenting chest X-rays that have been cited in the articles below. There are also a number of public and private datasets mentioned in these articles that were prepared specifically for COVID-19 classification. Combining a segmentation unit and classification model together is an especially challenging task. The following works below are all studies that influenced how we ultimately implemented our final system.

Rajaraman et al. [14] created a segmentation – classification deep learning pipeline to diagnose COVID-19 that included an ensemble of iteratively pruned CNNs. The authors trained several CNN models (VGG-16/VGG-19 [30], Inception-V3 [34], Xception [55], DenseNet-201 [31], etc.) after their dataset had been preprocessed by a U-Net [46] segmentation module that included a Gaussian dropout layer [127]. The authors of this paper tried to employ many different ensemble strategies and, in the end, found that weighted averaging produced the best results. The authors of this paper unfortunately listed Kermany et al.'s [16] dataset as being contained in their dataset which likely contributed to exaggerated evaluation metrics. It is incorrect to bias a dataset with only certain categories of the dataset having images of children's lungs.

Alom et al. [78] designed an X-ray-based system that diagnoses COVID-19 with a NABLA-N segmentation network [128] and an Inception Residual Recurrent Convolutional Neural Network (IRRCNN). Their X-ray model is initially trained on a normal vs. pneumonia dataset first as more images are in the public sphere for making such a comparison. After

obtaining acceptable performance on this separate task, they fine-tune their model on a smaller COVID-19 dataset. This segmentation-classification pipeline ultimately achieves a final test accuracy of 84.67 percent. The authors of this paper, unfortunately, used Paul Mooney's chest X-ray dataset on Kaggle [116] to obtain pneumonia images when training their final classifier. This contains images from Kermany et al.'s dataset [16] of children's lungs, which means unfortunately that Alom et al.'s [78] classifier was incorrectly biased.

Yeh et al. [20] combined several public datasets as well as datasets from several private medical institutions when training their segmentation-classification pipeline. Unlike the two previous studies, the authors of this work look like they have constructed an unbiased dataset. They do, however, reference several private datasets that are unavailable to the research community. It is therefore impossible to directly compare our pipeline against their work. They initially trained a U-Net segmentation model [46] as a preprocessing step to exclude non-informative regions of CXRs from their model. Yeh et al. [20] trained this segmentation unit on the Montgomery County X-ray Set and the Shenzhen Hospital X-ray Set [57]. After training their segmentation unit, they obtained a dice similarity coefficient of 88 percent. Following this preprocessing step, they trained a DenseNet-121 [31] classifier on segmented images and obtained a COVID-19 sensitivity of 83.33% on their validation set. Their hold-out test set contained 306 COVID-19 images and their final COVID-19 sensitivity on this test set corrected to 81.8 percent.

Horry et al. [54] developed a segmentation–classification deep learning pipeline for diagnosing COVID-19 that was trained and tested on a relatively small preprocessed dataset. While Horry et al.'s [54] final curated dataset was not biased, it contained only 100 COVID-19 images, so it is difficult to ultimately know how well their work would translate to a larger number of images. Horry et al. [54] additionally removed images from their dataset which contained features they believed their model would have difficulty classifying. The authors' segmentation model was not based on a deep learning model. They simply used OpenCV's GrabCut function and reasoned that "that the lung area could be considered the foreground of the X-Ray image" [54]. After preprocessing they trained five base models with their segmented images (VGG-16 [30], VGG-19 [30], Inception-V3 [34], Xception [55], and

ResNet-50 [29]). Their best base model (VGG-19 [30]) ultimately achieved an F1-score of 81 percent.

Wehbe et al.'s [19] published deep learning pipeline that was trained on the largest COVID-19 X-ray dataset we have found reported in the literature. The authors developed their pipeline by working in collaboration with a private US medical institution. Their large classification dataset is therefore inaccessible to the public at this time. This dataset also appears to have not been improperly biased with the inclusion of incorrect data. The authors were aware of the need to divide their training and test sets by patient number. The authors chose to train their U-Net-based segmentation module [46] on the Montgomery [57] and JSRT [61] datasets. Wehbe et al. [19] in their study also created an ensemble model to detect COVID-19. Their final model contained a weighted average of 6 popular CNNs (Inception [34], Inception-ResNet [129] Xception [55], and ResNet-50 [29], and DenseNet-121 [31]). An important reason to include this paper in our discussion is that the authors managed to perform an interesting study that up until now we have not seen reproduced elsewhere. The authors commissioned a study involving five radiologists to determine the effectiveness of experts in the field in differentiating COVID-19 from other illnesses. This is important when trying to approximate Bayes error prior to building a deep learning model. Wehbe et al.'s [19] compared the results of their model with the performance of expert radiologists and discovered their model to a minor extent outcompetes them. Their final binary weighted average model obtained a final accuracy of 82% on their test set. The expert radiologists manually obtained a consensus accuracy of 81% on the same images. These final results coincided very nicely with one another.

Tabik et al. [56] created a dataset dubbed the "COVID-GR-1.0" dataset which was used in training their "COVID-SDNet" model in diagnosing COVID-19. Their dataset was divided in a novel fashion whereby COVID-19 positive patients were subdivided into four risk categories (normal-PCR+, mild, moderate, and severe). The authors created this dataset to see how many of weak COVID-19 cases would be analyzed by a prospective classifier correctly. More often than not, in COVID-19 datasets, there is an unequal number of severe COVID-19 patients. Typically, patients who end up undergoing a radiological examination

end up being patients experiencing increased complications. COVID-GR-1.0 is a small but well-curated dataset that has utility in that it can be employed to determine a classifier's efficacy on weak COVID-19 images. Tabik et al.'s [56] pipeline consisted of a segmentation module and a classification module that performs "inference based on the fusion of CNN twins." [56] The authors used a U-Net [46] segmentation module and trained it on the Montgomery County X-ray dataset [57], the Shenzhen Hospital X-ray datasets [57] and the RSNA Pneumonia CXR challenge dataset [58]. They calculated the smallest rectangle around each segmented image and added a border containing 2.5% of the pixels around each rectangle to obtain their final masked images. The X-rays they segmented were, therefore, never fully masked. The authors did not want to exclude relevant information in these images that could contain useful diagnostic information. After performing binary classification on their segmented COVID-GR-1.0 dataset, Tabik et al.'s [56] classifier obtained a COVID-19 sensitivity of 72.59%.

Teixeira et al. [59] designed a segmentation–classification pipeline used to diagnose COVID-19 that consisted of a U-Net [46] and InceptionV3 [34] CNN. Their U-Net [46] segmentation module was trained on images and masks that were hand-picked from a mixture of public datasets ([57], [60], [61]). The number of images and mask pairings they chose in the Darwin V7 labs [60] segmentation dataset (489) was significantly lower than the total number of pairings available in that dataset (6504). This approach looks as though it allowed them to train their U-Net [46] to have a higher dice similarity coefficient (0.982) than other segmentation units we have seen in the literature for this task. For classification they otherwise used the RYDLS-20 dataset [62]. They had developed this dataset in a previous work and further added images to it to create a new "RYDLS-20-v2" dataset. They attempted to use several classifiers but ultimately found that using an InceptionV3 [34] CNN resulted in giving them their best overall multiclass performance metrics.

Oh et al. [50] published a novel "patch-based deep neural network architecture with random patch cropping" [50] for detecting COVID-19. Their model initially begins with a preprocessing step whereby a fully convolutional DenseNet-103 [31] segments incoming chest X-rays. The authors thereafter use a ResNet-18 [29] on the segmented images for

classification. The authors generate 100 randomly cropped patches from the previously segmented chest X-rays and feed those patches through ResNet-18s [29] as well. In this process, the authors have selected a sufficient number of lung patches to ensure that the entire surface area of the segmented lungs is covered. The authors of this paper unfortunately selected images from Kermany et al. [16] to include in their work and thereby biased their classifier.

Abdullah et al. [63] implemented a segmentation – classification pipeline that used a unique segmentation unit and ensemble model for classification. Their segmentation unit, the Res-CR-Net, is a new kind of segmentation model the authors introduced in a previous study [64] that does not contain the same encoder-decoder structure that the popular U-Net [46] contains. According to the authors, the Res-CR-Net "combines residual blocks based on separable, atrous convolutions [65, 66] with residual blocks based on recurrent NNs [67]." [64] The authors trained their Res-CR-Net [64] on several open-source sets of masks and images [57, 60, 61]. They acquired their classification dataset from the Henry Ford Health System (HFHS) hospital in Detroit. This private dataset contained 1417 COVID-negative patients and 848 COVID-positive patients. The authors used this dataset to train a unique hybrid convnet called the "CXR-Net" that contains a Wavelet Scattering Transform (WST) block [68, 69], an attention block containing two MultiHeadAttention layers [70, 71], and several convolutional residual blocks. This segmentation-classification pipeline ultimately achieved an accuracy of 79.3% and an F1 Score of 72.3% on their test set.

5.3 Proposed Network Architecture

5.3.1 Segmentation Dataset

To train our segmentation model, we looked at the datasets used in our literature review and decided to use the Darwin V7 Labs dataset[60]. We opted in favor of this dataset for three reasons. The first reason was its overall size. The Darwin V7 Labs dataset [60] is significantly larger (6504 images/masks) than most lung segmentation datasets. This being the case, we were able to train a robust segmentation unit that could accurately operate on a

Table 5.1: Number of Images/Masks in the Preprocessed Darwin V7 Labs Dataset [60]

	Number of Image/Mask Pairings
V7 Labs preprocessed training set	5102
V7 Labs preprocessed test set	1275

wide range of chest X-rays. Our second reason for using the dataset involved considerations involving the regions of the chest X-rays that its masks cover. Most masks in popular lung segmentation datasets include only the lungs. The Darwin V7 Labs [60] masks, however, included space next to the lungs. This left room for the heart to not be excluded. Initially, we did not give the heart and its size any consideration. Eventually, we came to realize, however, that cardiomegaly (an enlarged heart) is found in 29.9% of COVID-19 patients [11]. This symptom would not show up with most general-purpose lung segmentation masks. Our third reason for using the Darwin V7 Labs dataset [60] was that its masks were created for patients with a variety of conditions. Some masks were created for normal patients and others were created for patients exhibiting a variety of lung pathologies including COVID-19, bacterial pneumonia, viral pneumonia, Pneumocystis pneumonia, fungal pneumonia, and Chlamydophila pneumonia.

Some preprocessing was required on the Darwin V7 labs dataset [60] to create a model that operated correctly on the segmentation unit we later created. The segmentation unit we chose for this study was a ResUnet [130], and this segmentation unit was designed for 256x256 images/masks. We needed to perform some data wrangling using the JSON files that were included with the dataset to ensure that images smaller than 256x256 were excluded. The JSON files provided with the Darwin V7 Labs dataset [60] had a field indicating which kind of X-ray each image was. We, therefore, were able to automate a process whereby we removed all of the lateral X-rays that were sparsely hidden throughout the dataset. Our dataset, therefore, solely contained posteroanterior (PA) X-rays. After preprocessing, we were left with 6377 masks/image pairings. We finally divided this preprocessed Darwin V7 Lab dataset [60] into the 80% training / 20% validation split shown in Table 5.1.

5.3.2 Classification Datasets

In medical imaging, the ability of a model to generalize to new examples typically is limited by the size of the training set. Because research into imaging COVID-19 is relatively recent, there is only approximately a year's worth of images that have been collected for classification purposes. For this reason, most published studies cannot present a model that can be deployed in a clinical setting. This study is no different, although in the work presented here we have taken significant steps forward in remediating several mistakes we have witnessed in the datasets of most papers.

