A COMPARISON OF DEEP NEURAL NETWORK ARCHITECTURES FOR COVID-19 DETECTION USING CT CHEST IMAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

MEHMET TUNAHAN SARIOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MEDICAL INFORMATICS

SEPTEMBER 2022

A COMPARISON OF DEEP NEURAL NETWORK ARCHITECTURES FOR COVID-19 DETECTION USING CT CHEST IMAGES

submitted by **MEHMET TUNAHAN SARIOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Health Informatics Department, Middle East Technical University** by,

Date: 02.09.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: **Mehmet Tunahan SARIOĞLU**

Signature :

ABSTRACT

A COMPARISON OF DEEP NEURAL NETWORK ARCHITECTURES FOR COVID-19 DETECTION USING CT CHEST IMAGES

SARIOĞLU, Mehmet Tunahan

M.S., Department of Health Informatics

Supervisor: Prof. Dr. Ünal Erkan Mumcuoğlu

September 2022, 52 pages

Coronavirus Disease 2019 (COVID-19) has rapidly spread around the world since December 2019. Due to the low confidence in the real-time reverse transcription-polymerase chain reaction (RT-PCR) test, which is considered the gold standard, CT images are frequently consulted for diagnosis. CT findings of COVID-19 patients has been analysed and documented in the literature, which are ground-glass opacities (GGOs), air bronchograms, vascular enlargement and halo sign. But, these CT findings encountered in COVID-19 disease are not very specific and vary during the course of the disease. In addition, it has common CT appearance and symptoms with many other infectious and non-infectious diseases. This thesis compares various deep transfer learning structures used to distinguish COVID-19 from other diseases using image processing techniques. In this study, the performances of the classification methods using five different convolutional neural network structures were compared. Unlike other similar studies, two public CT dataset was used together to train, test and validate the methods. Overfitting issues with the train data had been experienced and best scores are obtained using augmented data. Accuracy values of 81.65%, 86.08%, 90.82%, 79.75% and 81.01% were obtained in Xception, VGG 16, ResNet 50, Inception v3 and Inception ResNet v2 convolutional neural networks, respectively. The importance of hyper parameters such as epoch number, loss and activation functions used during training is also mentioned in this study.

Keywords: CT, COVID-19, Neural Networks, Transfer Learning, Classification

ÖZ

GÖĞÜS BT GÖRÜNTÜLERİYLE COVID-19 TESPİTİNDE DERİN SİNİR AĞI MİMARİLERİNİN KARŞILAŞTIRILMASI

SARIOĞLU, Mehmet Tunahan

Yüksek Lisans, Sağlık Bilişimi Bölümü

Tez Yöneticisi: Prof. Dr. Ünal Erkan Mumcuoğlu

Eylül 2022, 52 sayfa

Koronavirüs hastalığı 2019 Aralık ayından itibaren hızla tüm dünyaya yayıldı. Altın standart olarak kabul edilen gerçek zamanlı ters transkripsiyon-polimeraz zincir reaksiyonu (RT-PCR) testine olan güvenin düşük olması nedeniyle tanı için BT görüntülerine sıklıkla başvurulmaktadır. COVID-19 hastalarının BT bulguları, buzlu cam opasiteleri, hava bronkogramları, vasküler genişleme ve halo işareti olarak literatürde analiz edilmiş ve yayınlanmıştır. Ancak COVID-19 hastalığında karşılaşılan bu BT bulguları çok spesifik olmayıp hastalığın seyri sırasında da değişiklik göstermektedir. Ayrıca diğer birçok bulaşıcı ve bulaşıcı olmayan hastalık ile ortak BT görünümü ve semptomlarına sahiptir. Bu tez, görüntü işleme tekniklerini kullanarak COVID-19u diğer hastalıklardan ayırt etmek için kullanılan çeşitli derin transfer öğrenme yapılarını karşılaştırmaktadır. Bu çalışmada, beş farklı evrişimli sinir ağı yapısını kullanan sınıflandırma yöntemlerinin performansları karşılaştırılmıştır. Diğer benzer çalışmalardan farklı olarak, yöntemleri eğitmek, test etmek ve doğrulamak için iki genel BT veri seti birlikte kullanıldı. Eğitim verileriyle aşırı uyum sorunları yaşanmış ve en iyi puanlar artırılmış veriler kullanılarak elde edilmiştir. Xception, VGG 16, ResNet 50, Inception v3 ve Inception ResNet v2 evrişimli sinir ağlarında sırasıyla %81.65, %86.08, %90.82, %79.75 ve %81.01 doğruluk değerleri elde edilmiştir. Eğitim sırasında kullanılan epoch sayısı, kayıp ve aktivasyon fonksiyonları gibi hiper parametrelerin önemine de bu çalışmada değinilmiştir.

Anahtar Kelimeler: BT, COVID-19, Sinir Ağları, Transfer Öğrenimi, Sınıflandırma

To my beloved family.

ACKNOWLEDGMENTS

This study and the research behind it would not have been possible without the encouraging support of my supervisor, Ünal Erkan Mumcuoğlu. I would also like to extend my thanks to Assoc. Prof. Dr. Aybar Can Acar for their help and guidance in offering me the resources in running the trials. I would like to express my sincere thanks to Prof. Dr. Ziya Telatar for his comments and guidance on this research study.

I am also grateful to Emre Kaan Süslü, my friend from METU, for his feedbacks, and moral support.

In pursuing this project, no one has been more important to me than members of my family. I would like to thank my parents for being with me with their love and endless support in everything I do.

