

COVID-19 Classification in CT Images with Convolutional Neural Network-based Ensemble Learning

by

Dina Kushenchirekova

Submitted to the Department of Data Science
in partial fulfillment of the requirements for the degree of


Master of Science in Data Science

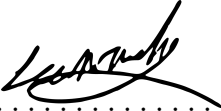
at the


NAZARBAYEV UNIVERSITY

April 2022

© Nazarbayev University 2022. All rights reserved.

Author 
Department of Data Science
April 26, 2022

Certified by 
Minho Lee
Assistant Professor
Thesis Supervisor

Certified by 
Adnan Yazici
Department Chair
Thesis Co-supervisor

Accepted by
Vassilios D. Tourassis
Dean, School of Engineering and Digital Sciences

COVID-19 Classification in CT Images with Convolutional Neural Network-based Ensemble Learning

by

Dina Kushenchirekova

Submitted to the Department of Data Science
on April 26, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science in Data Science

Abstract

The coronavirus infection has spread all over the world with great speed and the virus continues to grow and change. The COVID-19 infection that became a cause of the pandemic was a huge issue that people faced. Deep learning has a significant and important part in application of medical image analysis, and in this paper we use deep learning and convolutional neural network (CNN) methods. CNN helps us to classify our formations, since it is an effective tool at image classification. Deep learning is the field of Artificial Intelligence that copes with the classification problems, such as classifying and recognizing COVID-19 infection using computer tomography (CT) images that contain lungs. In the study, we utilize several of the most popular convolutional neural networks and evaluate them using the common metrics. Among 8 CNN architectures we used, which are VGG-19, VGG-16, MobileNetV2, Xception, ResNet50V2, DenseNet201, Inception-V3, and EfficientNetB3, the most efficient and outperforming was VGG-19, as it achieved the highest accuracy score. Specifically, the VGG-16 CNN architecture's accuracy on CovidX CT dataset is 0.97, on SARS-CoV-2 CT dataset is 0.95, and on UCSD COVID-CT dataset the score is 0.94. The arisen question now is how to properly utilize data mining to build an efficient detection system and mining framework. To answer the question we decide to use ensemble learning, which integrates fusion, modeling, and mining into a single model. Our proposal is ensemble learning algorithm that substantially stacks several neural network architectures into one. The logic behind the method is to extract features from the images using several of the above-mentioned models and combine the features into a "stack". The results suggest that the method performs better than each individual architecture. As the ensemble model considers each of the features and the losses provided by the models, the resultant loss is lower. This results in a higher accuracy score. In this way, we achieved the Ensemble model's accuracy of 0.9867 for the UCSD COVID-CT dataset, while the highest accuracy of the individual model was 0.945. As a result of the SVM integrated alternative methodology, ensemble model has shown the accuracy of 0.982 for SARS-COV-2 CT dataset.

Thesis Supervisor: Minho Lee
Title: Assistant Professor

Thesis Co-supervisor: Adnan Yazici
Title: Department Chair

Acknowledgments

First and foremost I am extremely grateful to my supervisors, Prof. Min-Ho Lee for invaluable advice, continuous support. His immense knowledge guidance and plentiful experience have encouraged me all the time of my thesis work. It has been an honor to work with the best professors and students of our country at Nazarbayev University. Finally, thanks to my parents, lab colleague Rakhat Abdrakhmanov and numerous friends who have been with me throughout this long process, always offering support and love.

Contents

1	Introduction	13
1.1	Motivation	13
1.2	Problem statement	14
1.3	Proposed approach	15
2	Related work	17
2.1	Convolutional neural network	17
2.2	Ensemble Machine Learning Algorithms	19
3	Methodology	21
3.1	Datasets	21
3.2	Ensemble learning algorithm	25
3.3	Models	28
3.4	Training	32
3.5	Evaluation	35
4	Experiments	37
4.1	Experimental Setup	37
4.1.1	Software	37
4.1.2	Evaluation Metrics	37
4.2	Comparison with Other Models	38
4.3	Comparison among other ensemble algorithms	39
4.4	Results	40

4.4.1	Performance comparison of the proposed CNN models	40
4.4.2	Performance of the Ensemble Learning algorithm	43
4.4.3	t-SNE (t-distributed stochastic neighbor embedding)	47
4.4.4	Heatmap visualization	49
5	Conclusion	51

List of Figures

3-1	Characteristic of radiological signs of coronavirus infection.	22
3-2	COVID-CT images from the SARS-COV-2 CT-Scan Dataset Covid-19 cases.	22
3-3	COVID-CT images from the UCSD COVID-CT Dataset Covid-19 cases.	23
3-4	COVID-CT images from the COVID-CT dataset Covid-19 cases. . . .	24
3-5	Ensemble learning methods structure.	25
3-6	Architecture of EfficientNet-B3 model.	32
4-1	Confusion matrix for EfficientNet-B3.	40
4-2	t-SNE for VGG16, VGG19 and EfficientNet-B3 for 3 datasets.	47
4-3	t-SNE for DenseNet201, InceptionV3 and MobileNetV2 for 3 datasets.	48
4-4	Predicting through Grad-CAM. Heatmap of Normal and Covid CT-scan images of models DenseNet201, VGG- 19, ResNet50V2.	50

List of Tables

3.1	SARS-COV-2 CT-Scan Dataset.	23
3.2	UCSD COVID-CT Dataset.	24
3.3	COVIDx-CT Dataset.	25
3.4	Computational time for each architecture.	33
4.1	Performance comparison of the proposed CNN models for CovidX CT dataset.	41
4.2	Performance comparison of the proposed CNN models for SARS-CoV-2 CT dataset.	42
4.3	Performance comparison of the proposed CNN models for UCSD COVID-CT dataset.	43
4.4	Performance of Ensemble models for SARS-CoV-2 CT dataset.	44
4.5	Performance of Ensemble models for USCD CT dataset.	45
4.6	Performance of Ensemble models for COVID-X dataset.	46
4.7	Performance of Ensemble with SVM.	46
4.8	Distribution of experimental data.	46

Chapter 1

Introduction

1.1 Motivation

In the modern world, we are surrounded by innovative technologies based on algorithms that resemble the work of the human brain in their specifics. As we know, not so long ago, deep learning was one of the key areas through which researchers have reached incredible heights in the field of medicine, namely in the analysis of medical images [1]. Deep learning and Artificial Intelligence can now solve non-routine tasks at a level close to humans, and sometimes better. This leads to the fact that with the help of such powerful tools, knowledgeable specialists can apply various methods such as machine learning that can make a huge contribution to medicine [2]. Based on this, the approaches will help improve treatment and care for people suffering from a particular disease. Health conditions such as COVID-19, oncological diseases, and ones that form tumors and spots on the lungs that affect them are currently one of the most important public health problems. Oncological diseases are among the leading causes of death worldwide [3]. But with the outbreak of the pandemic in 2019, the number of losses began to grow. As of May 2021, according to official generalized data, 3.4 million people died from coronavirus [4]. As it is stated in news portals, in many cases people died without passing any testing. Vaccines have been developed in some countries, and mass vaccination is underway in many [4]. Statistics provide comprehensive view on what is happening in the world and reveal of the burden of

the disease. The information described above is one of the reasons for conducting this study. Detection of such diseases at an early stage will help reduce the number of deaths and may also create new devices in medicine for detecting defects in medical images.

1.2 Problem statement

The task of working with medical images has always been difficult and time-consuming for even experienced specialists [5]. But with the right approach, we can get closer to solving the problem of classification and diagnosis of certain diseases. The number of people exposed to the COVID-19 is increasing [4]. Most often, people do not pay due attention to their health, thereby detecting the disease in the late stages, when it becomes more difficult to fight. Statistics show us unfavorable results, but with the help of deep learning, we can facilitate faster identification of problems [4]. In terms of solving the problem, namely classification, the issue is that we need to prepare the dataset first and then train a model on the special tools we have created. Such studies based on the recognition and analysis of the medical images, specifically on the nature and growth of cells, are very important for evaluating the effect of drugs and prescribing treatment and in some cases have much higher information content than standard indicators. Image analysis provides an invaluable tool for predicting the growth of primary cells and detecting lesion spots in the lungs.

Considering Convolutional Neural Network (CNN) architectures, the problem is finding the most suitable structure for COVID-19 recognition using Computer Tomography scans. There is a huge number of different most common CNN models that perform decently on ImageNet dataset. One of them, namely ResNet50 [6], uses residual blocks. Another one is DenseNet201, which utilizes the feed-forward fashion that is used in linear neural network [7]. The other one, namely, MobileNetV2, utilizes residual blocks and applies different fine-tuning techniques to provide mobility [8]. It is necessary to test each architecture according to its class affiliation [9]. Furthermore, in this study we try different Ensemble learning methods, and check

their performance. The problem is to find the most efficient Ensemble model for CT scans medium. This requires understanding of the models' structures and their compatibility between each other.