When we first started gathering data, we initially realized that publicly available datasets generally have very little metadata available. That being the case, we decided to build a classifier that works on images alone. While doing so, we came to realize that the classification datasets in many studies have been incorrectly assembled. The majority of papers that have focused on differentiating COVID-19 from similar illnesses have cited using Kermany et al.'s [16] images in their dataset. As we have previously mentioned in our related works section, this dataset is composed of children that are suffering from various forms of bacterial and viral pneumonia. Since the lungs of small children have different features than adult lungs, we realized these images should not be included in our final classification dataset. This dataset likely poses more of a problem in biasing classifiers that are trained on nonsegmented images. The bones of adults are fused and the bones of children are not fused. This is feature can easily be picked up by a CNN. Kermany et al.'s [16] dataset, however, still would pose an issue even with a segmentation unit as the spatial features of adult lungs would differ from those of children's lungs. The classifiers in studies that include this dataset, therefore, can pick up features both internal and external to the lungs that are inconsistent between adults' and childrens' lungs. This has, unfortunately, lead to the unfair biasing of several COVID-19 classifiers in the literature.

Another difficulty facing many studies is the lack of metadata accompanying images. At least some metadata is required alongside images to ensure that X-rays from individual patients do not get mixed in the training and test/validation sets. This problem of data leakage, we believe, is an issue in some studies we have reviewed. We find it disconcerting

that most studies do not mention how they ensured the separation of patients' X-ray scans between training and test sets. An enthusiasm surrounding finding the most images possible has resulted in a large number of images being harvested from medical research papers. Wang et al. [51] last year released a popular 'COVIDx5' dataset that has been able to avoid this pitfall. Their COVIDx5 dataset [51] is relatively large and we used 14,258 CXR images from their dataset. In total, this consists of 617 COVID-19 images, 8066 normal images, and 5575 pneumonia images.

We added more COVID-19 images to the COVIDx5 dataset [51] because of the large COVID-19 class imbalance that existed within it. We hoped it would help to reduce overfitting in our classifier. We therefore added 922 COVID-19 images from the MIDRC-RICORD-1C database [121] and 2474 images from the BIMCV dataset [122]. In total, we constructed a dataset that contains 4013 COVID-19 images, 8066 normal images, and 5445 pneumonia images. The images from the COVIDx5 dataset [51] had the necessary metadata needed to allow us to split these images into three sets (80% training/ 10% validation/ 10%test) without creating data leakage. The MIDRC-RICORD-1C dataset [121] and BIMCV dataset [122] were released long after the COVIDx5 dataset [51], and none of these datasets had any relation with one another. It was therefore possible to split the COVID-19 images within these datasets into three sets without creating data leakage between them. The BIMCV [122] COVID-19 images were entirely used in the training set and the COVIDx5 [51] COVID-19 images were entirely split evenly between the validation and test set. The MIDRC-RICORD-1C [121] COVID-19 images were used in all three sets. The MIDRC-RICORD-1C [121] images came with metadata. Fortunately, the metadata allowed us to be able to divide the images from the MIDRC-RICORD-1C [121] dataset by patient between our training and validation/test sets. In this way we were able to create the datasets shown in Tables 5.2 and 5.3. We created both multiclass (3-class) and binary datasets to later compare our segmentation-classification pipeline with models that are reported in various other papers. It was important to produce our large COVID-19 dataset with both validation and test sets to help mitigate concerns that have been brought up by Wehbe et al. [19] concerning overfitting.

Table 5.2: Number of Images in Our Multiclass Training and Test Sets

	COVID-19	**Normal**	**Pneumonia**
Multiclass Training Set	3209	7262	4771
Multiclass Validation Set	402	402	402
Multiclass Test Set	402	402	402

Table 5.3: Number of Images in Our Binary Training and Test Sets

	COVID-19	**Non-COVID-19**
Binary Training Set	3209	12033
Binary Validation Set	402	402
Binary Test Set	402	402

Table 5.4: Number of Images in the COVID-GR-1.0 Training and Test Sets [56]

	COVID-19	**Normal**
COVID-GR-1.0 Training Set	340	340
COVID-GR-1.0 Test Set	86	86

In addition to the above dataset that we created, we also directly tested our model on another dataset that was used in Tabik et al.'s [56] study. We wanted to test our segmentation-classification against Tabik et al.'s [56] pipeline because their model worked on many of the same principles ours did. Their model used a segmentation algorithm that leaves more pixels surrounding the lungs in the images they segment. It has been difficult to find segmentation-classification pipelines like our own with unbiased and correctly constructed datasets. We were unable to find a study to directly compare ourselves against that uses a segmentation-classification pipeline and has a larger public dataset. Tabik et al.'s [56] study used a very conservative dataset that was meant to measure the performance of a deep learning model on weaker COVID-19 cases. Their "COVID-GR-1.0" binary dataset has 426 COVID-19 patients and 426 normal patients. The authors originally split this dataset into a 80% training / 20% test split. The dataset split in this format is shown in Table 5.4.

5.3.3 System Design

We set out to construct our deep learning segmentation-classification pipeline by first choosing an appropriate segmentation module to preprocess our classification dataset. We tested the preprocessed Darwin V7 Labs dataset [60] on a host of different segmentation modules including the popular U-Net [46], the ResUnet [130], the ResUNet-a [131], the TransResUNet [132] and U-Nets containing VGG and DenseNet backbones. Before training, we required the images in our preprocessed V7 Labs dataset [60] to undergo additional preprocessing in the form of image augmentation. During augmentation, we set the rotation range to 180 degrees, width/height shift ranges to 30%, shear range to 20%, zoom range to 20%, and set horizontal flips to true. We ultimately found that our best results on the preprocessed Darwin V7 Labs dataset [60] were obtained using Zhang et al's ResUnet [130]. We therefore decided to move forward using this segmentation module in our pipeline. The ResUnet [130] on our preprocessed V7 Labs dataset ultimately obtained a dice similarity coefficient of 95.04% after 45 epochs. This segmentation module uses a 7-level architecture shown in Fig 5.1. Its architecture can be understood by dividing it conceptually into three main parts. The first part of the architecture is an encoder that fits the images input into the module into smaller and more compact representations. The last main segment of this architecture is the decoder which "recovers the representations to a pixel-wise categorization, i.e., semantic segmentation." [130] The second middle part of the classifier serves as a bridge between the encoder at the ResUNet's [130] input and the decoder at the ResUNet's [130] output.

Having discussed the segmentation portion of the deep learning pipeline, we now move on to discussing the models that we have constructed for classifying COVID-19 images. We trained our preprocessed multiclass training set on a DenseNet-201 [31], a ResNet-152 [29], and a VGG-19 [30]. Each of these models was set to pretrained ImageNet weights. While designing each of these models we added an extra dense layer and dropout layer to the end of each model. The DenseNet-201's [31] extra dense layer contained 128 neurons. The ResNet-152's [29] extra dense layer contained 1024 neurons. The VGG-19's [30] extra dense layer contained 4096 neurons. Each of the activation functions in these dense layers was set to a

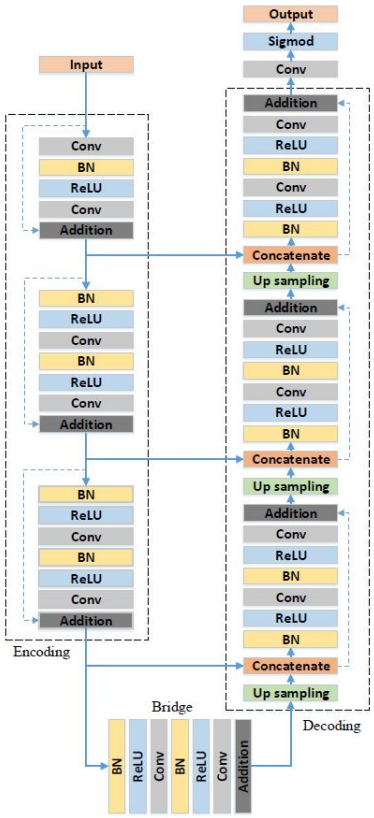


Figure 5.1: ResUnet architecture [130]

ReLU activation. The dropout layer added to the end of each model was set to a dropout rate of 10 percent. This helped each model to avoid overfitting. We constructed both binary and multiclass versions of all of these classifiers. For the binary version of each classifier, we replaced the final softmax layer of each classifier with a single neuron containing a sigmoid activation function. For the multiclass version of each of these classifiers, our final layers contained three neurons each and had a softmax activation function.

Prior to training our DenseNet-201 [31], ResNet-152 [29], and VGG-19 [30] CNNs, we noticed that a class imbalance existed in our multiclass and binary datasets. There were lower amounts of COVID-19 images in comparison to the other categories of images in our datasets. We, therefore, needed to weigh the loss functions of our classifiers to correct for this imbalance. Prior to training our classifiers, we additionally used image augmentation on the segmented images from our ResUNet [130] to prevent overfitting in our classifiers. Using Kera's ImageDataGenerator class, we set the rotation range to 15%, the width/height range

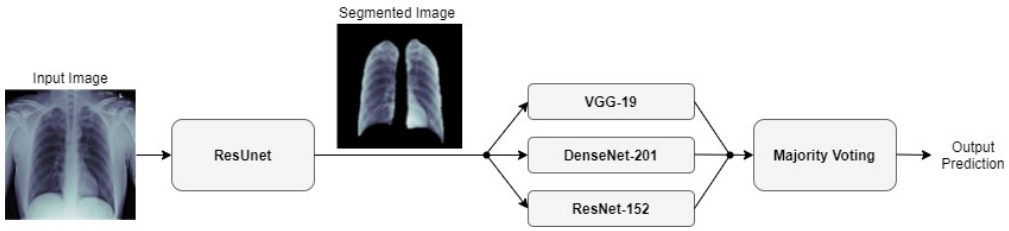


Figure 5.2: Proposed network architecture for COVID-19 classification with majority voting.

to 15%, the shear range to 15%, the zoom range to 15%, and horizontal flips to true. Our training and test set batch sizes were set to 32. In addition to segmenting and augmenting our classification datasets, we also normalized our data. In doing so, we ensured that the scaled data in each batch had a mean of zero and a standard deviation of one.

After our initial preprocessing steps, we trained the final fully-connected layers of each classifier alone for five epochs. We used the ADAM optimizer during this training and kept the ADAM optimizer set to its default settings. After performing this training, for each classifier we progressively unfroze each model's layers and fine-tuned our models at a fixed learning rate of $1x10^{-5}$ until each model hit its highest possible validation accuracy. Prior to unfreezing progressive layers in our models, we froze the moving mean and moving variance of the batches in our models' batchnormalization layers to keep these parameters fixed to their pretrained ImageNet weights. After training each of our CNNs to their optimal validation accuracies, we constructed a majority voting ensemble and a weighted average ensemble that combined all of our classifiers together. The weighted average ensemble's VGG-19 and DenseNet-121 were weighted more heavily with weights of 0.4, while the ResNet-152 had a weight of 0.2. After the probabilities of each classifier were combined with their corresponding weights, a probability for each class could be determined. The class with the highest probability was chosen as the final prediction. The majority voting classifier worked by assigning a final vote to each classifier and the category with the most votes was the final prediction. We constructed both a binary version and a multiclass version of each type of ensemble classifier. Illustrations showing our overall deep learning pipelines can be observed in Figs. 5.2 and 5.3.