Most importantly, this study could not have been done without my loving and supportive wife, Beril Deniz, and my wonderful child, Atlas, who has been an endless inspiration to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
BGR	Blue Green Red
CGAN	Conditional Generative Adversarial Network
CIFAR-10	Canadian Institute For Advanced Research - 10
CNN	Convolutional Neural Network
COVID-19	Corana Virus Disease - 2019
CPU	Central Processing Unit
CT	Computed Tomography
CXR	Chest X-ray
ELU	Exponential Linear Unit
FN	False Negative
FP	False Positive
GGO	Ground Glass Opacity
GPU	Graphics Processing Unit
ILSVRC-2012	The ImageNet Large Scale Visual Recognition Challenge - 2012
KL	Kullback-Leibler
KNN	K-Nearest Neighbour
LDA	Linear Discriminant Analysis
MAE	Mean Absolute Error
MBE	Mean Bias Error
MSE	Mean Squared Error

PDF	Portable Document Format
ReLU	Rectified Linear Unit
ResNet	Residual Network
RGB	Red Green Blue
RT-PCR	Reverse Transcription Polymerase Chain Reaction
SARS-CoV-2	Severe Acute Respiratory Syndrome Coronavirus 2
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
VGG	Visual Geometry Group
XAI	Explainable Artificial Intelligence
xDNN	Explainable Deep Neural Network

CHAPTER 1

INTRODUCTION

This chapter includes brief introduction of this study. Problem definition, background research, classification techniques and similar studies are stated in this chapter.

1.1 Motivation and Problem Definition

SARS-CoV-2 virus was first diagnosed in Wuhan, China in November 2019. SARS-CoV-2 was defined as a pandemic disease by World Health Organization in March 2020 that continues to be effective all over the world [8]. When it was examined as a coronal shape in electron microscope, it was named as corona virus disease (COVID). Due to the high rate of spread, the disease reached a pandemic size in a short time. The total number of cases reported in the more than 2 years since its emergence is as seen in Table 1. Not only patients and healthcare workers, but all people in the world are affected by the COVID-19. Its impact on the world economy has dragged many countries into an economic crisis. Especially in underdeveloped and developing countries, cheaper solutions are needed. According to Bedford et al., both technical and financial supports are needed by the lower-income and middle-income countries to prevent and diagnose the COVID-19 disease [9].

Table 1: Number of COVID-19 cases reported on 1st July of 2022[1]

Country	Total Cases	Total Deaths	Active Cases
World	553,919,775	6,360,499	18,994,099
USA	89,515,140	1,043,303	3,466,984
India	43,500,504	525,168	123,746
Brazil	32,434,200	671,764	888,754
France	31,208,925	149,585	1,438,403
Germany	28,392,629	141,292	1,525,537

The Reverse transcription-polymerase chain reaction (RT-PCR) test is considered the gold standard for identifying patients with COVID-19. Wang et al. show that specimens collected from bronchoalveolar lavage fluid result highest positive rates [10].

Cross-infection risk, inadequate test kits, cost, and test result wait time highlight the importance of machine learning approaches [11].

1.2 Medical Imaging in COVID-19 Diagnosis

When the suspicion of COVID-19 is evaluated by the doctor during the acute course of the disease, the RT-PCR test is initially requested from the patient. Chest imaging is requested for patients whose PCR test is considered suspicious by the doctor. Radiologists try to diagnose by comparing COVID-19 and other types of pneumonia on chest imaging. Mostly applied imaging techniques are computed tomography(CT) and chest X-ray(CXR).

As statistically described in Das et al., the abnormalities found in chest CT include findings on chest X-ray images for the same patients [2].

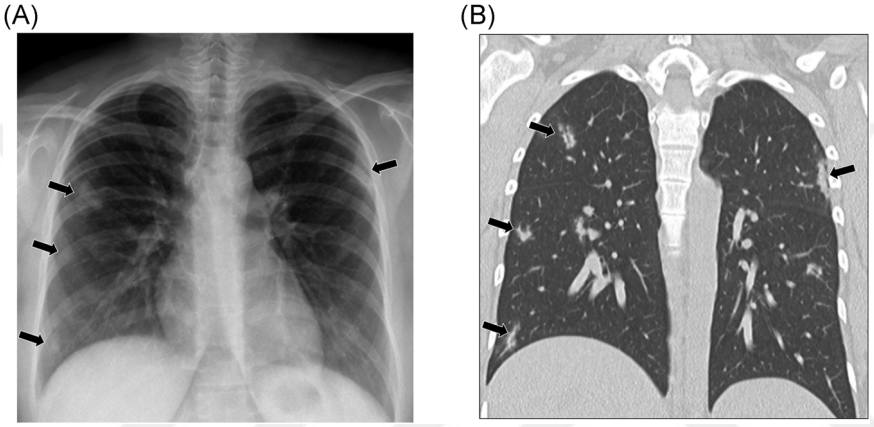


Figure 1: A patient with positive COVID-19 RT-PCR test. (**A**) Frontal chest radiograph (**B**) Coronal lung window CT image [2]

1.3 Findings in CT Images

Duration of the patient's symptoms, patient history, clinical signs, and other imaging findings are considered when diagnosing COVID-19. Computed Tomography (CT) and Chest X-ray (CXR) are also considered as significant imaging techniques. In order to make a consistent assessment, it is important to know the findings of other diseases beforehand.

CT findings are not very specific and COVID-19 disease has common symptoms with many infectious and non-infectious diseases [12]. As it is stated in the Figure 2, the similarities between COVID and non-COVID patients make it very difficult for radiologists to diagnose COVID-19.

CT findings in COVID-19 disease vary during the course of the disease. The early stage findings are unilateral or bilateral ground glass opacities (GGOs) in CT images. Day after day, GGOs become to consolidate. Appearance of ground glass opacities in CT images is also called as crazy-paving pattern. In addition to the peripheral distribution, vasodilation, and reverse halo sign, the crazy-paving pattern is one of the most common findings [13]. Findings in COVID-19 pneumonia and Non-COVID viral pneumonia diseases have similar features such as consolidation of the GGOs, peripheral lesion distribution and rounded morphology.