1.3 Proposed approach

As a starter, we will introduce the peculiarities of COVID-19 disease and ways to recognize it on time. Pneumonia that is associated with coronavirus infection (COVID-19 pneumonia) is a special type of lung lesion that more accurately reflects the term "pneumonitis". This implies the involvement of interstitial lung tissue, alveolar walls and vessels in the pathological process. That is, inflammation develops in all structures of the lungs, which are involved in gas exchange. This prevents the normal saturation of blood with oxygen. Computed tomography (CT) of the lungs is an excellent method, but it is not used to diagnose coronavirus as such, but to diagnose viral pneumonia. For many patients, fortunately, the infection proceeds without inflammation of the lungs or with minimal lung damage. As part of this research paper, we used deep learning models to identify formations on CT images. In order to conduct the study, we collected a set of the images, which will be tested using a set of specially trained CNN models for image classification. Furthermore, the ensemble learning algorithm was used to improve the results and reduce the errors, since combining two or more models leads to a better performance, thereby improving chances of correctly identifying the disease.

Chapter 2

Related work

2.1 Convolutional neural network

Reviewing the related papers, journal work of other researchers and their solutions, it was helpful to start work and gain novel ideas. The research revealed that each work was done with a closer goal but using different approaches. CNN based research papers show comparatively high results that are consistently improving. The implementation of this architecture was done on publicly available datasets. The use of such technologies is beneficial in terms of solving social problems related to medical education.

Through transmission training, it was determined whether the CNN model works effectively with new sets of images by studying its architecture (i.e. Inception-v3) by Mahbub Husain, Jordan J. Bird, and Diego R. Faria [10]. Also one of the reviewed papers proposed an individual CNN architecture for classifying sections of images of light HRCT ILD patterns [11]. According to some of them, effective methods need to be used to extract signs from ML-based healthcare systems. However, we still don't know what effective functions are, and the methods available to extract functions are not very efficient. Much ongoing research is directed to studying and classifying CT-scan and X-ray or MRI images of tumors, COVID-19, etc. Compared to other solutions, these demonstrate a decent track record of achieving high classification results. For instance, according to Narayan Das et al. [12], the ResNet50 architecture

achieves the accuracy score of 92%, and VGG architecture achieves the value of the accuracy of 90%. Furthermore, Parnian Afshar et al. [13] proposed the solution that works with the smaller datasets. The approach was based on Deep Learning and achieved high accuracy scores.

Yana Sun et al. propose an automatic method for designing CNN architecture using genetic algorithms to efficiently solve image classification problems [14]. The article discusses the advantage of the proposed algorithm, that lies in its “automatic” characteristic, which implies that users do not need knowledge of the CNN domain when using the proposed algorithm, while they can still get a promising CNN architecture for given images [14].

Q. Li et al. proposed a customized CNN architecture to classify HRCT lung image patches of WILD patterns [11]. The results of this work showed that the described design is able to automatically extract distinctive features without manual work, thus this method can achieve good performance. They also stated that a properly designed network structure and similar methods such as intensive screening and distortion of input data that eliminate the problem of overfitting, effectively solved these problems.

Mohammad Sajjad et al. also conducted a research on classification of the brain tumors. The authors used deep CNN with extensive data expansion. Across all the models that were presented in this work, CNN used augmented data to fine-tune its classification of brain tumor grades [15].

Paulo Lacerda et al. used a Hyperband optimization algorithm in the optimization process, CNN turned to the diagnosis of the disease SARS-Cov 2 (COVID-19). They used the Options framework. Cnn models were trained on 2175 computed tomography (CT) images. As a result, they propose a CNN model, which is VGG 16 with five initial modules and 128 neurons in two fully connected layers. Their proposed model achieved 82% results [16].

2.2 Ensemble Machine Learning Algorithms

Giorgio Giacinto and Fabio Roza researchers with their work proposed an approach related to ensemble learning, namely, the authors talked about the design of effective ensembles of neural networks. A large set of neural networks was used, and the presented approach was aimed at selecting a subset of networks that were most error-independent. Thus, the authors achieved effective results [17].

Byoungchul Ko et al. [18] provided another method that is based on the ensemble algorithms and random forest classifier. They also proved that the algorithm they used was more efficient compared to other classification methods when using training datasets.

R. Lavanyadevi et al. [19] proposed a method that includes mechanically recognizing semantically meaningful areas in an image. The researchers were able to detect malignancy and make diagnoses by correlating every pixel in the image and the sticker that signifies a semantically meaningful part. The features of neighboring double examples and gray level co-occurrences are removed from brain images with benign or malignant or normal images.

Zi Wei Zhi et al. [20] developed a diagnostic prototype system for COVID disease that has been put forward. Their proposed model has been tested and trained on various datasets. CT scans from uninfected people were used. The results of computed tomography accuracy, which was 95.8 percent, were also described extensively.

In another article, Ying Bi et al. offer a training system that is automated using an ensemble algorithm and applying GP (EGP) for image classification. The method presented in this paper combines not only the study of the features, but also the selection and training of the classifier function, as well as the combination into a single program tree. The results of the work were positive, as EGP provided better performance, thereby improving ensembles. The authors of this paper were the first researchers to use automatic ensemble creation and GP (EGP) [21].

Hasan Rusel, Seule Yasar, and Kemal Kolak conducted a study that was aimed at web development, specifically at the development of free software which purpose

is to diagnose and detect brain tumors [22]. The results of the software, as it has demonstrated its competitiveness and reliability in the identification and diagnosis of three types of brain tumors. The authors also provided a link to their web software that is available in 2 languages.

In the "Diagnosis of nodular formations in the lungs based on computed tomography images based on ensemble training", a classification method using computed tomography was proposed [23]. To evaluate the performance of the proposed system, they used 60 computerized tomography (CT) scans assembled by the Lung Image Database Consortium (LIDC), and as a result, all these techniques has shown an improvement in diagnostics.

Xiaobo Li et al. in their article developed a system that worked even when training data was insufficient [24]. The system is based on the ensemble transmission to improve the classification accuracy. The authors also proposed a weighted sampling method for transmission training, which was called TrResampling and used the TrAdaBoost algorithm. The algorithm is used to adjust the weights of the source data and the target data.

Chapter 3

Methodology

3.1 Datasets

Computed tomography is a highly modified X-ray, which represents a computerized imaging device mounted inside of rotating "bagel". The process of computed tomography includes shifting the table on which the patient is located in an anterior-posterior direction, while the tube and detector rotate around the table so that the patient is constantly between them. Since the patient is constantly moving back and forth, the trajectory described by the X-ray beam on the patient takes a form of spiral [25]. The detector gets several thousand projections of each cross-section of the body at different times. After that, with the help of special logarithms of computer processing, and reconstruction, we get a three-dimensional data set: a set of very thin cross-sections of the human body, from which we can then arbitrarily rebuild any other planes. The following are examples of what the lungs look like on a CT-scan of a coronavirus CT infection. In patients with inflammation caused by coronavirus infection, tomograms show a characteristic radiological sign, which can be viewed in Figure 3-1.

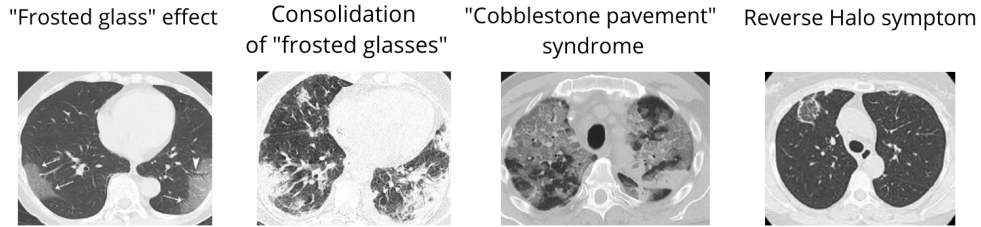


Figure 3-1: Characteristic of radiological signs of coronavirus infection.

For this study three open-source datasets were used. These datasets encouraged us, as novice researchers, to apply our proposed technology in the field of image classification to achieve new opportunities in the fight against this infectious disease. The SARS-CoV-2 dataset, for comparison purposes, contains 1252 CT scans that are positive for SARS-CoV-2 infection (COVID-19) and 1230 CT scans for patient populations who are not infected with SARS-CoV-2, for a total of 2482 CT scans 3-2. The information was gathered from genuine patients at Sao Paulo hospitals, and the goal of this data collection is to encourage artificial intelligence research and development for determining whether a patient is diagnosed with SARS-CoV-2 by monitoring his or her computed tomography scans [26]. Images from the COVID-CT Sao Paulo, Brazil dataset are displayed in Figure 3-2 and appear in the first row while non-Covid-19 cases appear in the second row. COVID-CT dataset composition is shown in Table 3.1.

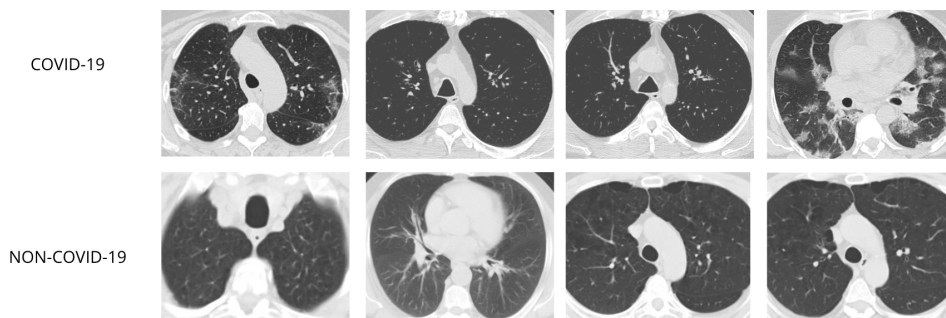


Figure 3-2: COVID-CT images from the SARS-COV-2 CT-Scan Dataset Covid-19 cases.