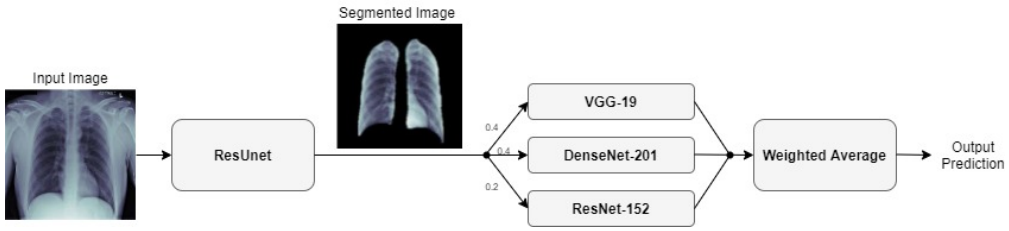


Figure 5.3: Proposed network architecture for COVID-19 classification with weighted averaging.

5.4 Experimental Results

5.4.1 Performance Evaluation

Within the COVID-19 deep learning literature, we have found that most studies report common evaluation metrics. To compare our models against the literature we have reviewed, we have chosen to report the accuracy, sensitivity, specificity, F1-Score, precision, recall, negative predictive value (NPV), positive predictive value (PPV), and area under the receiver operating characteristic curve (AUC-ROC) of our deep learning pipeline.

We first set out to train our multiclass and binary DenseNet-201 [31], ResNet-152 [29], and VGG-19 [30] models for five epochs. On each model, we obtained a validation accuracy that ranged between 70 and 80 percent. This largely mirrored the performance of expert radiologists who had their expertise measured in a research study led by Wehbe et al. [19]. We performed this initial work using our multiclass and binary training sets before moving on to test ourselves against Tabik et al.'s [56] model (which was trained on the "COVID-GR-1.0" dataset). During this initial stage, we worked toward increasing the accuracy of all three of these classifiers by unfreezing each model during training progressively.

On our multiclass dataset set, we obtained final validation set accuracies of 82.16% on our DenseNet-201 [31], 84.25% on our ResNet-152 [29], and 81.09% on our VGG-19 [30]. Likewise, on our multiclass dataset set, we obtained final test set accuracies of 82.42% on our DenseNet-201 [31], 81.84% on our ResNet-152 [29], and 77.53% on our VGG-19 [30]. The test accuracies we obtained all saw a decrease of 2% - 4% from their corresponding validation set accuracies. When we ensembled all three classifiers into majority voting and weighted

Table 5.5: The Performance of Our Classifiers on Our Multiclass Dataset

Classifier	Val. Acc.	Test Acc.	Val. COV. Sen.	Test COV. Sen.
DenseNet-201	82.16%	82.42%	84.32%	82.09%
ResNet-152	84.25%	81.84%	82.59%	76.86%
VGG-19	81.09%	77.53%	81.34%	75.62%
Weighted Avg. Ens.	87.40%	84.07%	85.32%	81.34%
Maj. Voting Ens.	87.14%	84.00%	86.07%	81.84%

Table 5.6: The Performance of Our Classifiers on Our Binary Dataset

Classifier	Val. Acc.	Test Acc.	Val. COV. Sen.	Test COV. Sen.
DenseNet-201	89.55%	88.43%	88.81%	85.82%
ResNet-152	85.70%	82.09%	91.04%	84.82%
VGG-19	89.55%	84.55%	89.30%	83.08%
Weighted Avg. Ens.	91.17%	91.17%	91.79%	91.79%
Maj. Voting Ens.	90.67%	88.18%	91.29%	87.06%

average ensembles, we saw an increase in performance on our validation and test sets. For our weighted average ensemble, we obtained a validation set accuracy of 87.40% and a test set accuracy of 84.07%. For our majority voting ensemble, we obtained a validation set accuracy of 87.14% and a test set accuracy of 84.00%. In both instances, we found that the test set accuracies of both ensembles outperformed our best individual classifier (DenseNet-201 [31]) by more than 1.5%. The overall performance of our three classifiers and our ensembles on our multiclass validation and test sets can be seen in Table 5.5. Our binary classifiers were trained in the same way as our muilticlass classifiers. The overall performance of our three classifiers and our ensembles on our binary validation and test sets can be seen in Table 5.6. Tables 5.7 - 5.10 show a larger suite of statistics generated on the multiclass and binary test sets using both our weighted average and majority voting ensembles. Figs. 5.4 - 5.7 show the corresponding confusion matrices generated by our weighted average and majority voting ensembles on our multiclass and binary test sets. Fig. 5.8 shows the AUC-ROC curves generated by our weighted average ensembles.

Table 5.7: Weighted Average Ensemble Performance Metrics After Training on Our Multiclass Training Set

	TP	TN	FP	FN	Acc.	Sens.	Spec.	PPV	NPV	F1
COVID-19	327	737	67	75	0.88	0.81	0.92	0.83	0.94	0.81
Normal	362	742	55	40	0.92	0.90	0.93	0.87	0.95	0.88
Pneumonia	325	734	70	77	0.88	0.81	0.91	0.82	0.91	0.81

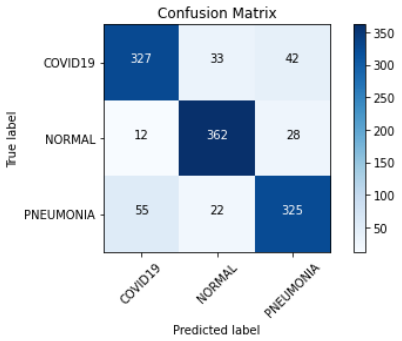


Figure 5.4: Confusion matrix from weighted average ensemble after training on our multiclass training set.

Table 5.8: Majority Voting Ensemble Performance Metrics After Training on Our Multiclass Training Set

	TP	TN	FP	FN	Acc.	Sens.	Spec.	PPV	NPV	F1
COVID-19	329	729	75	73	0.88	0.82	0.91	0.81	0.91	0.82
Normal	362	754	50	40	0.93	0.90	0.94	0.88	0.95	0.89
Pneumonia	322	736	68	80	0.88	0.81	0.92	0.83	0.90	0.81

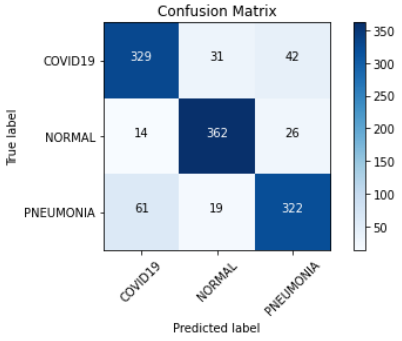


Figure 5.5: Confusion matrix from majority voting ensemble after training on our multiclass training set.

Table 5.9: Weighted Average Ensemble Performance Metrics After Training on Our Binary Training Set

	TP	TN	FP	FN	Acc.	Sens.	Spec.	PPV	NPV	F1
COVID-19	369	364	38	33	0.91	0.92	0.91	0.91	0.92	0.91
Non-COVID-19	364	369	33	38	0.91	0.91	0.92	0.92	0.91	0.91

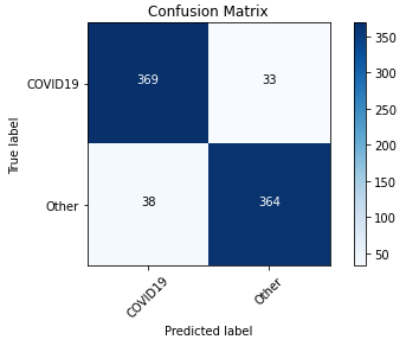


Figure 5.6: Confusion matrix from weighted average ensemble after training on our binary training set.

Table 5.10: Majority Voting Ensemble Performance Metrics After Training on Our Binary Training Set

	TP	TN	FP	FN	Acc.	Sens.	Spec.	PPV	NPV	F1
COVID-19	350	359	43	52	0.88	0.87	0.89	0.89	0.87	0.88
Non-COVID-19	359	350	52	43	0.88	0.89	0.87	0.87	0.89	0.88

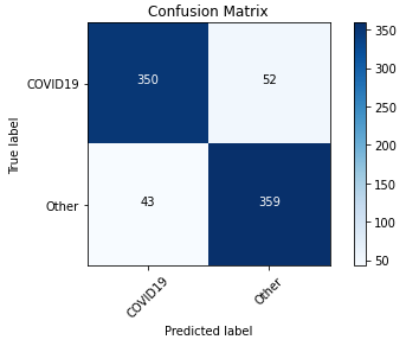
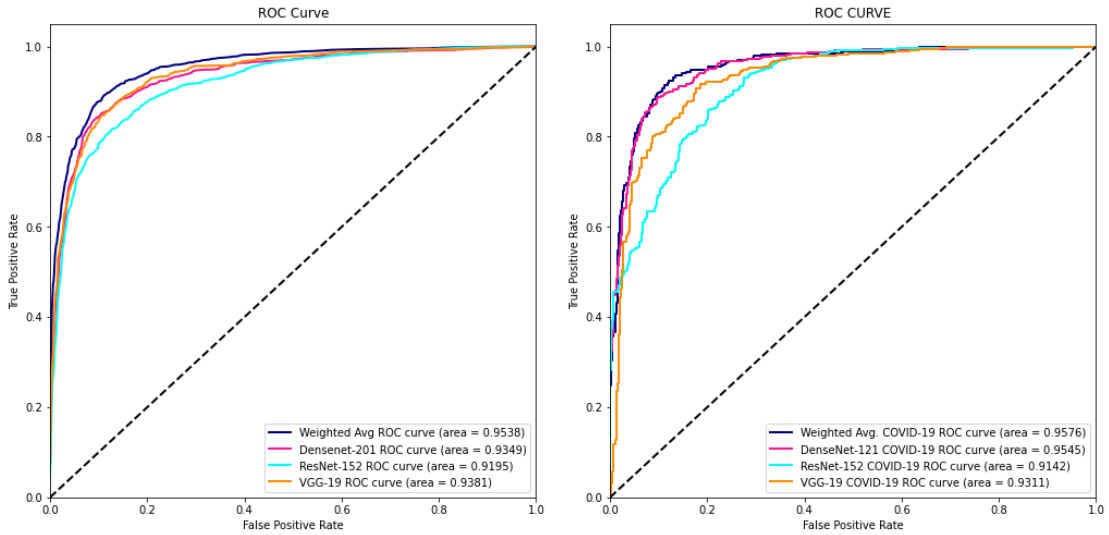
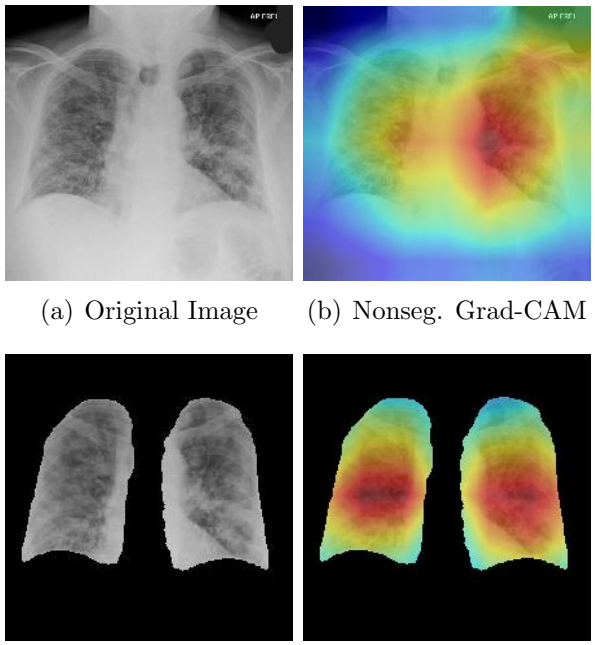


Figure 5.7: Confusion matrix from majority voting ensemble after training on our binary training set.