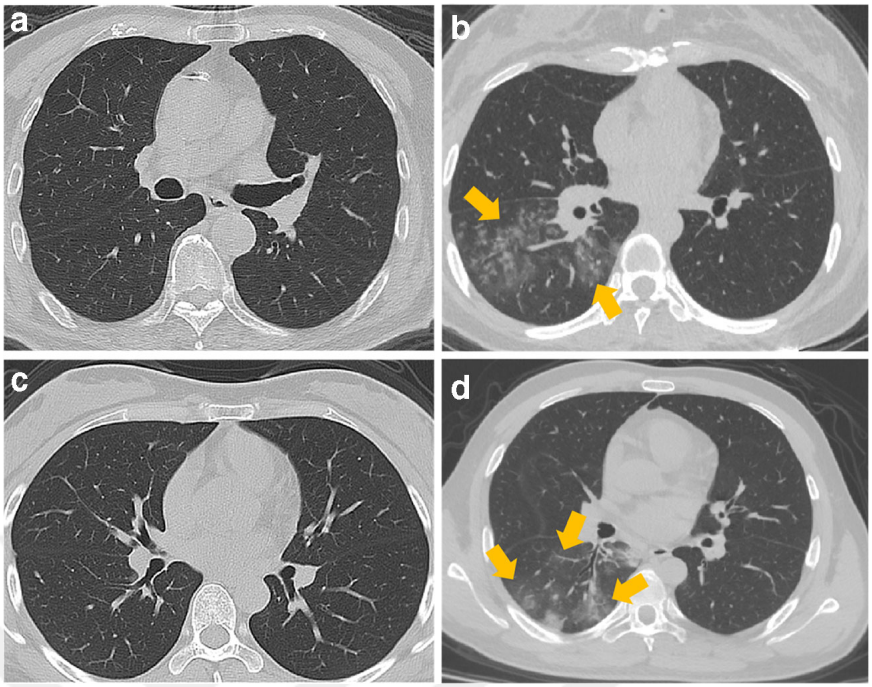


Figure 2: CT images of the non COVID-19 patients (**a, b**) and COVID-19 positive patients (**c, d**) [3]

1.4 CNN

Convolutional Neural Network (CNN) is a kind of an Artificial Neural Network (ANN) which has at least one layer containing convolution process in place of simple matrix multiplications [14]. CNNs are mainly designed for image diagnosis and classification studies. Number of studies using CNN is increasing, mostly in image classification field.

A convolutional neural network is structured with an input layer, an output layer and hidden layers. The layers between input and output layers are called as hidden layers. Outputs on hidden layers are masked by the activation function. Convolution operations are executed on the hidden layers. The output of the one layer is the input of the next layer. In a CNN, the input is a three dimensional tensor. The feeding image data for a convolutional layer has a shape of input height x input width x input channels. Pooling layers are designed to reduce the dimension before the next convolutional layer by combining multiple neurons into a single neuron. It's fully connected layer when every neurons in a layer connect with every neurons in the next layer.

The learning includes iteratively adjusting bias and weights. These biases and weights are used in the function computed by the neural network. Every neuron computes an output value using input values, biases and weights.

1.5 Network Training

The main purpose of the network training is finding best weights for the model. Networks are trained through predetermined number of iterations. Each iteration of the training procedure is called epoch. Weights are updated for every epoch. In the end of each epoch, an error gradient is calculated with the current state of the model. For most of the studies, it is not possible to feed the neural networks with the whole train data set in one epoch. The main reason for this is that it requires high memory to feed the neural network with an excessive amount of input. Therefore, in each epoch, the network model is trained with the subsets of the training data set. The batch is a subset of the images in the training set used in feeding the neural network at once. The number of images contained in a batch is the batch size. Error gradient is updated after every trained batch. After each batch in an epoch has been trained, the weights are updated by calculating estimates based on the validation set.

One of the objectives of network training is to achieve good performance on both of the training and test runs. This is an undesirable result if the network model performs well with the training set but cannot reach the same predictions with a new data. When the model learns the training set well but does not perform a good performance with the test data, it is called overfitting. Overfitting is a serious problem in network training. In order to avoid overfitting; dropout, data augmentation, early stopping and cross validation are commonly used on neural networks.

1.5.1 Dropout

Dropout means ignoring some random neurons on every iteration. In the study of Srivastava et al., the main purpose of dropout is explained as preventing neurons from adapting too much [15]. To drop out a neuron, input of this neuron is set to 0. Dropout layer works only when training the model, when evaluating scores with the test set this layer is not used. Dropping random neurons on every iteration means training different neural networks. Different neural networks cannot fit the same data in the same way, and it helps reduce overfitting.

1.5.2 Data Augmentation

Data Augmentation is a technique often applied to feed neural network with more training data. Augmenting the data helps the network adapt to more complex data, and provides flexibility for researchers working with few inputs. Transformation, cropping, scaling, contrast and brightness adjustment are augmentation techniques applied in image datasets. Combinations of rotation, flipping, zooming, stretching, shearing can be used for increasing transformed image count. Labels for the transformed samples should remain as originals[16].

1.5.3 Early Stopping

Early stopping is used for the same purpose as dropout. Both of these techniques aim to prevent overfitting. When the lower loss value is targeted, early stopping parameters stop the network training when the loss does not decrease by the desired amounts after a few iterations. Iteration count and minimum delta for the loss value are parameters for the early stopping function.

Otherwise, if the goal of the training is maximizing accuracy value, it can be considered to use delta for change of the accuracy. For this purpose, network training is stopped when the accuracy value is not increased as desired amount after several iterations. Parameters should be set as aiming to maximize the accuracy value.

Without early stopping, network models are trained more than satisfying number of iterations. After training for a longer period, the probability of overfitting the data increases and it is seen that the test accuracy begins to decrease.

1.5.4 Cross Validation

Most research studies start with splitting data between train and test sets. Cross validation aims to select the best model by testing with a separated part of the train dataset. For this, the data set is divided into multiple folds. The number of folds can vary according to the study; it is called K-fold Cross Validation. For every iteration, one of the folds is used as a test dataset, and the training procedure is called for every single fold. It results better estimates by training the model with multiple train/test sets. But it means cross-validation has a higher computation time. When number of folds increases, the time required for training also increases.

1.6 CNN Architectures

Convolutional neural networks have two main features that affect the operating speed and performance, network depth and convolutional filter size. In 2015, The VGG 16 and VGG 19 architectures were first introduced by the study of Simonyan et al., the effect of network depth was investigated on very small filters containing 16 and 19 layer depths, respectively [17]. VGG architecture was pre-trained on the ILSVRC-2012 dataset, and comparisons with the previous state-of-the-art solutions on large-scale image recognition was made using different datasets. Due to the comparison results, VGG architecture is applicable to different image classification fields such as medical imaging. Layers of the VGG network are described in Figure 3.