Table 3.1: SARS-COV-2 CT-Scan Dataset.

Type	<i>Non-Covid-19</i>	<i>Covid-19</i>	<i>Total</i>
Train	988	749	1737
Validation	200	173	373
Test	199	173	372

UCSD COVID-CT, the second dataset, has the lowest average number of samples above all the others. The data was in TXT file format and in separate folders with png images, which was quite convenient to use [27]. The repository also has meta-information which consists of patient ID, patient information, Gender, image caption, and age. All images in this dataset were gathered from COVID19-related papers from medRxiv, bioRxiv, NEJM, JAMA, Lancet, etc [28]. Examples of data are listed on Figure 3-3 and Table 3.2.

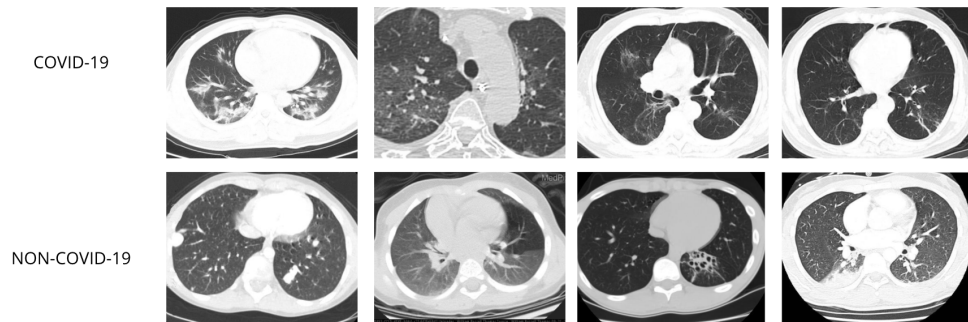


Figure 3-3: COVID-CT images from the UCSD COVID-CT Dataset Covid-19 cases.

Table 3.2: UCSD COVID-CT Dataset.

Type	<i>Non-Covid-19</i>	<i>Covid-19</i>	<i>Total</i>
Train	177	120	297
Validation	40	25	65
Test	39	24	63

The third dataset, COVIDx-CT Dataset, contains volumetric chest CT scans and a comparison CT image dataset derived from CT imaging data collected by the China National Center for Bioinformation, with 104,009 images from 1,489 patient cases [29]. COVIDx CT data structure is separated into two variations: A and B. The first variant consists of cases with confirmed diagnoses, while the second variant includes the entirety of the first option as well as some cases that are supposed to be correctly diagnosed but have been poorly verified [30]. More than ten people were involved in the data collection process. The dataset allocates a large amount of memory since there was a lot of excess in it. During the beginning of the virus, datasets were updated with the addition of new CT images of patients. The sample images and dataset analysis are provided in Figure 3-4 and Table 3.3.

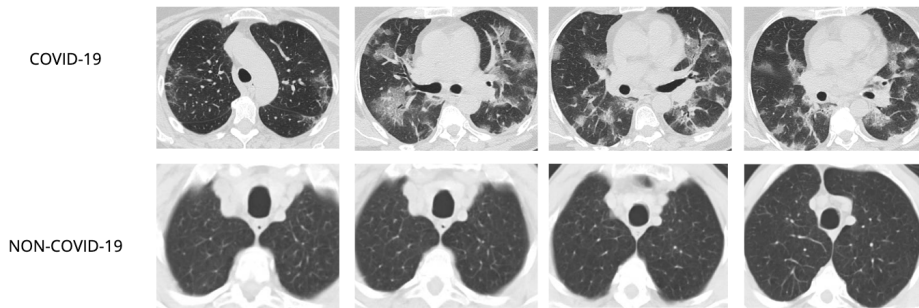


Figure 3-4: COVID-CT images from the COVID-CT dataset Covid-19 cases.

Table 3.3: COVIDx-CT Dataset.

Type	<i>Non-Covid-19</i>	<i>Covid-19</i>	<i>Total</i>
Train	27201	12520	39721
Validation	9107	4529	13636
Test	9450	4346	13796

3.2 Ensemble learning algorithm

In this chapter, we consider the Ensemble learning algorithm as well as its methods. The ensemble algorithm is one of the decently performing methods in machine learning. In the study, we used the algorithm to classify medical images, specifically COVID-CT datasets. Speaking of how the algorithm works, it combines several models to get the better result and performance, which are the most significant aspects of the identification of different diseases. In our work, we were convinced that ensemble classification models can be a powerful machine learning tool. The main value is that it can eliminate potential errors made by any individual classifier, thereby improving the performance of the programmatic model (Fig. 3-5).

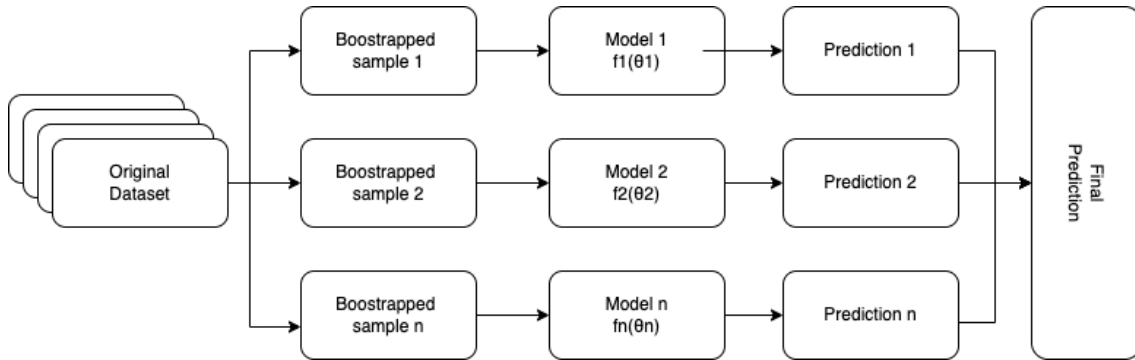


Figure 3-5: Ensemble learning methods structure.

Ensemble learning performs better than the individual models it comprises, be-

cause it gives us the error of each prediction. Thus, when the algorithm combines two predictions, it improves the indications and reduces the number of mistakes [31]. There are three most popular methods for the combination of different models' forecasts: Bagging, Stacking and Boosting.

One of the methods we consider in the study is stacking the features extracted by several neural networks. First, we took multiple pretrained Convolutional Neural Network models and fine-tuned their last convolutional layers using 3 of the described datasets. For the fine-tuning process we used the Categorical Cross-Entropy loss function. It basically calculates the difference between the actual and predicted likelihoods. The ideal value of the loss is 0. The formula for the latter is as follows:

$$Loss = - \sum_{i=1}^{outputsize} y_i \cdot \log(\hat{y}_i) \quad (1)$$

Furthermore, another important part of the fine-tuning is freezing the dense layers to fine-tune only convolution blocks. This was done using Keras built-in functionalities, such as the attribute trainable of the keras.layers. By making the dense layers "non-trainable", we can fine-tune only the convolutional architecture of the model. This was done so that only pre-trained part of the model was tuned using the CT scan datasets. If we fine-tune the models with the unfrozen dense layers, this will result in the large gradient shifts and in the elimination of the pre-trained features of the architectures we use.

We then used the models to extract features from the images of the datasets. The features were then stacked into another so-called dataset. To classify the features that were extracted by the models, we used one of the most efficient linear classifiers, namely Logistic Regression. The equation for the logistic regression, where p is the probability of correct prediction, is as follows:

$$y = \log\left(\frac{p}{1-p}\right) \quad (2)$$

Beforehand, we used Global Average Pooling to transform 2 dimensional features into 1 dimensional data. The reason of why the ensemble method works is that it diminishes the losses of individual network, which results in higher evaluation accuracy.

Then, we also utilized Support Vectors Machines (SVM) algorithm instead of Logistic Regression. What SVM does is mapping the input data into the points in n-dimensional space, with n as a number of features. Then, it solves the classification problem and finds the hyper-plane that separates the data into several classes. The algorithm is capable of both classification and regression, however, in the task we used Support Vectors Classifier, namely, LinearSVC. Other SVM classifiers include SVC and NuSVC that utilize "one-versus-one" multi-class strategy, while LinearSVC uses "one-vs-the-rest" way of multiple class classification. The method utilizes hinge loss function to solve the main problem, where the hinge loss for a given input x in its simplest form (Equation 3.1):

$$loss(y) = max(0, 1 - y \cdot x) \tag{3.1}$$

For the given SVM problem, the hinge loss is as follows:

$$hingeloss = \sum_{i=1} max(0, 1 - y_i(w^T \phi(x_i) + b)) \tag{3.2}$$

In Equation 3.2, ϕ stands for identity function. The main goal of the SVM is to minimize the weights parameters. In combination with the hinge loss, the formulation of the main problem is as follows:

$$min_{w,b} - \frac{1}{2}w^T w + C \sum_{i=1} max(0, 1 - y_i(w^T \phi(x_i) + b)) \tag{3.3}$$

C in the Equation 3.3 stands for the penalty term that regulates the penalty for allowing some points to be located at some distance from the boundary.