(a) (b)

Figure 5.8: AUC-ROC graphs of (a) Our multiclass weighted average ensemble trained on our multiclass training set and (b) Our binary weighted average ensemble trained on our binary training set.



(a) Original Image (b) Nonseg. Grad-CAM

(c) Segmented Image (d) Segmented Grad-CAM

Figure 5.9: Example of a segmented and non-segmented Grad-CAM heatmap produced by our DenseNet-201.

Table 5.11: Our Binary Models Vs. COVID-SDNet on the COVID-GR-1.0 Dataset [56]

Classifier	Val. Acc.	Val. COV. Sen.
Weighted Avg. Ens.	76.74%	77.91%
Maj. Voting Ens.	76.16%	73.26%
COVID-SDNet	76.18%	72.59%

After training and testing our segmentation-classification pipeline on our datasets, we also tested our binary pipeline directly against Tabik et al.'s [56] COVID-SDNet model. The details of their publicly available "COVID-GR-1.0" dataset [56] are provided in Section 5.3.2. It should be noted that Tabik et al.'s [56] dataset is smaller than ours and composed in a fashion whereby the authors collaborated with radiologists to intentionally incorporate weaker COVID-19 images into their dataset. This being the case, lower performance metrics should be expected out of this dataset. These two datasets have been designed to deal with separate problems and a detailed discussion concerning these differences is presented in the following section. Table 5.11 shows how our models compared against Tabik et al.'s [56] COVID-SDNet model.

Every deep learning expert working in computer vision understands that it is necessary to validate the final version of a classifier after it has been trained. In medical imaging, saliency maps are widely employed on computer vision models to ensure that these models are correctly identifying important features in an image. In radiology, it is common for deep learning models to incorrectly focus on necklaces, medical devices, and the text within X-ray scans. The reason we included a segmentation unit in our study was to ensure that our model's CNNs were rejecting unnecessary image details outside of the boundaries of the lungs. We used a Grad-CAM [39] in this study to ensure that our segmentation module was doing its job correctly in assisting our models to pick up the correct features of COVID-19. A Grad-CAM [39] functions by using the final feature maps in the last convolutional layer of a CNN to signal regions of importance within an image. We were interested in studying our CNNs that were trained on segmented images. We therefore devised a plan to compare them with CNNs that were trained on nonsegmented images. Fig. 5.9 shows the performance of our a DenseNet-201 [31] after being trained on segmented and nonsegmented X-rays. Our

DenseNet-201 [31] was one of the three CNNs that we used in constructing our majority voting and weighted average ensembles. Part (b) of Fig. 5.9 shows the performance of our DenseNet-201 [31] on a test image after it was trained without a segmentation module. The red parts of the heatmap indicate the primary parts of the image that the DenseNet-201 [31] focused on when determining a patient has COVID-19. The orange/yellow portions of the heatmap represent areas of medium importance. The green/blue areas of the Grad-CAM [39] heatmap represented areas that were the least important diagnostically in determining that a patient is COVID-19 positive. Unfortunately, portions of the red and orange/yellow parts of the heatmap in part (b) of Fig. 5.9 are focused on areas outside of the lungs. The area that the Grad-CAM [39] partially focused on in the upper right-hand side of the image was a problem. This area should have been irrelevant to a COVID-19 diagnosis. When our DenseNet-201 [31] was trained on segmented images however, its behavior improved as is shown in part (d) of Fig. 5.9. We monitored the performance of our model in this way to ensure that our model was picking up the features of COVID-19 that we highlighted in section 5.1.

5.4.2 Discussion

Wehbe et al. [19] conducted an important study that measured the performance of practicing radiologists on a private COVID-19 vs. non-COVID-19 dataset. In our work, we took it upon ourselves to build a COVID-19 dataset of comparable size. We wanted to measure our pipeline's ability to compete with the radiologists in their study and their model. We were more specifically interested in comparing our pipeline's COVID-19 sensitivity with the radiologists in Wehbe et al.'s [19] study given the problems concerning RT-PCR test sensitivity we have read about in scientific journals. The radiologists' consensus sensitivity in Wehbe et al.'s study [19] was 70%. All of our ensembles, including those trained on the weaker images in the "COVID-GR-1.0" dataset [56], obtained a higher COVID-19 sensitivity. The COVID-19 sensitivity of the five expert radiologists in Wehbe et al.'s [19] study versus that of our ensembles' can be seen in Table 5.12.

Table 5.12: The COVID-19 Sensitivity of Five Expert Radiologists in Wehbe et al.'s Study [19] Vs. Our Classifiers

Group/Individual/Classifier	COV. Sens.
The Consensus of Expert Radiologists	70%
The Best Radiologist	76%
The Worst Radiologist	60%
Weighted Avg. Ensemble (Our Binary dataset)	91.79%
Weighted Avg. Ensemble (COVID-GR-1.0 dataset [56])	77.91%

As can be seen in Table 5.12, when we compare our ensemble models with the performance of the radiologists in Wehbe et al.'s [19] study, we outperform even the best radiologist's COVID-19 sensitivity. In Table 5.12, another item that stands out is the difference in sensitivity between the ensemble we trained on our binary dataset versus the ensemble we trained on the COVID-GR-1.0 dataset [56]. This discrepancy can be explained by the higher number of weak COVID-19 images that were intentionally placed by radiologists in the "COVID-GR-1.0" dataset [56]. Tabik et al. [56] created the "COVID-GR-1.0" dataset to measure the performance of their classifier on COVID-19 images that are more difficult to classify. Even after we trained our ensemble model on this extremely conservative dataset, we still managed to obtain a higher sensitivity than the radiologists in Wehbe et al.'s [19] study.

When we constructed our binary dataset, we built our dataset so as to respond to a criticism that Wehbe et al. [19] mentioned in their paper concerning the size of public datasets. Wehbe et al.'s [19] study found that the consensus accuracy and sensitivity of expert radiologists are 81% and 70% respectively. After training their ensemble model, Wehbe et al. [19] found that their system achieved a test accuracy of 82% and test sensitivity of 75%. Many other studies however have reported performance metrics that are much higher than this. Wehbe et al. [19] explained this by showing how models with extremely high metrics often have very small COVID-19 datasets. They posited that if the number of COVID-19 images in these other studies increased, these models would see a correction. They believed that early COVID-19 deep learning models were overfitting on small COVID-19 datasets. We therefore set out to construct a larger COVID-19 dataset than any other public

COVID-19 dataset we have seen in the literature thus far. We felt that it was additionally important to create separate validation and test sets in order to ensure that overfitting does not occur. For the same reason, we also ensured that each of our CNNs had dropout layers in their second last layers.

Wehbe et al.'s [19] criticism of small public datasets was not the only concern we have ended up discovering when using public datasets. We later realized that many public datasets include images from Kermany et. al.'s [16] dataset which contains the chest X-rays of young children suffering from various forms of pneumonia. It is incorrect to take a model that was trained on children's X-rays and deploy it on adult X-rays. When we attempted to use such a dataset for training one of our CNNs, we obtained extremely high-performance metrics (accuracy/sensitivity between 98% - 100%). We noticed that several deep learning segmentation-classification pipelines [14, 78, 50] made this mistake. In addition to this, we have come to suspect that some authors may have unintentionally biased their classifiers by mixing multiple images from individual patients in their training and test sets. In Table 5.13 we compare our work with other segmentation-classification pipelines that have not made the mistake of incorrectly biasing their models with improperly constructed datasets.

Table 5.13: Performance of Similar Segmentation-Classification Pipelines Without Dataset Composition Issues

Research Paper	**Seg. DSC**	**Acc.**	**COV. Sens.**	
Yeh et al. [20]				
3-class	0.88	-	82%	
Wehbe et al. [19]				
2-class	-	82%	75%	-
Abdullah et al. [63]				
2-class	0.96	79%	-	
Tabik et al. [56]				
2-class (COVID-GR-1.0 dataset)	-	76%	73%	
Ours				
Best 3-class Ens. (Maj. Vot.)	0.95	84%	82%	
Best 2-class Ens. (Wei. Avg.)	0.95	91%	92%	
2-class (COVID-GR-1.0 dataset)	0.95	77%	78%	

Our best three-class and two-class ensembles should only be compared against the first three classifiers in Table 5.13. Our three-class and two-class ensembles were trained on a dataset that we built after gathering as many COVID-19 images as possible. The authors of the first three papers in Table 5.13, composed their datasets in the same way. The COVID-GR-1.0 dataset [56], however, was trained intentionally on weak COVID-19 images resulting in a classifier that should be treated in isolation. In comparing our segmentation unit with Yeh et al.'s [20] U-Net [46] segmentation model, our ResUNet [130] achieved a dice similarity coefficient that was 7 percent higher. In terms of dataset size, our COVID-19 dataset contained over 3000 more COVID-19 images. Yeh et al. [20] had a smaller dataset, therefore, and were more likely to have overfit their model. Our model was, therefore, more likely to face downward pressure in our performance metrics. Our three-class model, however, was still capable of obtaining the same COVID-19 sensitivity as Yeh et al.'s [20] model. It likely was able to do so with the help of better segmentation and the use of a majority voting ensemble. This indicates that on datasets that are constructed with as many COVID-19 images as possible, a three-class model (COVID-19 vs. Normal vs. Pneumonia) can reasonably achieve a COVID-19 sensitivity of 82%. Our two-class weighted average ensemble outperformed Wehbe et al.'s [19] classifier by a substantial margin. This may have been caused by a difference in our approach to segmentation. Wehbe et al.'s [19] classifier was trained to crop out the smallest rectangle that a patient's lungs can fit within. Our segmentation unit was trained on a set of masks that removed more pixels than Wehbe et al.'s [19] segmentation unit. It still managed though to not eliminate the pixels showing the heart. Our weighted average ensemble also outperformed Abdullah et al.'s [63] model despite our having a segmentation unit that under-performed Abdullah et al.'s Res-CR-Net [64] by one percent. We obtained a two-class accuracy that was 12 percent better than Abdullah et al.'s [64] classification model. We believe this is a result of our having constructed an extremely robust weighted average classification ensemble.

It should be noted that there are instances where using a segmentation unit can reduce a model's accuracy. While segmentation units should generally always help a classifier's accuracy, we have noticed in our work that classifiers without a segmentation unit can lock onto features of an image that are external to the lungs. Sometimes this helps to increase a

CNN's ability to classify particular images. For instance, if one category of images has more text than another you might notice the Grad-CAM [39] heatmaps for that category focusing on text. Our segmentation unit removed this possibility from happening and ultimately allowed us to boost our model's accuracy in a more honest fashion.

The approach to creating datasets that is followed by the vast majority of research papers is to obtain as many COVID-19 images as possible. During the early stages of the coronavirus pandemic, there was a lack of COVID19 images and many papers were being published that likely were overfitting on datasets containing only a couple of hundred COVID-19 images. Tabik et al. [56] published their paper when fewer COVID-19 images existed and therefore their paper only contained 426 COVID-19 images. The authors of this paper obtained the help of an expert radiologist. This radiologist located PCR positive images that did not have the visual features of COVID-19. They infused their dataset with such images and wanted to see the effect this would have. They eventually found that their classifier could identify COVID-19 in 85 to 97 percent of moderate to severe images. Mild COVID-19 images, however, could only be diagnosed correctly 46 percent of the time. They did not publish the accuracy of their classifier on Normal PCR positive images. We have to imagine that the accuracy for Normal PCR positive images was even lower. In total, their classifier had a final accuracy of 76 percent and COVID-19 sensitivity of 73 percent. When our binary weighted average ensemble was trained on their dataset, it achieved a 77 percent accuracy and a 78 percent COVID-19 sensitivity.