Residual networks are firstly explained as a solution of the vanishing gradient problem. Vanishing gradient is the problem with the extremely small gradient preventing the weights from updating [18]. Training is meaningless when the weights are not updated. In order to overcome this problem, shortcut connections added between layers. He et al. explained improvements on this architecture by comparing with the VGG nets [5]. Up to 152 layers, ResNet is 8 times deeper than VGG nets, however it has lower complexity. Instead of presenting this architecture with training on ImageNet dataset, He et al. mentioned that their study contains analysis of training on CIFAR-10 dataset. Layers of the ResNet architecture are stated in Figure 4.

Instead previously described architectures, it is aimed to achieve a good performance at relatively low cost using Inception network. There are variety of inception networks that contains similar architectures, Inception-v1, Inception-v2, Inception-v3, Inception-v4 and Inception - ResNet v1 and v2. The defined convolutional layers are explained as in Figure 5. The main evolution between the versions is the idea of having neurons with smaller input channels. To achieve this purpose, 1x1 convolution layers are added before other layers (3x3 and 5x5 convolution layers). In the study of Szegedy et al.,

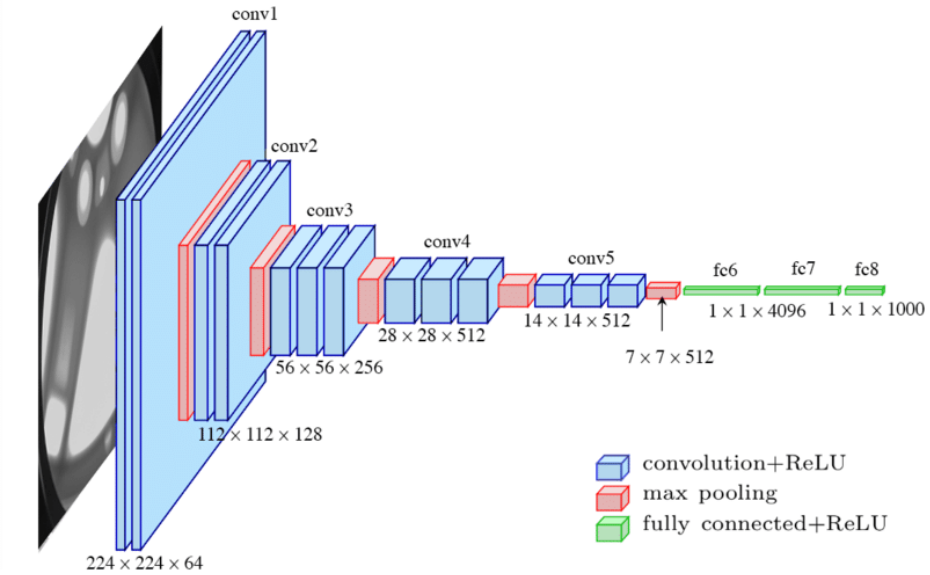


Figure 3: VGG architecture described by Ferguson et al.[4]

proposed a combination of ResNet and Inception architectures which is called Inception ResNet [6]. There are two variations of Inception ResNet network, v1 and v2, and Inception Resnet v2 suggested better results compared to v1.

Chollet et al. described Xception architecture into three parts, entry flow, middle flow and exit flow [7]. At the beginning of the entry flow, 299x299x3 images are expected as input, and at the end of the exit flow, 2048 dimensional vectors are output. Complete architecture of the Xception is drawn from the study of Chollet et al., and can be illustrated in Figure 6.

1.7 Performance Metrics

This section explains the meaning of the comparison metrics used in this study. Since this study classifies the binary problem, this section focuses only on metrics related to binary classes.

From Table 2, TP means the number of times it predicted correctly as COVID-19 positive, FP means the number of times it predicted correctly as COVID-19 negative, FN shows the number of wrong predictions as COVID-19 negative, and FP indicates the number of wrong predictions as COVID-19 positive.

Evaluation metrics used to check the performance of deep learning architectures: Accuracy, Sensitivity (Recall), Specificity, Precision.

Table 2: Confusion Matrix

		Predicted	
		Positive	Negative
True	Positive	TP	FN
	Negative	FP	TN

Accuracy is the number of correctly predicted cases divided by the total number of images in the test set, and given as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Sensitivity(Recall) is one of the critical metric used in medical studies. Shortly explained as the number of correctly predicted COVID-19 positive cases divided by the total COVID-19 positive cases.

$$Sensitivity(Recall) = \frac{TP}{TP + FN} \tag{2}$$

Specificity is the number of correctly predicted COVID-19 negative cases divided by the total predicted COVID-19 negative cases.

$$Specificity = \frac{TN}{TN + FN} \tag{3}$$

Precision is the number of correctly predicted COVID-19 positive cases divided by the total predicted COVID-19 positive cases.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

1.8 Similar Studies

This study proposes a COVID-19 diagnosis method that uses CT scan images. In this section methods used in COVID-19 pneumonia diagnosis studies are investigated.

Many of the articles in the literature mention that deep learning techniques are less costly than RT-PCR test. The Study of Wang et al. describes a deep learning system for diagnosis in less than 10 seconds after a CT scan within minutes [19]. It was mentioned that the findings described in previous radiological studies were determined by this deep learning system. Deep learning techniques are very successful in differentiating two patients with very similar CT findings despite being COVID-19 positive and COVID-19 negative [20]. Semi-supervised learning methods also studied to make diagnosis on COVID-19 CT findings. Ma Jun et al., offers a method combining classical active contour model with convolutional neural networks [21]. Combination of pre-trained models with different classifiers gives promising results. DenseNet with SVM classifier is experimented in order to make a diagnosis about the severity of the COVID-19 patients by Zekuan et al. [22].

When performing CT chest imaging, respiratory movement has a significant impact on CT slices. To eliminate motion artifacts, CT images are captured when patients hold a single long breath. Due to the acute development of COVID-19, determining the severity of the disease is an important area of research. By analyzing histogram data, Berta et al. analyzed images of COVID-19 positive and negative patients and studied the respiratory cycle [23].