Bagging - helps us to reduce the spread in datasets. The technique significantly impacts the reduction of correlation. Eventually, when compared to a single decision

tree, the average value of all predictions from several trees is superior and unbiased.

Boosting - as we said earlier, the algorithms work with several models simultaneously. The application of boosting improves the model's evaluation results. There are several instances of the meta-algorithms that utilize boosting: AdaBoost, gradient boosting, and others.

Another Ensemble method we utilized is called Weighted Average Ensemble. The Weighted Ensemble approach is an extended model averaging method. Model averaging is the type of ensemble learning, where each individual neural network architecture contributes to the resultant prediction in the same way as other individual models. In a Weighted Ensemble approach, however, the individual weights are assigned to each model depending on its performance (see the equation below). In this way, the model that performs with a better accuracy will get the higher weight. The sum of the weights should be equal to one.

$$WeightedAverage = \sum_{i=1}^{models} A_i \cdot w_i \quad (3)$$

3.3 Models

Convolutional Neural Network

When working on my thesis, we were often asked why we use CNN models specifically. Below we will try to describe the reason in details. The task is to create CNN models to classify medical images and achieve high accuracy results. The image classification problem is to receive the initial image and to output its class (covid, non-covid, etc.) or a list of the probabilities for each class [32]. The convolutional neural network architectures fit the problem of image classification perfectly, which is why the method was used. Below we will describe each CNN model that we use [33]. The reason for utilization of these models lies in the fact that they successfully perform with the similar domain providing promising results. In this study, those models have provided very competitive results outreaching all the initially set expectations.

Convolutional Neural Networks is needed to analyze tiny data features in order to obtain a larger perspective. It is evident from recent study that CNN performs best in terms of image classification abilities.

VGG-19

VGG-19 is a CNN architecture that that comprises 19 levels of different Neural Network layers. It is one of the most popular convolutional neural networks, which is simple and practical enough and can perform with the state-of-the-art efficiency [13]. VGG-19 model was pretrained on vast numbers of sample images and implements the architectural style that combines Image regularization, Convolution, ReLU, and Max Pooling. Researchers used kernels with the stride size of one pixel, which allowed them to cover the entire image concept. VGG can be assumed as a more in-depth version of AlexNet. The system is made up of convolutional and fully connected layers help achieve a high model accuracy classification [34].

MobileNetV2

MobileNetV2 is a CNN model which was designed for mobile applications. Sandler et al. modified the previous version of the MobileNet to provide a better performance and efficiency that are capable of competing with other novel convolutional neural network architectures [8]. The architecture is especially compatible with the frameworks that were also created by the authors. One of them is SSDLite, which was created to efficiently utilize the MobileNet models to recognize objects. The baseline that was used for the development of the MobileNetV2 architecture was an inverted residual structure. The principle behind it is that they used bottleneck layers as input and output layers of the residual block. The structure of the MobileNetV2 model comprises, first, convolution layer that contains 32 filters and, second, 19 residual bottleneck layers [8]. Model a very effective feature extraction for object detection and segmentation and also faster in performance and are useful for mobile applications.

VGG-16

The VGG-16 architecture is a Convolutional Neural Network model that was presented by Simonyan and Zisserman [35]. The architecture shows one of the best performances on the ImageNet dataset which comprises 1000 classes of 14 million

images. VGG16 achieved the accuracy score of 92.7% that was one of the 5 highest results on the dataset. The architecture is somewhat similar to the AlexNet model's structure. However, to improve the model in terms of efficiency, the authors replaced large kernel-sized filters with 3 3x3 kernel-sized filters, where each is subsequent to another. The model consists of 13 convolutional layers, 3 dense layers, and 5 pooling layers [35]. It is also built in the Keras application library, where it can be accessed as pretrained on the ImageNet dataset. Moreover, VGG model outperformed other models with 92.7 and showed top-5 test accuracy, and won 1st and 2nd place in the 2014 ILSVRC competition.

ResNet-50

The ResNet50 architecture is a Deep Residual Neural Network. That is, the architecture of the model was also created by stacking several residual blocks. The ResNet convolutional neural network models are all based on the same principle. The authors used the VGG CNN architecture as a baseline for creating the residual analogue [6]. The ResNet-50 architecture, however, has one significant difference in the structure. To optimize the training process, the authors transformed the building block into a bottleneck. As the name suggests, the ResNet50 model comprises 50 neural network layers that provide one of the highest accuracies among other state-of-the-art architectures [6]. The use of this model is positive for saving computing resources and training time to develop this work.

InceptionV3

One of the main goals of the Inception architectures is coping with two issues related to Convolutional Neural Network architecture. One of them is increasing the depth of the model and improving its performance. The Inception architecture was initially planned as a baseline for the GoogLeNet Convolutional Neural Network model. The Inception models utilize the "Inception modules" that comprises convolutional filters. One of the benefits of the InceptionV3 CNN architecture is that the width of every stage and the total stages number can be modified [36].

Xception

The Xception is a Convolutional Neural Network architecture that utilizes regu-

lar convolution layers, Inception modules, and the depthwise separable convolution operation. The author [37] used the Inception modules that were described above for separation of the regular convolution and the depthwise separable convolution. The latter is substantially an Inception module, however, it consist of the largest possible number of towers. The overall structure comprises the depthwise separable convolution blocks that each followed by MaxPooling layer, which are linked using shortcuts that are used in the ResNet architectures [37]. Thus, due to its structure, this architecture is very easily defined and modified.

DenseNet201

The Dense Convolutional Network (DenseNet) architecture is another popular CNN model. It uses a feed-forward connecting method, which is used in the linear neural networks [7]. That is, the method was basically created to link the dense layers between each other. The architecture of the DenseNet models utilizes several dense blocks, which combines a number of convolution layers that are connected in a feed-forward fashion. The structure of the DenseNet201 is as follows: a regular convolution layer, pooling layer, 4 Dense blocks, each of which has 2 convolutional layers, that are linked using a one by one sized convolution layer and an average pooling layer, and classification layer, which consists of global average pooling and fully-connected dense layer. Each dense block contains different number of the pair of one by one and three by three convolution layers. Although, the layer has a large number of parameters, it provides one of the highest accuracies and one of the best performances for image classification [7]. It offers a totally different architecture comparing to the other ones, it has dense block where the convolutional layers are connected in a feed forward fashion.

EfficientNet-B3

The model is one of the most powerful CNN models, when comparing to ResNet and other popular CNN models. As a result, demonstrating greater accuracy and efficiency [38]. The architecture of EfficientNet-B3 model is displayed on the Fig. 3-6 below. Efficient models follow the same general structure as the other popular image recognition models and contain the blocks of convolutional layers followed by fully connected layers. The models commonly have 7 blocks with sub-blocks that consist of Memristive Binary Convolution layers (MBCConv) with skip connections which improves recognition capability and optimizing training. EfficientNet-B3 has 12 million parameters and is one of the smallest models used in our training. However, its MBCConv layers and structural simplicity provide one of the best accuracy results with little training time.

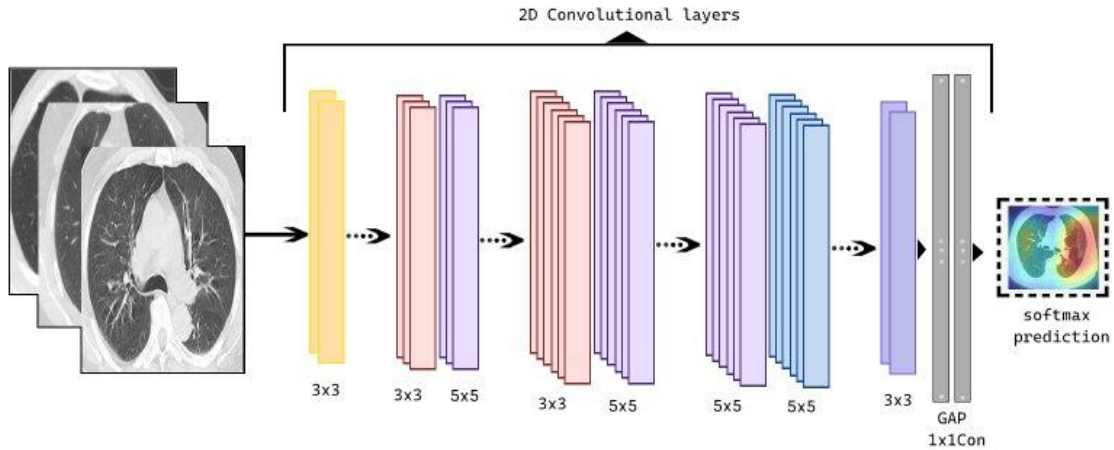


Figure 3-6: Architecture of EfficientNet-B3 model.

3.4 Training

First, the publicly available datasets that we utilized to train our Convolutional Neural Network architectures are all differently sized. The UCSD and CovidX CT databases contain images from various sources. This implies that the images have different resolution, rotation angle, and brightness. This allows us to find the most suitable structure for meta-learner and the optimal strategy for fine-tuning.