Tabik et al.'s [56] dataset was the only dataset that we could obtain that allowed us to directly compare our pipeline with another author's segmentation-classification pipeline. It has been difficult to find publicly available datasets such as this one where the authors have made clear how they segmented and classified their images. Tabik et al. [56] did not report a dice similarity coefficient because they segmented their images in such a way so as to create a small cropped rectangle around the lungs. This is similar in principle to how we segmented our images. We chose the Darwin V7 Labs dataset [60] for training our segmentation unit because the masks in this dataset left more room around the lungs to show the heart. We believe that if a segmentation unit were to remove these pixels, that

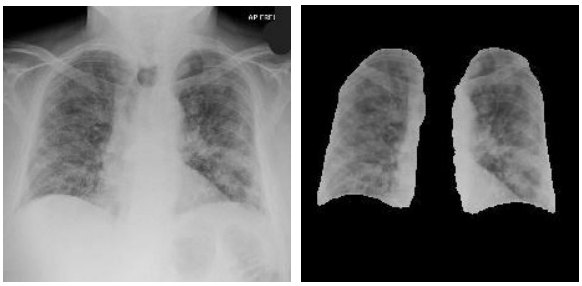
COVID-19 symptoms like cardiomegaly could go unobserved by a classifier. We believe that our weighted average ensemble is ultimately what allowed us to achieve an improved accuracy and improved COVID-19 sensitivity when comparing our model with Tabik et al.'s [56] model. Our segmentation unit also likely helped as well, as it rejected a greater number of superfluous pixels around the lungs in comparison to Tabik et al.'s [56] segmentation methodology.

By now, when comparing the research studies in chapters 4 and 5, the reader is likely wondering why the sensitivities in the second study are lower than the sensitivities in the first study. After all, segmentation is supposed to improve the performance of a downstream classifier. So why does the second study with its advanced methods not outperform the first study? One of the most significant differences between the models in both studies is the first study's use of pretraining. When designing a system to have a segmentation unit alongside a pretrained model, the pretrained model should originally be trained on segmented images. The original ChexNet architecture [45] that our COV-SNET model was based on originally was trained on over 100,000 non-segmented images. This is an important point. Segmentation takes away a lot of the image that was originally available to be classified. In this chapter, we previously discussed that cardiomegaly (an enlarged heart) is found in 29.9% of COVID-19 patients [11]. While the segmentation unit we used in the second study usually left the heart alone, in a minority of instances, segmentation failed to keep the heart within our images. This problem can be seen in the two X-ray scans processed by our segmentation unit in Fig. 5.10. The lungs of most images in our classification dataset typically contained a black or grey background that was easy for a segmentation unit to analyze. The segmentation unit, therefore, did a very good job at leaving the lungs alone in most images. That, however, was not always true of the heart.

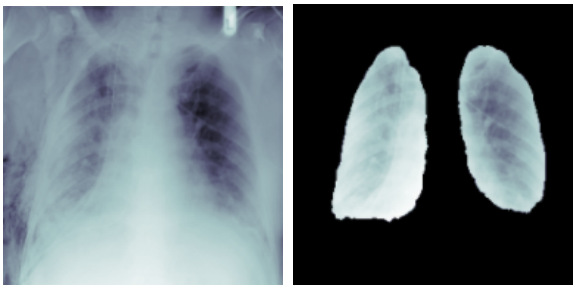
When we first implemented our segmentation unit, we trained it on the Darwin V7 Labs dataset [60] because it had masks that were designed to keep the pixels containing a patient's heart. We eventually came to realize, however, that our segmentation unit occasionally removed the heart in problematic images. After analyzing the Grad-CAMs [39] of our classified images with and without a segmentation unit, we came to discover that our

CNNs often focused on the heart when determining whether a patient is COVID-19 positive. Our segmentation unit, therefore, occasionally removed information that our COV-SNET model relies on. The training of the original ChexNet [45] model that was utilized in the first study relied on full X-ray images. One of the 14 pathologies it was trained to pick up on was cardiomegaly. Since the pretrained model we used in our COV-SNET model was trained to diagnose cardiomegaly, our COV-SNET model was able to use the features of cardiomegaly when diagnosing a COVID-19 patient. When we used a segmentation unit on our COV-SNET model, it underperformed. This led us to work with ImageNet pretrained CNNs to see if we could obtain any improvement. Our ImageNet pretrained CNNs outperformed our COV-SNET model in classifying segmented X-rays. Segmentation also improved the performance of our ImageNet pretrained CNNs. Our ImageNet pretrained CNNs were trained on segmented images and were never pretrained to pick on the features of cardiomegaly. They therefore still under-performed the version of our COV-SNET model that was trained on nonsegmented images. Ensembling our ImageNet pretrained CNNs further improved the performance of our segmentation-classification pipeline. It allowed us to outcompete the other COVID-19 segmentation-classification pipelines that we have found in the literature. We were not capable, however, of outcompeting our COV-SNET model from chapter 4.

We have also found that COVID-19 images can occasionally contain a massive amount of white area within a patient's lungs in severe COVID-19 cases. This is additionally another area where a segmentation unit can struggle. Most segmentation units are trained on datasets where the lungs mostly have a black or grey background. It is difficult to segment lungs that are completely full of white opacities. This is because a segmentation unit can interpret very white lungs as belonging to the background that is supposed to be segmented out. This becomes a problem when there is an insufficient number of X-rays in a segmentation dataset with severe pneumonia. While the Darwin V7 Labs dataset [60] contained pneumonia images, there was not a sufficient number of severe cases where the lungs were mostly white. We believe this would have helped our segmentation unit to perform better on severe images. An example of this problem from our segmentation unit is shown in Fig. 5.11. It is difficult to know how much of an effect this had on our final classifier. In the majority of images, this problem could not be found. We believe, however, that this may have contributed to

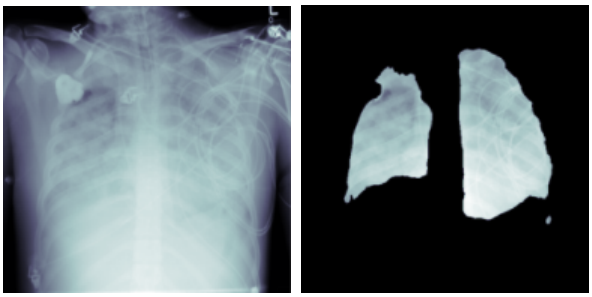


(a) Original Image (b) Segmented Image



(c) Original Image (d) Segmented Image

Figure 5.10: Comparing a good vs. problematic X-ray scan processed by our ResUnet [130]. The heart in the right lung should not be removed.



(a) Original Image (b) Segmented Image

Figure 5.11: An example of our segmentation unit struggling with extremely white lungs.

reduced performance metrics and that more severe pneumonia images were necessary for training our segmentation unit.

Out of the ensembles we constructed, the weighted average ensemble seems to be the most reliable in producing the best results. The weighted average ensembles outperform the majority voting ensembles in all performance metrics, except for in 3-class sensitivity (a half percent minor difference). The test and validation accuracy metrics of the 2-class

and 3-class weighted average ensembles were higher than those produced by the majority voting ensembles. When comparing the performance metrics between the 3-class weighted average ensemble and 3-class majority voting ensemble, it can be seen that they closely match each other. The 2-class weighted average ensemble outcompeted the 2-class majority voting ensemble by a substantial margin in all of our recorded metrics. The test sensitivity of our 2-class majority voting classifier was 4.74% higher than the test sensitivity of the 2-class majority voting classifier.

Chapter 6

Conclusion

6.1 Meeting the Objectives

This research aimed to design X-ray-based deep learning models capable of diagnosing COVID-19. The research in our studies was limited to working on diagnosis alone as there was an insufficient amount of metadata to work on prognosis. The focus of this research was to achieve the highest sensitivity possible in comparison with other deep learning computer vision models and molecular tests. RT-PCR tests are presently considered the gold standard for COVID-19 testing and they have been reported to have a sensitivity ranging between 70 to 90 percent [6]. Initially, there were significant limitations in terms of the number of X-ray images available. The authors of most papers in the literature review used datasets that were incorrectly assembled. Deep learning models trained on public datasets prior to early 2021 all have experienced dataset size limitations in terms of the number of COVID-19 X-rays available. This has made all of these models susceptible to possible overfitting. There are now, however, several thousand COVID-19 X-rays publicly available. This recent surge in available COVID-19 X-rays allows for past authors still working in this space to check and see if their models will ultimately correct when training with a larger dataset. The two-class and three-class datasets that we have constructed contain the largest number of publicly available COVID-19 images that we have found in the literature.

At the beginning of our research, we started out hoping to benchmark our study against a popular dataset. This led us to use Wang et al.'s [51] COVIDx dataset. The COV-SNET

model ended up achieving a higher sensitivity than their model. The COV-SNET model additionally was trained with more COVID-19 images. This ultimately allowed us to ensure that our model was not overfitting on a dataset containing only a limited number of COVID-19 images. The COV-SNET models are currently capable of obtaining a higher COVID-19 sensitivity than all other models we have reviewed in the literature so far. We have restricted this analysis to those models that do not improperly use Kermany et al.'s [16] dataset or otherwise make any observable dataset composition mistakes. The COV-SNET models led to promising evaluation metrics in comparison with expert radiologists in the field [19]. We achieved two-class and three-class COVID-19 sensitivities of 95 percent. This study also achieved sensitivities that are superior to an RT-PCR test.

In training the segmentation-classification pipeline in our second study, we were ultimately able to design several ensembles that generated promising results. Our best two-class weighted average ensemble achieved a 91 percent COVID-19 accuracy and 92 percent COVID-19 sensitivity. We were additionally able to outcompete a segmentation-classification pipeline from Tabik et al. [56] that we directly compared our pipeline against. Our segmentation-classification pipeline also achieved a better overall sensitivity than all of the other models we indirectly compared ourselves against. These comparisons were limited to other published models that have been trained on correctly assembled datasets. In the end, our second study also achieved a higher sensitivity than that of an RT-PCR test.

6.2 Advantages and Shortcomings of the Proposed Deep Learning Systems

The COV-SNET model clearly showed that pretraining a deep learning model on a related set of images is a successful methodology. This was apparent from the beginning, when we started experimenting with Rajpurkar's ChexNet [45] model. The images that were used to train the original ChexNet were all frontal chest X-rays. They were all labeled as belonging to one of 14 different pathologies. This makes the original ChexNet model extremely versatile and allows it to be easily repurposed for diagnosing thoracic diseases like

COVID-19. We also slightly modified the original ChexNet [45] to have extra classification and dropout layers. This allowed us to avoid overfitting. Our dataset of COVID-19 X-rays is larger than any we have been able to find in the literature. We have divided it into training, test, and validation sets and ensured that there is no data leakage between our validation and test sets. Our dataset's size and structure has helped us to avoid the kind of overfitting that has been reported in previous studies.