Some of the studies include different stages of the COVID-19 infection. Xiong et al. highlighted 38% of the patients developed into severe disease 7 days after the admission [24]. Using radiomics in COVID-19 detection is hypothesized by Xiong et al.. Radiomics is previously used in diagnosing the severity of lung cancer by detecting pulmonary nodules [25].

Data is one of the most valuable things for any research field. Limited data availability is one of the major problems of health and medical research. Researchers apply data augmentation techniques to increase the number and types of data. Turkoglu's study showed differences between working with augmented data and working without data augmentation [26]. Most of the deep architectures (SVM, KNN, LDA, AlexNet, GoogleNet, etc) give better results with data augmentation than without data augmentation described in Turkoglu's work.

Instead of studying with only binary classes(COVID-19 and non COVID-19), some of the studies mentioned multiple classes containing other diseases that is identified by chest imaging techniques. In Le's work, chest X-ray images are processed for COVID-19 detection and 98.54% and 99.06% accuracy values are resulted on binary and multiple classes respectively [27]. COVID-19 findings in chest CT images have common signs with different diseases. Current studies contributed to detect and diagnose the COVID-19 pneumonia. After this pneumonia, valuable informations and studies will remain to be used in other diseases such as cancer, pneumonia, cystic fibrosis, and other acute lung diseases [28].

In this study, unlike most of the similar studies, the number of samples in the training set is increased with data augmentation techniques and the use of multiple data sets.

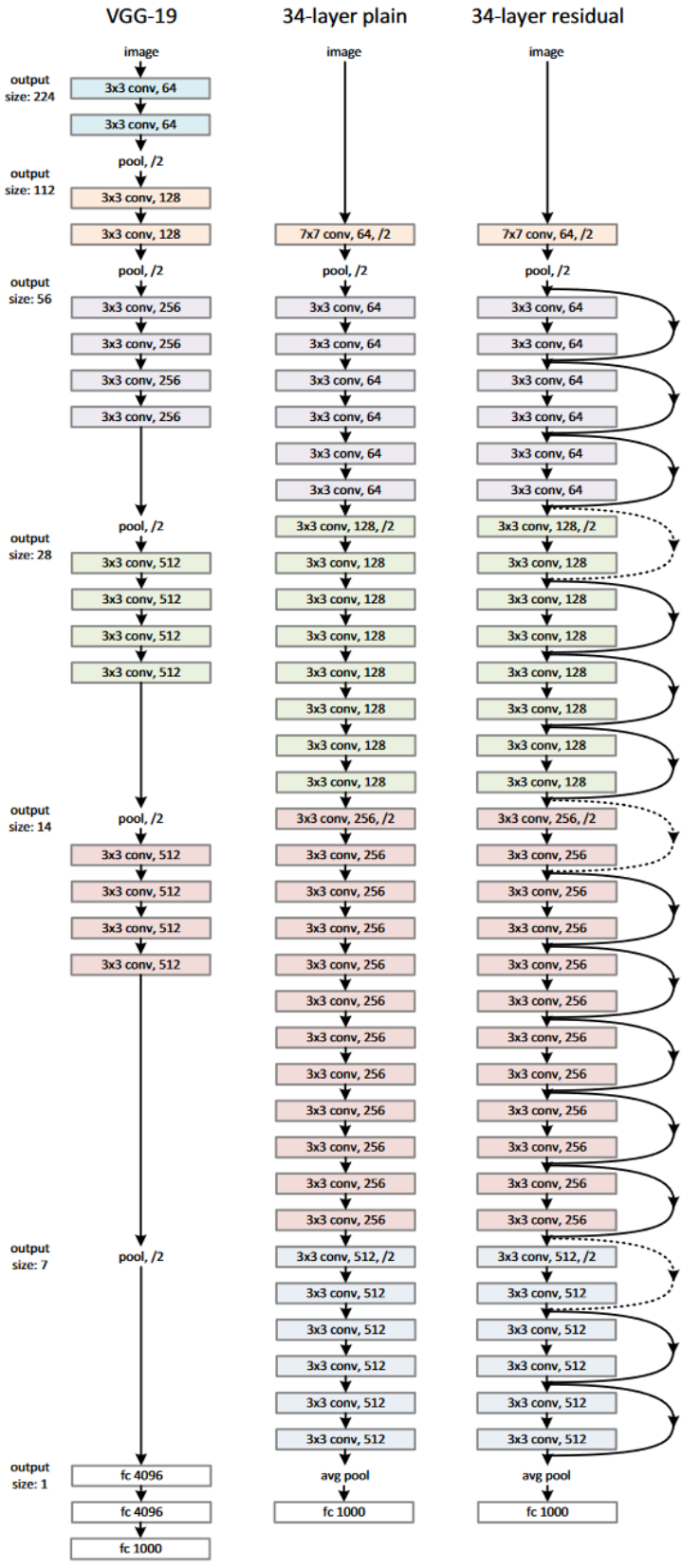


Figure 4: ResNet architecture described by He et al [5]. **A** is the VGG-19 model as a reference. **B** is a network with 34 layers. **C** is a residual network with 34 layers.