The first step before training was the use of fundamental pre-processing. Each image was aligned, cropped, and regularized in terms of brightness. Although this added the overall consistency to bigger datasets, pre-processing of smaller ones might lead to the data corruption and provide inconsistent images to the models, which eventually results in the lower performance. This implies that we need to thoroughly consider the selection of meta-learner block and the parameters of its layers [39]. For instance, when there is not enough data the larger amount of features might result in overfitting [40]. Furthermore, the wideness of the layers provide almost the same efficiency as the deepness of a series of layers [6].

Table 3.4: Computational time for each architecture.

Architecture	Batch size	Epoch	Total comp.t(s)
VGG-16	32	150	890
VGG-19	32	150	900
MobileNetV2	32	150	720
ResNet-50	32	150	960
InceptionV3	32	150	808.5
Xception	32	150	1395
DenseNet201	32	150	945
EfficientNet	32	150	1050

In the Table 3.4 the computational time complexity for each architecture is represented. The time complexity represents the speed, with which the algorithm performs on some input. One can notice that the input parameters for training purposes were the same including batch size and number of epochs. The total time complexity of the CNN architecture depends on several factors, including its deepness, wideness, com-

plexity, number of parameters, and others. The results suggested that MobileNetV2 performed the fastest. This was because the model had been designed for mobile devices that did not possess enough of computational power. Thus, MobileNetV2 was less complex than others. We can also see that the most of other CNN architectures performed almost with the same speed with time complexity ranging from 808.5 to 1050 seconds. The slowest model was Xception. This implies that the architecture was either the most complex or utilized the largest number of parameters.

Nevertheless, both varieties require strong regularization methods. In this study we used L1 and L2 regularization to regularize parameters through penalties. Penalties of higher values were applied on the first layer of the meta-learner block, and penalties that were ranging from 0.3 to 0.5 were applied to dropout layers. Heckel and Yilmaz [41] proposed early stopping solution to solve the double descent problem. However, for this study we trained our CNN models for 100-150 epochs on a par. According to Nakkiran et al. [42], this ensures consistency and averts gradients' explosion.

Transfer learning is an effective method that is a subsection of machine learning, the purpose of which is to apply knowledge obtained from one task to another target task. As we know, in deep learning there are two common strategies for Transfer learning namely: function extraction and fine tuning [43]. The first option implies that only the weights of some newly added layers are optimized during training, while in the second option all the weights are optimized for a new task. To do this, we used the built-in Keras functionalities, such as trainable attribute of the keras.layers. Using the attribute, we can upload the pre-trained CNN models as head and make it as "non-trainable". Then we add the dense layers and make them "trainable". The attribute accepts the Boolean values. We established that fine-tuning performs more successfully than feature extraction. As the CT datasets we have used contain a number of images with different spatial sizes, they need to be resized beforehand in order to guarantee their compatibility with the model input size.

3.5 Evaluation

The convolutional neural network architectures we used are simple, use memory efficiently, and scale to differently sized and formatted databases. According to our results, VGG19 and DenseNet201 are the most efficient and provide the best performance with regard to accuracy and loss. The structures of the models' meta-learner block's layer were based on the same foundation, which might suggest that the models could be potentially united in a single more efficient model. Furthermore, t-SNE results suggested that the models managed to correctly extract the features from the CT images and achieved the highest accuracy among other models.

During the training process, we also noticed some patterns that resulted in the overall architectures' performances. If we consider the randomly selected training data, the performances of the architectures were consistent and similar. However, some datasets tend to have the CT scan images taken from a single patient. This implies that the model could be overfitted with the concrete instance from the large dataset. The solution for this is utilizing proper filtering of the training set.

Another part of the study was to evaluate the methods we utilized, such as different data augmentation and pre-processing, using appropriate metrics. The metrics that were used in the work are sensitivity, specificity, and f1 score. The metrics touch upon many different aspects of the datasets, including their inconsistencies. This implies that these are one of the most objective indicators of success. The values of sensitivity, specificity, and f1 score also suggest that VGG19 and DenseNet201 are most accurate and best performing architectures.

Chapter 4

Experiments

4.1 Experimental Setup

This section describes the tools that were used during experiments and evaluation indicators and also analyzes the settings of hyperparameters which will be described in detail below.

4.1.1 Software

The experiments that were done for this study were carried out with the help of certain tools. The code was written using the latest version of Python, Keras libraries, and TensorFlow frameworks. All the necessary tools were installed on a computer with a GeForce GTX 970 GPU that is a highly performing graphics card.

4.1.2 Evaluation Metrics

In machine learning tasks, metrics are used to evaluate the quality of models and compare various algorithms. The work presents metrics that are commonly used for evaluation in various studies [44]. Before describing the metrics, it is necessary to understand each concept. To describe metrics in terms of classification errors, we used confusion matrix [45]. The size of the confusion matrix is N by N , where N is the number of classes. The well-known indicators that utilize the matrix were used

to illustrate the characteristics of data in statistical results.

Sensitivity - is an indicator that is a proportion of correctly classified positive observations. Therefore, the higher the sensitivity, the better performance on the positive instances is provided by the classifier.

$$Sensitivity = \frac{TP}{TP + FN} \quad (4)$$

Specificity - is an indicator that refers to a proportion of true-negative classifications in the total number of negative observations.

$$Specificity = \frac{TN}{TN + FP} \quad (5)$$

F-score - is a metric that combines information about the accuracy and completeness of the algorithm. The indicator helps us evaluate the performance of the convolutional neural network model in terms of binary classification.

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

Accuracy - is an indicator that depicts the proportion of the correct answers provided by our algorithm models.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

4.2 Comparison with Other Models

CNN models are the most common solution for the range of image classification tasks. To understand the difference between the models, let's start with the LeNet-

5 model. LeNet-5 consists of 7 Neural Network layers: 3 convolutional layers, 2 subsample layers, and 2 fully connected layers.

AlexNet is made up of 8 levels [46]: 5 convolutional layers, 2 fully connected layers, and a dense output layer. Comparing to LeNet-5, AlexNet has a much larger architecture. As a result of the need in reducing the number of parameters in the CONV layers and in optimizing the training process, **VGGNet** was developed. Simonyan and Zisserman [35] used deeper configuration of AlexNet [46] and they proposed it as VGGNet.

4.3 Comparison among other ensemble algorithms

Ensemble learning is a well-known machine learning algorithm. Analyzing the sources, it became clear to us that the algorithms play an invaluable role, since they are flexible and can be applied to a variety of purposes. Regression and classification tasks benefit from ensemble estimation because they reduce bias and variance, thereby improving model performance.

A powerful *Adaptive Boosting Algorithm*, AdaBoost, arranges a sequence of weak classifiers such that the weakest classifier at each point is the optimal option for correcting the errors introduced by the previous classifier. Boosting is the process of arranging weak classifiers in such a way that the best choice for each weak classifier corrects errors that have been made by the previous classifier.

Random Forest Algorithm: A random forest is one of the tree-based machine learning algorithms that combine the abilities to solve several decision trees. To predict the outcome of such a tree solution, each node works with a random subset of functions. In the final stages, a random forest collects all the results obtained, integrates the results of individual decision trees, and evaluates outputs the final result.

Histogram-based Gradient Boosting: In general, the histogram-based Gradient Boosting algorithm is similar to the Gradient Boosting algorithm, with the exception that it's compatible with the dataset.

4.4 Results

4.4.1 Performance comparison of the proposed CNN models

In this investigation, 8 models were estimated on 3 different datasets. As a result of the experiments, VGG-19 outperformed other models with a mean accuracy of 95.3%. The results suggest that we might apply this model or its variation in practical medical application or future classification works on COVID-19.

We present summary of results of the EfficientNet-B3 model on test dataset in the confusion matrix 4-1 below. Predicting correctly 2.4 thousand samples it achieved the overall accuracy of 76% on test samples. However, the model tended to mark samples of COVID-19 infected as negative. Possible reason for this behaviour is non-uniform distribution of samples over classes of about 15% difference. Change of the model's meta-learner block or unfreezing last convolution layers might help the model capture more features without changing dataset for training.

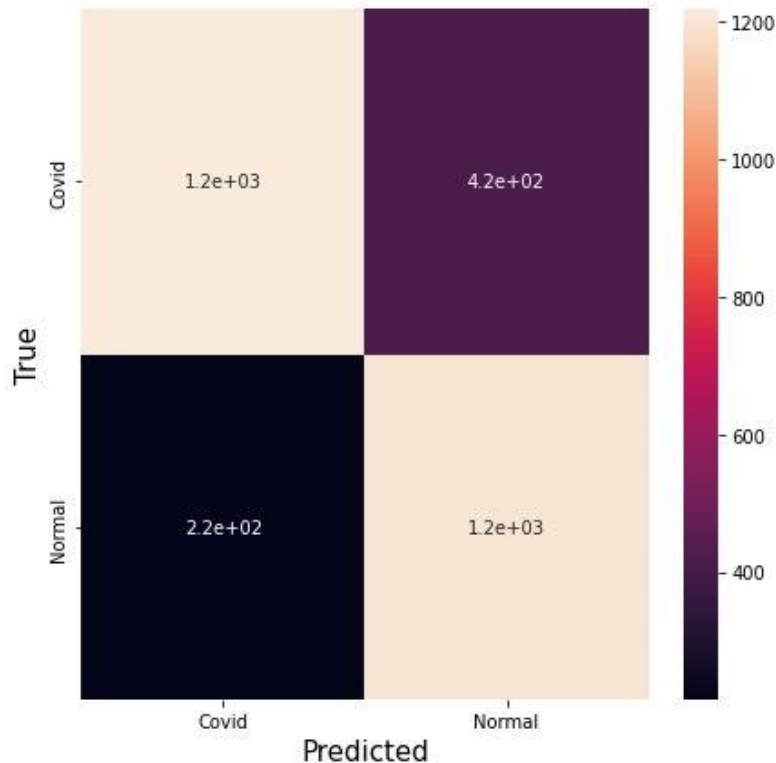


Figure 4-1: Confusion matrix for EfficientNet-B3.