We originally believed we could improve on the COV-SNET models by pairing them with a well-trained segmentation unit. This, unfortunately, was not possible. We later discovered that our segmentation unit removed parts of our X-ray images that assisted the COV-SNET model to diagnose COVID-19. Since cardiomegaly is found in 29.9% of COVID-19 patients, we came to realize that the COV-SNET model's performance was negatively affected. When using a Grad-CAM [39], we noticed that the COV-SNET models often focused on the heart when diagnosing COVID-19. There was nothing we could do, unfortunately, to correct for the shortcomings of our segmentation dataset. We later came to find out that the Darwin V7 Labs dataset [60] did not have enough COVID-19 masks and corresponding images to train a segmentation unit that will always segment COVID-19 images accurately. Errors from our segmentation unit dragged down the performance of our final classifiers. We eventually found that we had more success pairing our segmentation unit with ImageNet-based CNNs. They seemed to perform slightly better than models based on a ChexNet architecture [45]. We were unable to obtain the sensitivities of the classifiers in chapter 4 with individual ImageNet-based CNNs, so we constructed an ensemble of them to try to reach our previous study's sensitivities. While we failed to reach our previous study's sensitivities, we managed to perform better than the other segmentation-classification pipelines we have found in the literature. We also found that our Grad-CAM [39] heatmaps improved with segmentation and that our classifiers were focusing on more of the areas that they should be. We therefore obtained better validation results (in the visual sense) for our CNNs when using segmentation.

Should segmentation, therefore, have assisted our second study's classifier in diagnosing COVID-19? The answer is mixed. Our segmentation unit was not perfect due to

the limitations of our segmentation dataset. We therefore had errors in how our classifier processed X-rays. We believe that our segmentation unit required a better assortment of COVID-19 and pneumonia masks/images. That was the real limitation that we experienced in our second study. Ensembling was not able to make up for the fact that we needed a richer segmentation dataset than is currently available. With a better segmentation dataset, segmentation certainly would have assisted our classifier in diagnosing COVID-19.

6.3 Recommendations

Our models, as demonstrated in the experimental results, showed promising characteristics in terms of the Grad-CAM heatmaps and performance metrics they produced. Our models are currently not ready to be implemented in a clinical setting. For deep learning models such as ours to be advanced into a clinical setting, the medical community and AI experts require further collaboration. To the best of our knowledge, no study has been performed whereby every single incoming patient at a medical facility was tested for COVID-19 with an X-ray and RT-PCR test simultaneously. The COVID-19 images that can be found in public datasets tend to come from patients that were showing increased complications from their illness. In private datasets, the same problem likely exists as well since radiological evaluations are typically reserved for patients showing a concerning trend in the development of their illness. It is important to find out the proportion of incoming patients at a medical clinic that are COVID-19 positive after blind X-rays get administered to every patient. Anyone wanting to clinically implement a deep learning system such as ours may also benefit from blindly administering competing molecular tests (RT-PCR tests), antigen tests, and antibody tests on the same patients during this data-gathering stage. Additional metadata concerning each patient's age, sex, and relevant background details would also help tremendously.

We experienced some significant limitations due to the quantity of COVID-19 and pneumonia masks/images in our segmentation dataset. Better segmentation datasets are required. We need more segmentation datasets that have masks for both the lungs and

the heart. These datasets need to contain more severe cases of pneumonia and COVID-19. They need to contain images with opacities that are mostly filling the insides of the lungs. Aside from this, we need more COVID-19 X-ray images. These images need to come with accompanying metadata that outlines their original source. There are many studies with incorrectly assembled datasets. There is a need for trusted institutions to gather properly sourced images and for these images to be made available in an organized fashion to the public.

6.4 Contributions

From chapters 4 and 5, our findings indicate that it is possible to achieve a COVID-19 sensitivity with a deep learning X-ray-based model that is in line with or better than an RT-PCR test. This is important since novel independent methodologies are sometimes required to determine a patient's COVID-19 status. RT-PCR tests often initially give an incorrect reading when determining a patient's diagnosis. If an RT-PCR test is initially negative but a doctor strongly suspects the patient has COVID-19, an X-ray may be a good option to quickly determine the patient's diagnosis. This may be especially important if the patient has any risk factors that a COVID-19 diagnosis could complicate.

In this research, we have shown the effects of pretraining, segmentation, and ensembling on training an X-ray-based deep learning model in diagnosing COVID-19. Pretraining a model on a related task with over a hundred thousand X-rays has been shown to be an an extremely successful technique. Segmentation should be able to marginally improve the performance metrics of a classifier, but we need better segmentation masks and images to improve the quality of our segmentation. Ensembling was also shown to be a marginally effective technique. Our weighted average and majority voting ensembles improved the sensitivity of our classifiers in the second study.

6.5 Final Remarks

While the results of the two research studies presented in this work look promising, more work is required to implement them in a clinical setting. The same can be said for all of the other X-ray studies that we have reviewed. The addition of more COVID-19 images to public databases will no doubt help to further inform the research community as to which approaches are the most promising. Medical institutions in countries all over the world are in need of new diagnostic modalities that can help increase available COVID-19 testing capacity. Deep learning X-ray technology remains a promising candidate for fulfilling this incredibly important need. We believe that our models are a promising step forward towards radiologically automating the detection of COVID-19. With a little more time and resources invested in data-gathering processes, we believe that an X-ray-based COVID-19 deep learning model could one day allow for a truly better standard of care.

Bibliography

[1] Government of Canada. "covid-19 daily epidemiology update," jul. 09, 2021. [Online]. Available: https://health-infobase.canada.ca/covid-19/epidemiological-summary-covid-19-cases.html

[2] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841517301135

[3] 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 for coronavirus disease 2019 (covid-19) in china: A report of 1014 cases," *Radiology*, vol. 296, no. 2, pp. E32–E40, 2020, pMID: 32101510. [Online]. Available: https://doi.org/10.1148/radiol.2020200642

[4] Y. Fang, H. Zhang, J. Xie, M. Lin, L. Ying, P. Pang, and W. Ji, "Sensitivity of chest ct for covid-19: Comparison to rt-pcr," *Radiology*, vol. 296, no. 2, pp. E115–E117, 2020, pMID: 32073353. [Online]. Available: https://doi.org/10.1148/radiol.2020200432

[5] L. Luo, D. Liu, X.-l. Liao, X.-b. Wu, Q.-l. Jing, J.-z. Zheng, F.-h. Liu, S.-g. Yang, B. Bi, Z.-h. Li, J.-p. Liu, W.-q. Song, W. Zhu, Z.-h. Wang, X.-r. Zhang, P.-l. Chen, H.-m. Liu, X. Cheng, M.-c. Cai, Q.-m. Huang, P. Yang, X.-f. Yang, Z.-g. Han, J.-l. Tang, Y. Ma, and C. Mao, "Modes of contact and risk of transmission in covid-19 among close contacts," *medRxiv*, 2020.

[6] "Covid-19 laboratory testing qas," Public Health Ontario, July 17, 2021. [Online]. [Online]. Available: https://www.publichealthontario.ca/-/media/documents/lab/covid-19-lab-testing-faq.pdf?la=en

[7] L. Kucirka, S. Lauer, O. Laeyendecker, D. Boon, and J. Lessler, "Variation in false-negative rate of reverse transcriptase polymerase chain reaction–based sars-cov-2 tests by time since exposure," *Annals of Internal Medicine*, vol. 173, no. 4, pp. 262–267, 2020, pMID: 32422057. [Online]. Available: https://doi.org/10.7326/M20-1495

[8] Y. Pan, X. Li, G. Yang, J. Fan, Y. Tang, J. Zhao, X. Long, S. Guo, Z. Zhao, Y. Liu, H. Hu, H. Xue, and Y. Li, "Serological immunochromatographic approach in diagnosis with sars-cov-2 infected covid-19 patients," *Journal of Infection*, vol. 81, no. 1, pp. e28 – e32, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0163445320301754

[9] T. Liang, Z. Liu, C. Wu, C. Jin, H. Zhao, Y. Wang, Z. Wang, F. Li, J. Zhou, S. Cai, Y. Liang, H. Zhou, X. Wang, Z. Ren, and J. Yang, "Evolution of ct findings in patients with mild covid-19 pneumonia," *European Radiology*, vol. 30, no. 9, pp. 4865–4873, Sep. 2020.

[10] F. Song, N. Shi, F. Shan, Z. Zhang, J. Shen, H. Lu, Y. Ling, Y. Jiang, and Y. Shi, "Emerging 2019 novel coronavirus (2019-ncov) pneumonia," *Radiology*, vol. 295, no. 1, pp. 210–217, 2020, pMID: 32027573. [Online]. Available: https://doi.org/10.1148/radiol.2020200274

[11] D. Cozzi, M. Albanesi, E. Cavigli, C. Moroni, A. Bindi, S. Luvarà, S. Lucarini, S. Busoni, L. 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, 06 2020.

[12] J. Cleverley, J. Piper, and M. M. Jones, "The role of chest radiography in confirming covid-19 pneumonia," *BMJ*, vol. 370, 2020.

[13] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, "Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2003.11055

[14] S. Rajaraman, J. Siegelman, P. O. Alderson, L. S. Folio, L. R. Folio, and S. K. Antani, "Iteratively pruned deep learning ensembles for covid-19 detection in chest x-rays," *IEEE Access*, vol. 8, pp. 115 041–115 050, 2020.

[15] I. Apostolopoulos and M. Tzani, "Covid-19: Automatic detection from x-ray images utilizing transfer learning with convolutional neural networks," *Australasian physical and engineering sciences in medicine / supported by the Australasian College of Physical Scientists in Medicine and the Australasian Association of Physical Sciences in Medicine*, vol. 43, p. 635–640, Mar 2020.

[16] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122 – 1131.e9, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0092867418301545

[17] R. Hertel and R. Benlamri, "Cov-snet: A deep learning model for x-ray-based covid-19 classification," *Informatics in Medicine Unlocked*, vol. 24, p. 100620, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914821001106

[18] S. Wang, Y. Zha, W. Li, Q. Wu, X. Li, M. Niu, M. Wang, X. Qiu, H. Li, H. Yu, W. Gong, Y. Bai, L. Li, Y. Zhu, L. Wang, and J. Tian, "A fully automatic deep learning system for covid-19 diagnostic and prognostic

analysis," *European Respiratory Journal*, 2020. [Online]. Available: https://erj.ersjournals.com/content/early/2020/05/19/13993003.00775-2020

[19] R. M. Wehbe, J. Sheng, S. Dutta, S. Chai, A. Dravid, S. Barutcu, Y. Wu, D. R. Cantrell, N. Xiao, B. D. Allen, G. A. MacNealy, H. Savas, R. Agrawal, N. Parekh, and A. K. Katsaggelos, "Deepcovid-xr: An artificial intelligence algorithm to detect covid-19 on chest radiographs trained and tested on a large us clinical dataset," *Radiology*, vol. 0, no. 0, p. 203511, 0, pMID: 33231531. [Online]. Available: https://doi.org/10.1148/radiol.2020203511

[20] C.-F. Yeh, H.-T. Cheng, A. Wei, H.-M. Chen, P.-C. Kuo, K.-C. Liu, M.-C. Ko, R.-J. Chen, P.-C. Lee, J.-H. Chuang, C.-M. Chen, Y.-C. Chen, W.-J. Lee, N. Chien, J.-Y. Chen, Y.-S. Huang, Y.-C. Chang, Y.-C. Huang, N.-K. Chou, K.-H. Chao, Y.-C. Tu, Y.-C. Chang, and T.-L. Liu, "A cascaded learning strategy for robust covid-19 pneumonia chest x-ray screening," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2004.12786

[21] R. Hertel and R. Benlamri, "A deep learning segmentation-classification pipeline for x-ray-based covid-19 diagnosis," unpublished, sent to Biocybernetics and Biomedical Engineering.