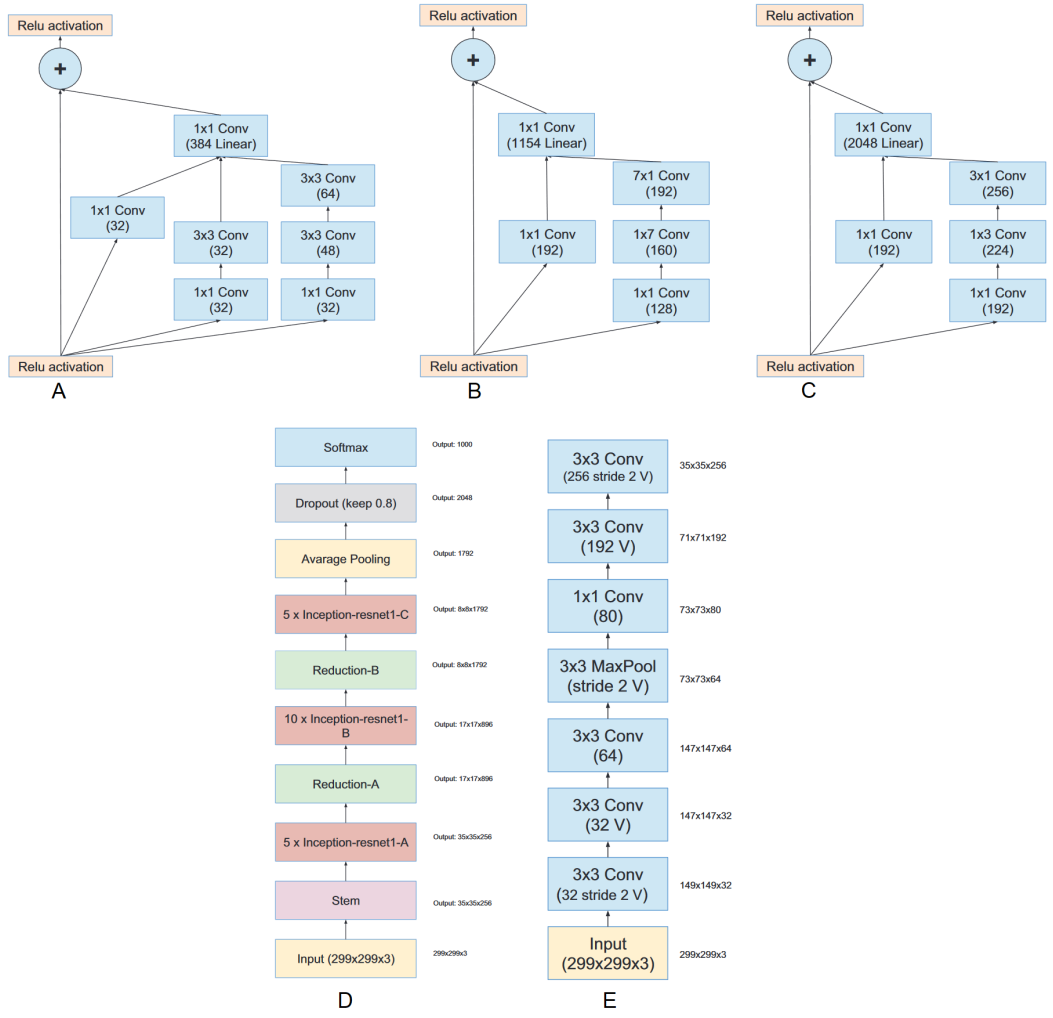


Figure 5: Inception architecture described by Szegedy et al [6]. **A**,**B** and **C** are the grid modules of the Inception ResNet 2 network, 35 x 35, 17 x 17 and 8 x 8 grid modules stated from left to right. **D** is the overall schema for the Inception ResNet v1 and v2 network. The stem of the Inception ResNet v1 is drawn as **E**.

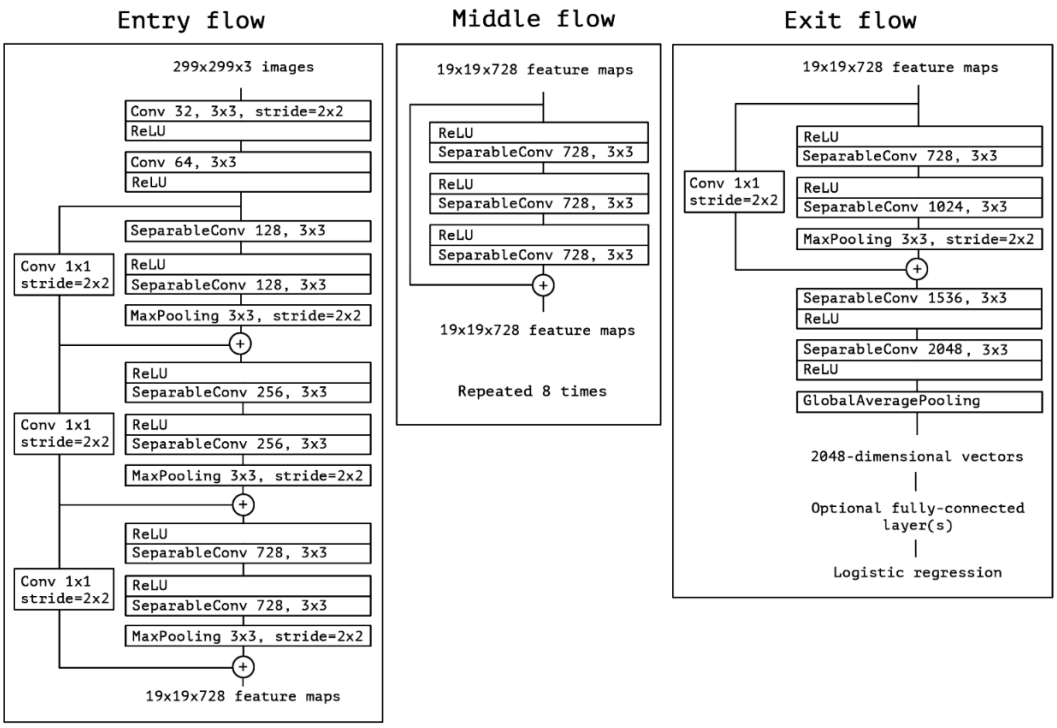


Figure 6: Xception architecture described by Chollet et al [7]. The data follows the path from the entry flow, then passes through the middle flow eight times, and finally ends the path with the exit flow.

CHAPTER 2

METHOD

In this chapter, methods are given in detail. In this study, Zhao's data set and SARS-CoV-2 CT-scan data set are used [29, 30]. Data augmentation techniques applied on image data set. After pre-processing steps are applied, deep transfer learning process was executed. The following sub sections explain these steps.