We included all the relevant metrics' results (f-score, accuracy, specificity, sensitivity) in tables 4.1-4.3. VGG-19 and DenseNet201 show slightly better overall performance than the rest of the models (VGG-16, Xception, ResNet50V2, MobileNetV2) with an average of 96.76% and 98.1% accordingly. The observation of the results that are presented in the table shows that DenseNet201 and VGG19 are more universally applicable and provide with the better performances. The reasons for their superiority are their depth and concatenation methodologies.

Table 4.1: Performance comparison of the proposed CNN models for CovidX CT dataset.

Models	acc	sensitivity	spec	f-score
ResNet50v2	0.96	0.95	0.96	0.97
VGG-16	0.97	0.98	0.96	0.97
VGG-19	0.95	0.95	0.95	0.96
InceptionV3	0.97	0.96	0.97	0.97
Xception	0.94	0.94	0.94	0.95
MobileNetV2	0.96	0.97	0.96	0.95
DenseNet201	0.97	0.97	0.97	0.96
Efficientnet-b3	0.80	0.77	0.85	0.81

Table 4.2: Performance comparison of the proposed CNN models for SARS-CoV-2 CT dataset.

Models	acc	sensitivity	spec	f-score
ResNet50v2	0.95	0.95	0.96	0.95
VGG-16	0.95	0.93	0.96	0.98
VGG-19	0.97	0.97	0.96	0.98
InceptionV3	0.96	0.95	0.95	0.97
Xception	0.93	0.93	0.93	0.95
MobileNetV2	0.94	0.93	0.94	0.94
DenseNet201	0.97	0.96	0.97	0.98
Efficientnet-b3	0.76	0.74	0.85	0.79

Table 4.3: Performance comparison of the proposed CNN models for UCSD COVID-CT dataset.

Models	acc	sensitivity	spec	f-score
ResNet50v2	0.88	0.88	0.89	0.85
VGG-16	0.92	0.92	0.92	0.92
VGG-19	0.94	0.91	0.98	0.97
InceptionV3	0.92	0.92	0.92	0.93
Xception	0.77	0.76	0.77	0.79
MobileNetV2	0.88	0.88	0.88	0.89
DenseNet201	0.91	0.92	0.91	0.92
Efficientnet-b3	0.94	0.92	0.92	0.93

4.4.2 Performance of the Ensemble Learning algorithm

For the evaluation of the performance of the Ensemble Learning algorithm we used 7 of our Convolutional Neural Network architectures and 3 CT datasets. First, considering the results obtained using SARS-CoV-2 CT dataset, the best accuracy was achieved by the MobileNetV2 CNN architecture with the value of 0.9850. The mean accuracy score for all 7 architecture was 0.945. The resultant ensemble model accuracy was 0.9900. The values of other accuracies are displayed in the Table 4.4.

Second, let us consider the results obtained using USCD CT dataset. The highest accuracy was achieved by VGG19 architecture with the value of 0.9450. The lowest accuracy of 0.7743 was provided by Xception model. The mean accuracy score for all

Table 4.4: Performance of Ensemble models for SARS-CoV-2 CT dataset.

Architecture	Ensemble model	
	acc	Ensemble model acc
InceptionV3	0.92000	0.990
VGG16	0.94500	
VGG19	0.91250	
MobileNetV2	0.98500	
Xception	0.92000	
ResNet50	0.96250	
DenseNet201	0.97250	

7 architecture was 0.8930. The resultant ensemble model accuracy was 0.9867. The values of other accuracies are displayed in the Table 4.5.

Lastly, we need to consider the results obtained using COVID-X dataset. The highest accuracy was achieved by VGG19 architecture with the value of 0.9630. The lowest accuracy of 0.8860 was provided by Xception model. The mean accuracy score for all 7 architecture was 0.9311. The resultant ensemble model accuracy was 0.9777. The values of other accuracies are displayed in the Table 4.6.

Each of the above described results suggest that the ensemble model performs better than the each individual Convolutional Neural Network architecture. When the models were trained and evaluated on the SARS-CoV-2 CT dataset, the obtained accuracy values were the highest for individual and Ensemble models. The highest difference, however, between the Ensemble model’s accuracy and the highest individual model’s accuracy was seen when the architectures were trained on USCD CT dataset. The lowest Ensemble model’s accuracy value was provided when the architectures were trained and evaluated on the COVID-X dataset.

Table 4.5: Performance of Ensemble models for USCD CT dataset.

Architecture	Ensemble model	
	acc	Ensemble model acc
VGG19	0.9450	0.9867
VGG16	0.9230	
InceptionV3	0.9230	
Xception	0.7743	
MobileNetV2	0.8865	
DenseNet201	0.9130	
ResNet50v2	0.8860	

Table 4.7 represents results of ensemble learning with SVM for three different opensource datasets. From the below table we can see the achieved results was established that these results outperform recent state-of-the-art research findings. For the UCSD COVID-CT dataset results showed 0.9643, for SARS-COV-2 CT 0.9820 and for the last dataset COVIDX CT achieved results was 0.9943.

Table 4.6: Performance of Ensemble models for COVID-X dataset.

Architecture	Ensemble model	
	acc	Ensemble model acc
VGG19	0.9630	0.9777
VGG16	0.9400	
InceptionV3	0.9430	
Xception	0.8860	
MobileNetV2	0.9065	
DenseNet201	0.9430	
ResNet50v2	0.9360	

Table 4.7: Performance of Ensemble with SVM.

Ensemble model –fine tuning + SVM				
Dataset	accuracy	sensitivity	specificity	f-score
UCSD COVID-CT	0.9643	0.9751	0.9544	0.9754
SARS-COV-2 CT	0.9820	0.9901	0.9888	0.9883
COVIDX CT	0.9943	0.9933	0.9912	0.9925

The details of the image subsets used here are given in Table 4.8. It should be noted that the test set classification count of the proposed approach for 2 different dataset with multiclass ResNet50, VGG-19, Efficientnet-b3 by images.

Table 4.8: Distribution of experimental data.

	Training images	Validation images	Testing images
ResNet50v2	1543	700	711
VGG-19	1543	711	800
Efficientnet-b3	1543	755	800

4.4.3 t-SNE (t-distributed stochastic neighbor embedding)

To add a transparency to the architectures we utilized, we used t-Distributed Stochastic Neighbor Embedding (t-SNE). t-SNE is based on the conversion of the resemblances of data instances to the probabilities. It furthermore relies on the minimization of the discrepancy within low- and high-dimensional set of instances. The t-SNE visualization method is commonly used to reduce the dimensionality and for displaying multidimensional databases.

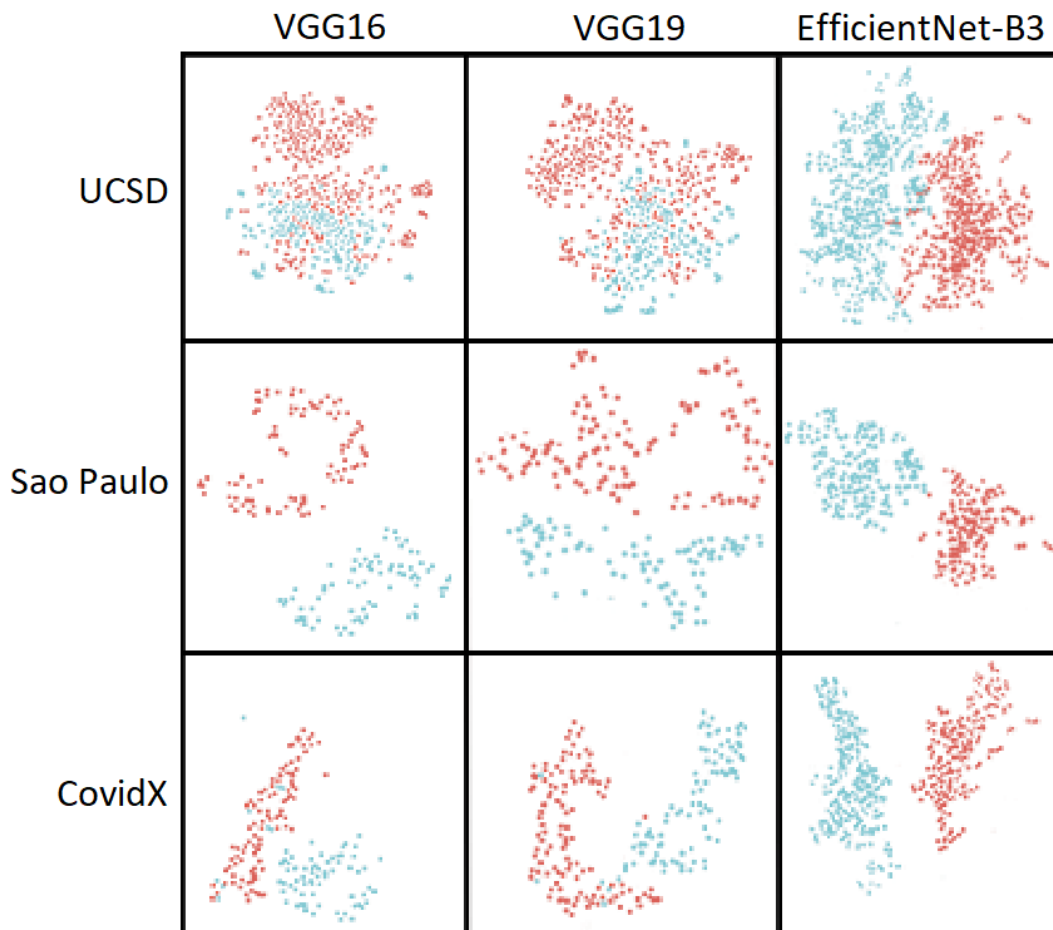


Figure 4-2: t-SNE for VGG16, VGG19 and EfficientNet-B3 for 3 datasets.