[22] R. Hertel and R. Benlamri, "Deep learning techniques for covid-19 diagnosis and prognosis based on radiological imaging," unpublished, sent to ACM Computing Surveys.

[23] L. Yu, *Handbook of COVID-19 Prevention and Treatment*, Mar. 2020.

[24] S. Simpson, F. U. Kay, S. Abbara, S. Bhalla, J. H. Chung, M. Chung, T. S. Henry, J. P. Kanne, S. Kligerman, J. P. Ko, and H. 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." *Radiology: Cardiothoracic Imaging*, vol. 2, no. 2, p. e200152, 2020. [Online]. Available: https://doi.org/10.1148/ryct.2020200152

[25] 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. N. Hung, T. P. W. Lam, M. D. Kuo, and M.-Y. Ng, "Frequency and distribution of chest radiographic findings in patients positive for covid-19," *Radiology*, vol. 296, no. 2, pp. E72–E78, 2020, pMID: 32216717. [Online]. Available: https://doi.org/10.1148/radiol.2020201160

[26] L. J. M. Kroft, L. van der Velden, I. H. Girón, J. J. H. Roelofs, A. de Roos, and J. Geleijns, "Added value of ultra-low-dose computed tomography, dose equivalent to chest x-ray radiography, for diagnosing chest pathology," *Journal of thoracic imaging*, vol. 34, no. 3, p. 179—186, May 2019. [Online]. Available: https://europepmc.org/articles/PMC6485307

[27] A. Christe, J. Charimo-Torrente, K. Roychoudhury, P. Vock, and J. E. Roos, "Accuracy of low-dose computed tomography (ct) for detecting and characterizing

the most common ct-patterns of pulmonary disease," *European Journal of Radiology*, vol. 82, no. 3, pp. e142 – e150, 2013, breast Imaging. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0720048X12004809

[28] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv*, 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[31] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *arXiv*, vol. abs/1608.06993, 2016. [Online]. Available: https://arxiv.org/abs/1608.06993

[32] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv*, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360

[33] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv*, vol. abs/1905.11946, 2019. [Online]. Available: http://arxiv.org/abs/1905.11946

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv*, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[35] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *arXiv*, Jun. 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[36] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, Z. Cheng, T. Yu, J. Xia, Y. Wei, W. Wu, X. Xie, W. Yin, H. Li, M. Liu, and B. Cao, "Clinical features of patients infected with 2019 novel coronavirus in wuhan, china," *The Lancet*, vol. 395, 01 2020.

[37] W.-J. Guan, Z.-y. Ni, Y. Hu, W. Liang, C.-Q. Ou, J.-x. He, L. Liu, H. Shan, C.-l. Lei, D. Hui, B. Du, L.-j. Li, G. Zeng, K.-Y. Yuen, R.-c. Chen, C.-l. Tang, T. Wang, P.-y. Chen, J. Xiang, and N.-s. Zhong, "Clinical characteristics of coronavirus disease 2019 in china," *New England Journal of Medicine*, vol. 382, 02 2020.

[38] J. Zhang, Y. Xie, Z. Liao, G. Pang, J. Verjans, W. Li, Z. Sun, J. He, Y. Li, C. Shen *et al.*, "Viral pneumonia screening on chest x-ray images using confidence-aware anomaly detection," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2003.12338v3

[39] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *CoRR*, vol. abs/1610.02391, 2016. [Online]. Available: http://arxiv.org/abs/1610.02391

[40] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of covid-19 cases using deep neural networks with x-ray images," *Computers in Biology and Medicine*, vol. 121, p. 103792, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010482520301621

[41] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016. [Online]. Available: http://arxiv.org/abs/1612.08242

[42] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[43] A. Haghanifar, M. M. Majdabadi, and S. Ko, "Covid-cxnet: Detecting covid-19 in frontal chest x-ray images using deep learning," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2006.13807

[44] A. Mangal, S. Kalia, H. Rajgopal, K. Rangarajan, V. Namboodiri, S. Banerjee, and C. Arora, "Covidaid: Covid-19 detection using chest x-ray," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2004.09803

[45] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv*, 2017. [Online]. Available: https://arxiv.org/abs/1711.05225

[46] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *arXiv*, 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[47] A. S. Al-Waisy, S. Al-Fahdawi, M. A. Mohammed, K. H. Abdulkareem, S. A. Mostafa, M. S. Maashi, M. Arif, and B. Garcia-Zapirain, "Covid-chexnet: hybrid deep learning framework for identifying covid-19 virus in chest x-rays images," *Soft Computing*, 2020.

[48] N. E. M. Khalifa, M. H. N. Taha, A. E. Hassanien, and S. Elghamrawy, "Detection of coronavirus (covid-19) associated pneumonia based on generative adversarial networks and a fine-tuned deep transfer learning model using chest x-ray dataset," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2004.01184

[49] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "Covidgan: Data augmentation using auxiliary classifier gan for improved covid-19 detection," *IEEE Access*, vol. 8, pp. 91 916–91 923, 2020.

[50] Y. Oh, S. Park, and J. C. Ye, "Deep learning covid-19 features on cxr using limited training data sets," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2688–2700, 2020.

[51] 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 radiography images," *Scientific Reports*, vol. 10, no. 19549, 03 2020.

[52] A. Wong, M. J. Shafiee, B. Chwyl, and F. Li, "Ferminets: Learning generative machines to generate efficient neural networks via generative synthesis," *CoRR*, vol. abs/1809.05989, 2018. [Online]. Available: http://arxiv.org/abs/1809.05989

[53] Z. Q. Lin, M. J. Shafiee, S. Bochkarev, M. S. Jules, X. Wang, and A. Wong, "Do explanations reflect decisions? A machine-centric strategy to quantify the performance of explainability algorithms," *CoRR*, vol. abs/1910.07387, 2019. [Online]. Available: http://arxiv.org/abs/1910.07387

[54] M. Horry, S. Chakraborty, M. Paul, A. Ulhaq, B. Pradhan, M. Saha, and N. Shukla, "X-ray image based covid-19 detection using pre-trained deep learning models," *engrXiv*, 2020.

[55] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv*, vol. abs/1610.02357, 2016. [Online]. Available: http://arxiv.org/abs/1610.02357

[56] S. Tabik, A. Gómez-Ríos, J. L. Martín-Rodríguez, I. Sevillano-García, M. Rey-Area, D. Charte, E. Guirado, J. L. Suárez, J. Luengo, M. A. Valero-González, P. García-Villanova, E. Olmedo-Sánchez, and F. Herrera, "Covidgr dataset and covid-sdnet methodology for predicting covid-19 based on chest x-ray images," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 12, pp. 3595–3605, 2020.

[57] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wáng, P.-X. Lu, and G. Thoma, "Two public chest x-ray datasets for computer-aided screening of pulmonary diseases," *Quant Imaging Med Surg.*, vol. 4, no. 6, p. 475–477, 2014.

[58] Radiological Society of North America, "Rsna pneumonia detection challenge," Kaggle, 2018. [Online]. Available: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data

[59] L. O. Teixeira, R. M. Pereira, D. Bertolini, L. S. Oliveira, L. Nanni, G. D. C. Cavalcanti, and Y. M. G. Costa, "Impact of lung segmentation on the diagnosis and explanation of covid-19 in chest x-ray images," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2009.09780

[60] Darwin V7 Labs, "Covid-19 chest x-ray dataset," 2020. [Online]. Available: https://darwin.v7labs.com/v7-labs/covid-19-chest-x-ray-dataset

[61] J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K. Komatsu, M. Matsui, H. Fujita, Y. Kodera, and K. Doi, "Development of a digital image database

for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules," *American Journal of Roentgenology*, vol. 174, no. 1, pp. 71–74, 2000.

[62] R. M. Pereira, D. Bertolini, L. O. Teixeira, C. N. S. Jr., and Y. M. G. Costa, "COVID-19 identification in chest x-ray images on flat and hierarchical classification scenarios," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2004.05835

[63] H. Abdulah, B. Huber, S. Lal, H. Abdallah, L. L. Palese, H. Soltanian-Zadeh, and D. L. Gatti, "Cxr-net: An artificial intelligence pipeline for quick covid-19 screening of chest x-rays," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2103.00087

[64] H. Abdallah, A. Liyanaarachchi, M. Saigh, S. Silvers, S. Arslanturk, D. J. Taatjes, L. Larsson, B. P. Jena, and D. L. Gatti, "Res-cr-net, a residual network with a novel architecture optimized for the semantic segmentation of microscopy images," *Machine Learning: Science and Technology*, vol. 1, no. 1, p. 045004, 2020.

[65] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv*, 2016. [Online]. Available: http://arxiv.org/abs/1606.00915

[66] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv*, 2017. [Online]. Available: http://arxiv.org/abs/1706.05587

[67] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[68] E. Oyallon, E. Belilovsky, and S. Zagoruyko, "Scaling the scattering transform: Deep hybrid networks," *arXiv*, 2017. [Online]. Available: https://arxiv.org/abs/1703.08961

[69] E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. Blaschko, and E. Belilovsky, "Scattering networks for hybrid representation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, p. 2208–2221, Sep 2019. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2018.2855738

[70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv*, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[71] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 4055–4064. [Online]. Available: http://proceedings.mlr.press/v80/parmar18a.html

[72] A. Amyar, R. Modzelewski, and S. Ruan, "Multi-task deep learning based ct imaging analysis for covid-19: Classification and segmentation," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/04/21/2020.04.16.20064709

[73] M. Polsinelli, L. Cinque, and G. Placidi, "A light cnn for detecting covid-19 from ct scans of the chest," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2004.12837

[74] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)." *arXiv*, vol. 2, 2016. [Online]. Available: https://arxiv.org/abs/1511.07289

[75] J. Mockus, *On Bayesian Methods for Seeking the Extremum*, ser. Lecture Notes in Computer Science, G. I. Marchuk, Ed. Springer, 1974, vol. 27. [Online]. Available: https://doi.org/10.1007/3-540-07165-2_55

[76] H. Ko, H. Chung, W. S. Kang, K. W. Kim, Y. Shin, S. J. Kang, J. H. Lee, Y. J. Kim, N. Y. Kim, H. Jung, and J. Lee, "Covid-19 pneumonia diagnosis using a simple 2d deep learning framework with a single chest ct image: Model development and validation," *J Med Internet Res*, vol. 22, no. 6, p. e19569, Jun 2020. [Online]. Available: http://www.jmir.org/2020/6/e19569/

[77] H. S. Maghdid, A. T. Asaad, K. Z. Ghafoor, A. S. Sadiq, and M. K. Khan, "Diagnosing covid-19 pneumonia from x-ray and ct images using deep learning and transfer learning algorithms," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2004.00038

[78] M. Z. Alom, M. Rahman, M. S. Nasrin, T. M. Taha, and V. K. Asari, "Covidmtnet: Covid-19 detection with multitask deep learning approaches," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2004.03747

[79] M. Z. Alom, M. Hasan, C. Yakopcic, and T. M. Taha, "Inception recurrent convolutional neural network for object recognition," 2017.