2.1 Data Set

Publicly shared data sets are used in this study. Zhao's dataset is one of the earliest published CT image data sets of the COVID-19 pandemic [29]. SARS-CoV-2 CT-scan data set is one of the largest COVID-19 CT image data sets published publicly [30]. 301 COVID-19 positive CT scan images are selected from Zhao's dataset, and combined with the images from SARS-CoV-2 CT scan dataset. 371 images from non COVID-19 patients on Zhao's dataset are combined with the non COVID-19 images from SARS-CoV-2 CT scan dataset. After combining images from two data sets, 1601 COVID-19+ CT scan images and 1627 non-COVID-19 CT scan images are used in this study. Table 3 shows the number of images collected from data sets.

Table 3: Number of COVID-19 Images in Data Sets

Data Set	COVID-19+	non-COVID-19
All	1553 (301+1252)	1601 (371+1230)
Zhao's Data Set[29]	349	397
SARS-CoV-2 Data Set[30]	1252	1230

2.1.1 Zhao's Data Set

Zhao's data set contains 349 COVID-19 CT images from 216 patients, and 397 non-COVID-19 CT images. These images are collected from preprints from medRxiv and bioRxiv by the first quarter of the 2021 [31, 32]. PDF files are analyzed and images are extracted with including patients' age, gender, and label about COVID-19 positive or not. Also after gathering images from PDF files, there is a significant loss on resolution. Instead of using all slices of the CT scans, some for the key slices are selected and added into the data set by the author. After communicating with an experienced

radiologist, authors of this data set commented that current image resolutions and single-slices of the images contains enough clinical evidence [29]. Due to the different resolutions on the images, this data set needs to be resized before feeding the neural network.

In this study some images from this data set are eliminated due to containing manually added signs of radiological findings. 48 COVID-19 and 26 non-COVID-19 images are excluded from the data set in this study. Some of the randomly selected images can be shown in Figure 7.

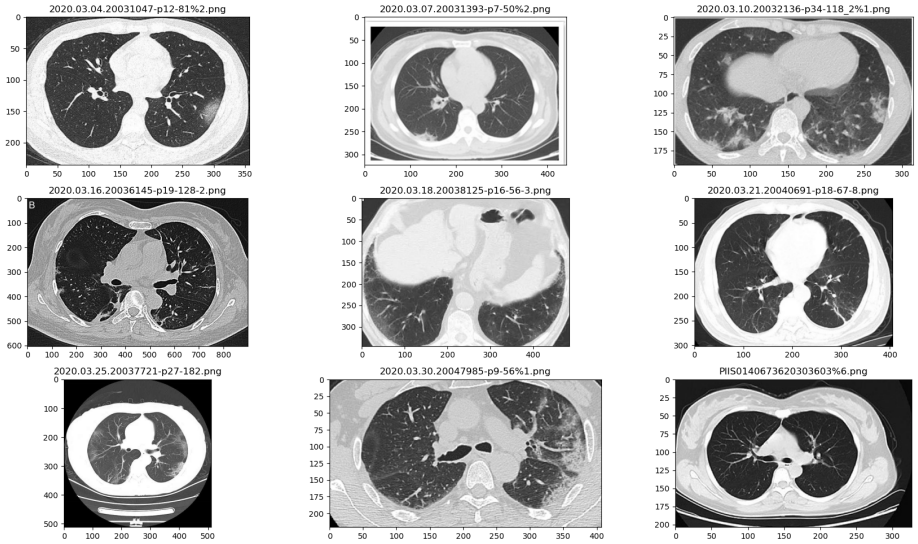


Figure 7: Randomly selected images from Zhao's CT scan data set

2.1.2 SARS-CoV-2 CT-scan Data Set

SARS-CoV-2 data set contains 1252 COVID-19 positive, 1230 COVID-19 negative images collected from an hospital in Sau Paulo, Brazil [30]. Images in this dataset were collected from 120 patients, half of whom were infected with SARS-CoV-2 and the others were not. Figure 8 shows some of the sample images from this data set.

This data set is used in a DenseNet based deep learning study [33]. Moreover, authors of that study mentioned the need of using multiple data sets for achieving better results.

2.2 Data Augmentation

Before the data augmentation step, the data was divided into 80%, 10%, and 10% segments as train, validation, and test, respectively. Train set contains 2.522 images, validation and test sets contain 316 images. Data augmentation was not applied to validation and test sets. The total number of images in the train set has been increased up to 8 times by augmentation techniques. After data augmentation, 20.176 images are generated to feed the model. In this study data augmentation was used with rotation between -/+5 degree with random probability and 50% probability of left/right flip. Samples from augmented images are shown in Figure 9.

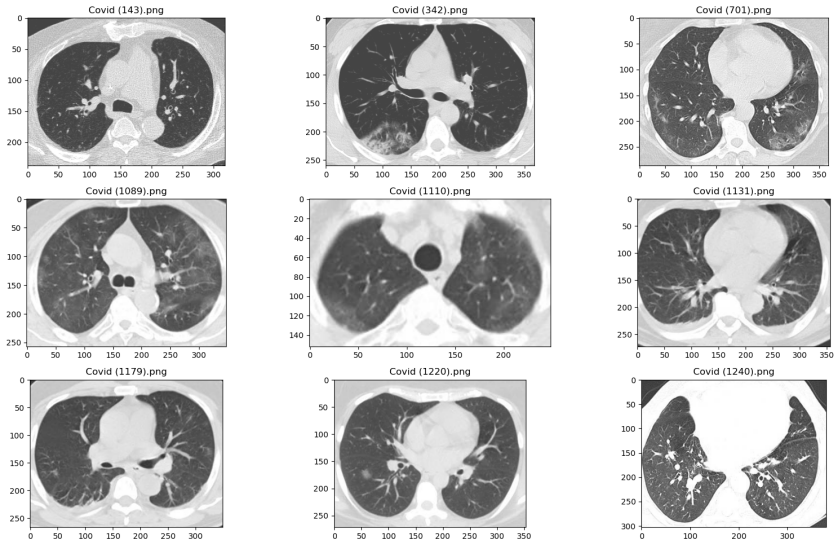


Figure 8: Randomly selected images from SARS-CoV-2 CT scan data set

Data augmentation techniques were performed using the Augmentor python library [16]. This library contains a wide variety of data augmentation techniques, such as rotating by desired angles (90, 180, 270), rotating by random angles, flipping up/down, flipping left/right, cropping to size, performing gaussian distortion and random distortion on the image. Moreover zoom, resize, scale, skew and shear operations can be performed using Augmentor library. In addition, Augmentor library provides an option to create and use augmented images on runtime istead of saving images on the disk.