Note: t-distributed stochastic neighbor embedding for UCSD, Sao Paulo, CovidX CT dataset. Presented models: VGG16, VGG19 and EfficientNet-B3.

Figure 4-2 display 2 dimensional visualization. t-SNE was implemented for 3 dataset namely SARS-CoV-2 CT scan dataset, CovidX CT and UCSD COVID-CT. First, as one can notice, each individual CNN model performed well on SARC-CoV-2 dataset. The two classes were accurately separated into two clusters, which suggest that the models interpret the data correctly.

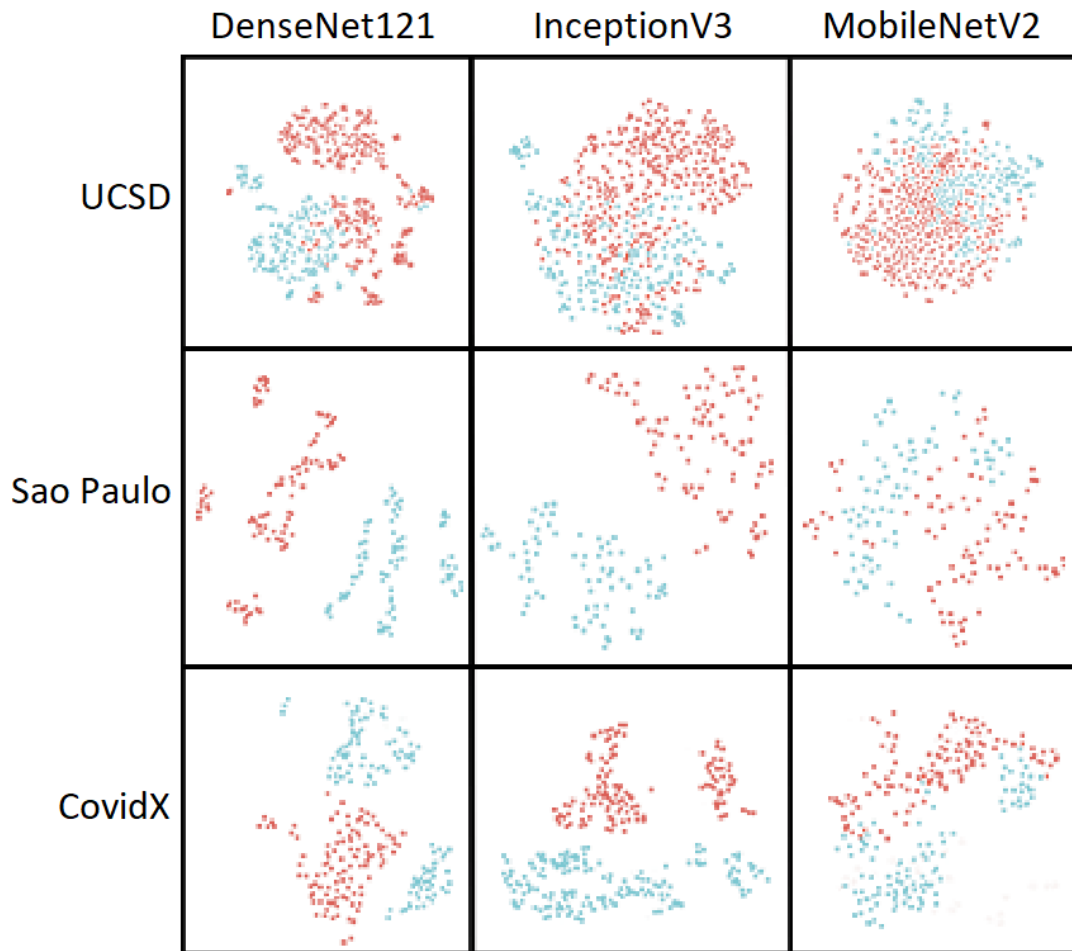


Figure 4-3: t-SNE for DenseNet201, InceptionV3 and MobileNetV2 for 3 datasets.

Note: t-distributed stochastic neighbor embedding for UCSD, Sao Paulo, CovidX CT dataset. Presented models: DenseNet201, InceptionV3 and MobileNetV2.

The least accurate result was provided by MobileNet. Although, it divided the data into two clusters, it confused some of the samples and did not create a distinct

boundary. Second, considering CovidX CT dataset, the results are less accurate than in the previous one. The least accurate was DenseNet, as it did not manage to separate the data into two different clusters. However, other models provided a precise separation of the dataset. Lastly, considering UCSD COVID-CT database, the resultant 2D visualization is the most precise. Each model accurately separated the data into two distinct clusters. There are only a few inaccuracies in MobileNet's t-SNE. Other than that, the results are very precise (Fig. 4-3).

4.4.4 Heatmap visualization

This section explains how to extract and assess regions with significant differential activation between two classes of samples using a CNN model and Grad-CAM algorithm [47]. We demonstrate the applicability of the algorithm to a variety of medical imaging problems with varying information resolution, as well as its performance in a single dimensional and multidimensional data environments.

The Figure 4-4 illustrates a 2D representation of the Grad-CAM heatmap evaluated on a CT scan of an individual patient with highlighted disease spots on the lung region. Grad-CAM is one of the class activation mapping (CAM) approaches that is employed in this study. The CAM framework allows us to combine data from a variety of sources to create an accurate demonstration of NN models using individual examples from each class and dataset. The color palette of the data is seen as a heatmap [48]. The framework is decently adaptable to emphasize patterns in a variety of picture and text instances. comprehend NN's prediction mechanism, The patterns are further projected onto CT images for a better comprehension of the convolutional neural network architecture's decision-making.

The reasons for using Grad-CAM Framework:

- Easy to use.
- Universally applicable to various imaging tasks.
- Provides insights on prediction process.

To demonstrate the Grad-CAM method in the study, we used 2D slices of lungs. The predictions of the CNN models differ, implying that the architectures highlight different parts of the CT scan. One can notice that each model has distinct decision making. The DenseNet201 architecture relies on the inner part of the lungs to predict normal and COVID-19 images. The VGG19 model, however, is mainly focuses on the regions that are potential indicators of COVID-19, such as opacities. In the future studies, we are planning to utilize the same approach for the analyzing interactive predictions in 3D CT instances.

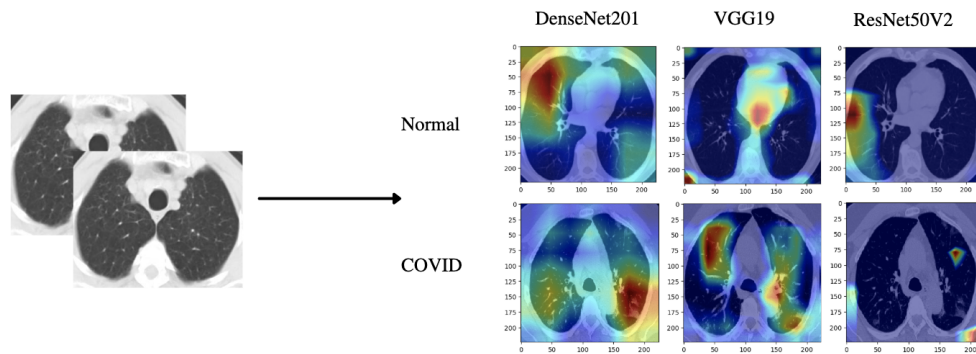


Figure 4-4: Predicting through Grad-CAM. Heatmap of Normal and Covid CT-scan images of models DenseNet201, VGG- 19, ResNet50V2.