[80] F. Shan, Y. Gao, J. Wang, W. Shi, N. Shi, M. Han, Z. Xue, and Y. Shi, "Lung infection quantification of covid-19 in ct images with deep learning," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2003.04655

[81] M. Han, G. Yao, W. Zhang, G. Mu, Y. Zhan, X. Zhou, and Y. Gao, *Segmentation of CT Thoracic Organs by Multi-resolution VB-nets*, ser. CEUR Workshop Proceedings, C. Petitjean, S. Ruan, Z. Lambert, and B. Dubray, Eds. CEUR-WS.org, 2019, vol. 2349. [Online]. Available: http://ceur-ws.org/Vol-2349/SegTHOR2019_paper_1.pdf

[82] F. Shi, L. Xia, F. Shan, D. Wu, Y. Wei, H. Yuan, H. Jiang, Y. Gao, H. Sui, and D. Shen, "Large-scale screening of covid-19 from community acquired pneumonia using infection size-aware classification," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2003.09860

[83] F. Santosa and W. W. Symes, "Linear inversion of band-limited reflection seismograms," *SIAM journal on scientific and statistical computing*, 1986.

[84] Z. Tang, W. Zhao, X. Xie, Z. Zhong, F. Shi, J. Liu, and D. Shen, "Severity assessment of coronavirus disease 2019 (covid-19) using quantitative features from chest ct images," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2003.11988

[85] R. D. Rudyanto, S. Kerkstra, E. M. van Rikxoort, C. Fetita, P.-Y. Brillet, C. Lefevre, W. Xue, X. Zhu, J. Liang, İlkay Öksüz, D. Ünay, K. Kadipaşaog˘lu, R. S. J. Estépar, J. C. Ross, G. R. Washko, J.-C. Prieto, M. H. Hoyos, M. Orkisz, H. Meine, M. Hüllebrand, C. Stöcker, F. L. Mir, V. Naranjo, E. Villanueva, M. Staring, C. Xiao, B. C. Stoel, A. Fabijanska, E. Smistad, A. C. Elster, F. Lindseth, A. H. Foruzan, R. Kiros, K. Popuri, D. Cobzas, D. Jimenez-Carretero, A. Santos, M. J. Ledesma-Carbayo, M. Helmberger, M. Urschler, M. Pienn, D. G. Bosboom, A. Campo, M. Prokop, P. A. de Jong, C. O. de Solorzano, A. Muñoz-Barrutia, and B. van Ginneken, "Comparing algorithms for automated vessel segmentation in computed tomography scans of the lung: the vessel12 study," *Medical Image Analysis*, vol. 18, no. 7, pp. 1217 – 1232, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S136184151400111X

[86] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 34, no. 2, pp. 187–220, 1972. [Online]. Available: http://www.jstor.org/stable/2985181

[87] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American Statistical Association*, vol. 53, no. 282, pp. 457–481, 1958. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1958.10501452

[88] X. Wang, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, and C. Zheng, "A weakly-supervised framework for covid-19 classification and lesion localization from chest ct," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2615–2625, 2020.

[89] F. Liao, M. Liang, Z. Li, X. Hu, and S. Song, "Evaluate the malignancy of pulmonary nodules using the 3-d deep leaky noisy-or network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3484–3495, 2019.

[90] S. Jin, B. Wang, H. Xu, C. Luo, L. Wei, W. Zhao, X. Hou, W. Ma, Z. Xu, Z. Zheng, W. Sun, L. Lan, W. Zhang, X. Mu, C. Shi, Z. Wang, J. Lee, Z. Jin, M. Lin, H. Jin, L. Zhang, J. Guo, B. Zhao, Z. Ren, S. Wang, Z. You, J. Dong, X. Wang, J. Wang, and W. Xu, "Ai-assisted ct imaging analysis for covid-19 screening: Building and deploying a medical ai system in four weeks," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/03/23/2020.03.19.20039354

[91] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: Redesigning skip connections to exploit multiscale features in image segmentation," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/1912.05074

[92] X. Mei, H.-C. Lee, K.-y. Diao, M. Huang, B. Lin, C. Liu, Z. Xie, Y. Ma, P. Robson, M. Chung, A. Bernheim, V. Mani, C. Calcagno, K. Li, S. Li, H. Shan, J. Lv, T. Zhao,

J. Xia, and Y. Yang, "Artificial intelligence–enabled rapid diagnosis of patients with covid-19," *Nature Medicine*, vol. 26, pp. 1–5, Aug. 2020.

[93] Y. Wang, X. Mei, C. Liu, T. Deyer, J. Zeng, C. Xia, J. Schefflein, L. Jia, H. Yu, F. Jiang, C. Yang, P. Zhou, H. Chang, P. Robson, A. Doshi, D. Mendelson, H. Zhu, C. Powell, Y. Yang, and W. Li, "A generalized deep learning approach for evaluating secondary pulmonary tuberculosis on chest computed tomography," *SSRN Electronic Journal*, Jan. 2019. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3441821

[94] Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, J. Chen, H. Zhao, Y. Jie, R. Wang, Y. Chong, J. Shen, Y. Zha, and Y. Yang, "Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/02/25/2020.02.23.20026930

[95] M. Rahimzadeh, A. Attar, and S. M. Sakhaei, "A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/09/01/2020.06.08.20121541

[96] H. X. Bai, R. Wang, Z. Xiong, B. Hsieh, K. Chang, K. Halsey, T. M. L. Tran, J. W. Choi, D.-C. Wang, L.-B. Shi, J. Mei, X.-L. Jiang, I. Pan, Q.-H. Zeng, P.-F. Hu, Y.-H. Li, F.-X. Fu, R. Y. Huang, R. Sebro, Q.-Z. Yu, M. K. Atalay, and W.-H. Liao, "Artificial intelligence augmentation of radiologist performance in distinguishing covid-19 from pneumonia of other origin at chest ct," *Radiology*, vol. 296, no. 3, pp. E156–E165, 2020, pMID: 32339081. [Online]. Available: https://doi.org/10.1148/radiol.2020201491

[97] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, K. Cao, D. Liu, G. Wang, Q. Xu, X. Fang, S. Zhang, J. Xia, and J. Xia, "Using artificial intelligence to detect covid-19 and community-acquired pneumonia based on pulmonary ct: Evaluation of the diagnostic accuracy," *Radiology*, vol. 296, no. 2, pp. E65–E71, 2020, pMID: 32191588. [Online]. Available: https://doi.org/10.1148/radiol.2020200905

[98] O. Gozes, M. Frid-Adar, H. Greenspan, P. D. Browning, H. Zhang, W. Ji, A. Bernheim, and E. Siegel, "Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection & patient monitoring using deep learning ct image analysis," *arXiv preprint arXiv:2003.05037*, 2020. [Online]. Available: https://arxiv.org/abs/2003.05037

[99] O. Gozes, M. Frid-Adar, N. Sagie, H. Zhang, W. Ji, and H. Greenspan, "Coronavirus detection and analysis on chest ct with deep learning," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2004.02640

[100] C. Jin, W. Chen, Y. Cao, Z. Xu, Z. Tan, X. Zhang, L. Deng, C. Zheng, J. Zhou, H. Shi, and J. Feng, "Development and evaluation of an ai

system for covid-19 diagnosis," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/06/02/2020.03.20.20039834

[101] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html

[102] L. Zhou, Z. Li, J. Zhou, H. Li, Y. Chen, Y. Huang, D. Xie, L. Zhao, M. Fan, S. Hashmi, F. Abdelkareem, R. Eiada, X. Xiao, L. Li, Z. Qiu, and X. Gao, "A rapid, accurate and machine-agnostic segmentation and quantification method for ct-based covid-19 diagnosis," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2638–2652, 2020.

[103] P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[104] B. Planche and E. Andres, *Hands-on Computer Vision with TensorFlow 2: Leverage Deep Learning to Create Powerful Image Processing Apps with TensorFlow 2.0 and Keras*. Packt Publishing, 2019. [Online]. Available: https://books.google.ca/books?id=XkL5xQEACAAJ

[105] A. Geron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, 1st ed. O'Reilly, 2017.

[106] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[107] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, ser. JMLR Proceedings. JMLR.org, 2015, vol. 37.

[108] S. Saha, "A comprehensive guide to convolutional neural networks — the eli5 way," Towardsdatascience, Dec. 15, 2018. [Online]. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[109] IBM. "convolutional neural networks," oct. 20, 2020. [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks

[110] DeepAI. "padding (machine learning)," accessed jun.25, 2021. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/padding

[111] M. Yani, S. Irawan, and M. S.T., "Application of transfer learning using convolutional neural network method for early detection of terry's nail," *Journal of Physics: Conference Series*, vol. 1201, p. 012052, 05 2019.

[112] C.-Y. Chiang, C. Barnes, P. Angelov, and R. Jiang, "Deep learning-based automated forest health diagnosis from aerial images," *IEEE Access*, vol. 8, pp. 144 064–144 076, 2020.

[113] Aqeel Anwar. "what is transposed convolutional layer?," mar. 06, 2020. [Online]. Available: https://towardsdatascience.com/what-is-transposed-convolutional-layer-40e5e6e31c11

[114] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 2642–2651. [Online]. Available: http://proceedings.mlr.press/v70/odena17a.html

[115] T. Rahman. (2020) Covid-19 radiography database. Kaggle. [Online]. Available: https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

[116] P. Mooney. (2018) Chest x-ray images (pneumonia). Kaggle. [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

[117] M. Z. Islam, M. M. Islam, and A. Asraf, "A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images," *Informatics in Medicine Unlocked*, vol. 20, p. 100412, 2020.

[118] H. Panwar, P. Gupta, M. K. Siddiqui, R. Morales-Menendez, P. Bhardwaj, and V. Singh, "A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images," *Chaos, Solitons, and Fractals*, vol. 140, p. 110190, 2020.

[119] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images," arXiv, 2020. [Online]. Available: https://arxiv.org/abs/2004.02696

[120] R. Karthik, R. Menaka, and H. M., "Learning distinctive filters for covid-19 detection from chest x-ray using shuffled residual cnn," *Applied Soft Computing*, vol. 99, p. 106744, 2021.

[121] E. Bilello, "Medical imaging data resource center (midrc) - rsna international covid-19 open radiology database (ricord) release 1c - chest x-ray covid+ (midrc-ricord-1c)," 2021. [Online]. Available: https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=70230281

[122] 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-covid19, datasets related to covid19's pathology course," 2020. [Online]. Available: https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/

[123] 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," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3462–3471.

[124] A. BenTaieb, J. Kawahara, and G. Hamarneh, "Multi-loss convolutional networks for gland analysis in microscopy," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, 2016, pp. 642–645.

[125] J. Arevalo, F. A. González, R. Ramos-Pollán, J. L. Oliveira, and M. A. Guevara Lopez, "Representation learning for mammography mass lesion classification with convolutional neural networks," *Computer Methods and Programs in Biomedicine*, vol. 127, pp. 248–257, 2016.

[126] A. Darwish, K. Leukert, and W. Reinhardt, "Image segmentation for the purpose of object-based classification," in *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477)*, vol. 3. Citeseer, 2003, pp. 2039–2041.

[127] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[128] M. Z. Alom, T. Aspiras, T. M. Taha, and V. K. Asari, "Skin cancer segmentation and classification with nabla-n and inception recurrent residual convolutional networks," *arXiv*, 2019. [Online]. Available: https://arxiv.org/abs/1904.11126

[129] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv*, 2016. [Online]. Available: http://arxiv.org/abs/1602.07261

[130] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[131] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, p. 94–114, Apr 2020.

[132] S. Reza, O. B. Amin, and M. Hashem, "Transresunet: Improving u-net architecture for robust lungs segmentation in chest x-rays," in *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 1592–1595.