Chapter 5

Conclusion

Considering the results achieved by our Deep Learning methods that include Convolutional Neural Networks and Ensemble Learning, we can state that Artificial Intelligence can play a crucial role in medicine and fight against COVID-19. Firstly, our study evaluated several most popular convolutional neural network architectures. Accuracy, sensitivity, specificity, and f-score were computed for the given CNNs. The models were fine-tuned and evaluated on the 3 of the open source datasets that contain Computer Tomography scans. The pretraining was done using one of the largest image datasets, namely ImageNet, which was also the dataset that was used as a benchmark. Using the list, one can rank the models in terms of their capability of COVID-19 recognition using CT images. The results suggested that the most accurate CNN model was VGG-19. Furthermore, we utilized the Ensemble learning algorithm for improving the accuracy scores of each individual architecture. The Ensemble model used the stack of the features extracted by individual models and logistic regression for classifying the features. The results suggested that in every case the performance and accuracy of the Ensemble model were superior to the same metrics achieved by CNN architectures. There are, however, several questions left that require further investigation. First, the structural features of the best performing models need to be considered to advance the performance of the architectures. Second, there is a large amount of other Ensemble learning methods that might perform even better and achieve a higher efficiency.

Bibliography

- [1] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.
- [2] Fei Wang, Lawrence Peter Casalino, and Dhruv Khullar. Deep learning in medicine—promise, progress, and challenges. *JAMA Internal Medicine*, 179(3):293–294, 2019.
- [3] Ahmed S Sultan, Mohamed A Elgharib, Tiffany Tavares, Maryam Jessri, and John R Basile. The use of artificial intelligence, machine learning and deep learning in oncologic histopathology. *Journal of Oral Pathology and Medicine*, 49(9):849–856, 2020.
- [4] Gardner L Dong E, Du H. An interactive web-based dashboard to track covid-19 in real time. *Lancet Inf Dis*, 2020.
- [5] D. Lu and Q. Weng. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870, 2007.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [7] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.
- [8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.
- [9] Weibin Wang, Dong Liang, Qingqing Chen, Yutaro Iwamoto, Xian-Hua Han, Qiaowei Zhang, Hongjie Hu, Lanfen Lin, and Yen-Wei Chen. Medical image classification using deep learning. In *Deep Learning in Healthcare*, pages 33–51. Springer, 2020.

- [10] Mahbub Hussain, Jordan J Bird, and Diego R Faria. A study on cnn transfer learning for image classification. In *UK Workshop on Computational Intelligence*, pages 191–202. Springer, 2018.
- [11] Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. Medical image classification with convolutional neural network. In *2014 13th International Conference on Control Automation Robotics and Vision (ICARCV)*, pages 844–848. IEEE, 2014.
- [12] N Narayan Das, Naresh Kumar, Manjit Kaur, Vijay Kumar, and Dilbag Singh. Automated deep transfer learning-based approach for detection of covid-19 infection in chest x-rays. *Journal of Innovation and Research in BioMedical engineering*, 2020.
- [13] Parnian Afshar, Shahin Heidarian, Farnoosh Naderkhani, Anastasia Oikonomou, Konstantinos N. Plataniotis, and Arash Mohammadi. Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images. *Pattern Recognition Letters*, 138:638–643, 2020.
- [14] Yanan Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Jiancheng Lv. Automatically designing cnn architectures using the genetic algorithm for image classification. *IEEE Transactions on Cybernetics*, 50(9):3840–3854, 2020.
- [15] Muhammad Sajjad, Salman Khan, Khan Muhammad, Wanqing Wu, Amin Ullah, and Sung Wook Baik. Multi-grade brain tumor classification using deep cnn with extensive data augmentation. *Journal of Computational Science*, 30:174–182, 2019.
- [16] Paulo Lacerda, Bruno Barros, Célio Albuquerque, and Aura Conci. Hyperparameter optimization for covid-19 pneumonia diagnosis based on chest ct. *Sensors*, 21(6):2174, 2021.
- [17] Giorgio Giacinto and Fabio Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9-10):699–707, 2001.
- [18] Byoungchul Ko, Ja-Won Gim, and JY Nam. Cell image classification based on ensemble features and random forest. *Electronics Letters*, 47(11):638–639, 2011.
- [19] R. Lavanyadevi, M. Machakowsalya, J. Nivethitha, and A. Niranjil Kumar. Brain tumor classification and segmentation in mri images using pnn. In *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*, pages 1–6, 2017.
- [20] Ziwei Zhu, Zhang Xingming, Guihua Tao, Tingting Dan, Jiao Li, Xijie Chen, Yang Li, Zhichao Zhou, Xiang Zhang, Jinzhao Zhou, et al. Classification of covid-19 by compressed chest ct image through deep learning on a large patients cohort. *Interdisciplinary Sciences: Computational Life Sciences*, 13(1):73–82, 2021.

- [21] Ying Bi, Bing Xue, and Mengjie Zhang. An automated ensemble learning framework using genetic programming for image classification. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 365–373, 2019.
- [22] Hasan Ucuzal, Şeyma Yaşar, and Cemil Çolak. Classification of brain tumor types by deep learning with convolutional neural network on magnetic resonance images using a developed web-based interface. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–5. IEEE, 2019.
- [23] Farzad Vasheghani Farahani, Abbas Ahmadi, and MH Fazel Zarandi. Lung nodule diagnosis from ct images based on ensemble learning. In *2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2015.
- [24] Xiaobo Liu, Zhentao Liu, Guangjun Wang, Zhihua Cai, and Harry Zhang. Ensemble transfer learning algorithm. *IEEE Access*, 6:2389–2396, 2017.
- [25] Gianluca Pontone, Stefano Scafuri, Maria Elisabetta Mancini, Cecilia Agalbato, Marco Guglielmo, Andrea Baggiano, Giuseppe Muscogiuri, Laura Fusini, Daniele Andreini, Saima Mushtaq, et al. Role of computed tomography in covid-19. *Journal of Cardiovascular Computed Tomography*, 15(1):27–36, 2021.
- [26] Plamen Angelov and Eduardo Almeida Soares. Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification. *MedRxiv*, 2020.
- [27] Jinyu Zhao, Yichen Zhang, Xuehai He, and Pengtao Xie. Covid-ct-dataset: a ct scan dataset about covid-19. *arXiv preprint arXiv:2003.13865*, 2020.
- [28] Tuan D Pham. A comprehensive study on classification of covid-19 on computed tomography with pretrained convolutional neural networks. *Scientific reports*, 10(1):1–8, 2020.
- [29] Hayden Gunraj, Linda Wang, and Alexander Wong. Covidnet-ct: A tailored deep convolutional neural network design for detection of covid-19 cases from chest ct images. *Frontiers in Medicine*, 7:1025, 2020.
- [30] Hayden Gunraj, Ali Sabri, David Koff, and Alexander Wong. Covid-net ct-2: Enhanced deep neural networks for detection of covid-19 from chest ct images through bigger, more diverse learning. *arXiv preprint arXiv:2101.07433*, 2021.
- [31] Zabit Hameed, Sofia Zahia, Begonya Garcia-Zapirain, José Javier Aguirre, and Ana María Vanegas. Breast cancer histopathology image classification using an ensemble of deep learning models. *Sensors*, 20(16), 2020.
- [32] Lingxi Xie, Richang Hong, Bo Zhang, and Qi Tian. Image classification and retrieval are one. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. Association for Computing Machinery, 2015.

- [33] Nadia Jmour, Sehla Zayen, and Afef Abdelkrim. Convolutional neural networks for image classification. In *2018 International Conference on Advanced Systems and Electric Technologies*, pages 397–402, 2018.
- [34] Aakash Kaushik. Understanding the vgg19 architecture. *Open-Genus Foundation*. Retrieved from: <https://iq.opengenus.org/vgg19-architecture>.
- [35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [37] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.
- [38] Gonçalo Marques, Deevyankar Agarwal, and Isabel de la Torre Díez. Automated medical diagnosis of covid-19 through efficientnet convolutional neural network. *Applied soft computing*, 96:106691, 2020.
- [39] Zeke Xie, Fengxiang He, Shaopeng Fu, Issei Sato, Dacheng Tao, and Masashi Sugiyama. Artificial neural variability for deep learning: on overfitting, noise memorization, and catastrophic forgetting. *Neural computation*, 33(8):2163–2192, 2021.
- [40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- [41] Reinhard Heckel and Fatih Furkan Yilmaz. Early stopping in deep networks: Double descent and how to eliminate it. *arXiv preprint arXiv:2007.10099*, 2020.
- [42] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.
- [43] AS Jokandan, H Asgharnezhad, SS Jokandan, A Khosravi, PM Kebria, D Nahavandi, S Nahavandi, and D Srinivasan. An uncertainty-aware transfer learning-based framework for covid-19 diagnosis. arxiv 2020. *arXiv preprint arXiv:2007.14846*.
- [44] Zabit Hameed, Sofia Zahia, Begonya Garcia-Zapirain, José Javier Aguirre, and Ana María Vanegas. Breast cancer histopathology image classification using an ensemble of deep learning models. *Sensors*, 20(16):4373, 2020.

- [45] Guy S Handelman, Hong Kuan Kok, Ronil V Chandra, Amir H Razavi, Shiwei Huang, Mark Brooks, Michael J Lee, and Hamed Asadi. Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods. *American Journal of Roentgenology*, 212(1):38–43, 2019.
- [46] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.
- [47] Yunyan Zhang, Daphne Hong, Daniel McClement, Olayinka Oladosu, Glen Pridham, and Garth Slaney. Grad-cam helps interpret the deep learning models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging. *Journal of Neuroscience Methods*, 353:109098, 2021.
- [48] Michael Maurus, Jan Hendrik Hammer, and Jürgen Beyerer. Realistic heatmap visualization for interactive analysis of 3d gaze data. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 295–298, 2014.