2.3 Pre-processing

Data set is splitted into train, validation and test sets. After splitting data, data augmentation steps are executed. Number of images in the training set is increased to 20.176 as previously described in the Section 2.2.

As shown in Figure 7 and Figure 8, images on the data sets have different resolutions. After data augmentation, the number of samples with the same resolution was increased as seen in Figure 9. Each layer in the neural network expects the same input shape for each iteration. All images have been reshaped to have the same width and height. Since original images are not square, resizing is done keeping the aspect ratio and images are zero padded to fit the desired dimensions.

2.4 Loss Functions

For each classification problem, loss and activation functions are applied to observe the performance of the trials. There are several options for the loss functions used on deep learning approaches. Mean absolute error, mean bias error, mean squared error functions are some types of the loss functions mostly used on regression problems. Mean absolute error (MAE) is one of the most simple loss functions. MAE function takes the sum of the absolute differences between predicted and actual

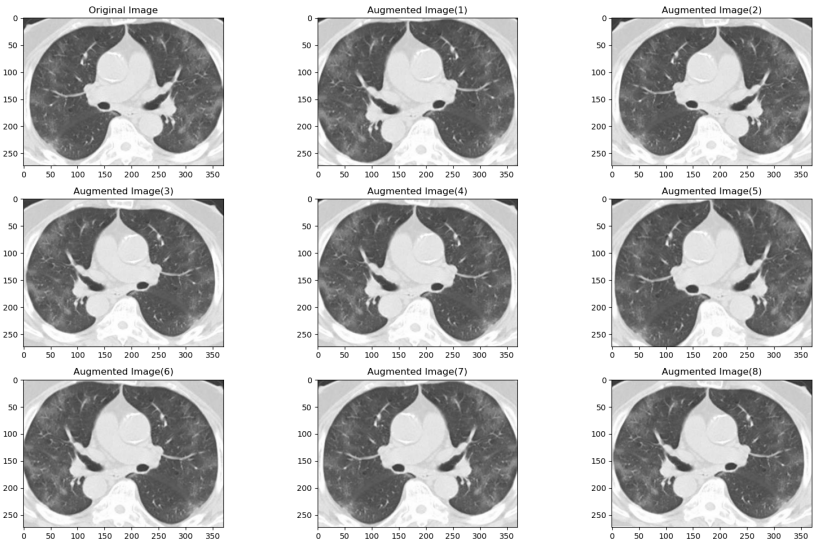


Figure 9: A series of images used in train set.

values. The mean absolute error is simply defined as:

$$MAE = \sum_{i=1}^{D} |x_i - y_i| \tag{5}$$

Mean bias error (MBE) is not used as much as the other functions because of the probability of the cancellation of positive and negative errors. MBE function takes the sum of the actual differences between actual and predicted values. The formula of the mean bias error is defined as:

$$MBE = \sum_{i=1}^{D} (x_i - y_i) \tag{6}$$

Mean squared error (MSE) is also the choice of the most data scientists for regression problems. MSE function takes the sum of the squared distances between predicted and actual values. The mean squared error is simply defined as:

$$MSE = \sum_{i=1}^{D} (x_i - y_i)^2 \tag{7}$$

For the classification problems, binary cross entropy, categorical cross entropy, Hinge loss, Kullback-Leibler divergence loss functions are frequently consulted. In this study, binary cross-entropy is used to calculate the loss for the binary classification problem. Cross-entropy loss between true labels and predicted labels are computed in this study. Binary cross entropy is commonly used for binary classification tasks with two classes. Binary cross entropy loss function penalizes the predictions depending on how close or far from the actual values. The formula of the binary cross entropy is given as:

$$Binary\ cross\text{-}entropy = -(y \log(p) + (1 - y) \log(1 - p)) \tag{8}$$

Categorical cross-entropy is a loss function very close to binary cross-entropy. This loss function is commonly used for categorical classification tasks. Categorical cross-entropy is the sum of binary

cross entropies for multiple classes. Equation of the categorical cross-entropy is defined as:

$$Categorical\ cross\text{-}entropy = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{9}$$

One of the commonly used loss function for classification problems is the hinge loss. This function is strongly used for support vector machines. Hinge loss is defined as:

$$Hinge\ Loss = max(0, 1 - y \cdot \hat{y}) \tag{10}$$

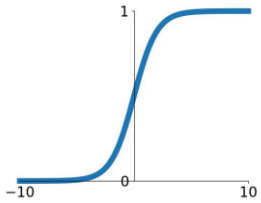
Kullback-Leibler (KL) divergence loss is a measure of how much a distribution differs from a reference distribution. KL is defined as:

$$KL(\hat{y}||y) = \sum_{c=1}^{M} \hat{y}_c \log \frac{\hat{y}_c}{y_c} \tag{11}$$

2.5 Activation Functions

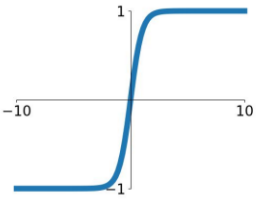
In artificial neural networks, activation functions are used to define the output of the nodes. Sigmoid, tanh, softmax, rectified linear unit (ReLU), leaky ReLU, and exponential LU (ELU) are some of the commonly used activation functions. In this study, the sigmoid activation function set in the last layer. Therefore, outputs are transformed into range between 0 and 1 by sigmoid activation function. This transformation process helps to solve the binary classification problem. Sigmoid activation function is defined as:

$$Sigmoid = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{12}$$



Tanh is an activation function which is closer to sigmoid function, additionally tanh function's range is between -1 and 1. Tanh function is defined as:

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{13}$$

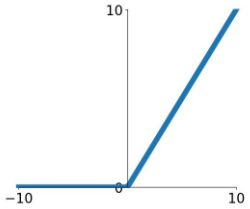


Softmax activation function is one of the different activation functions. This function transforms output values into probabilities. Softmax function is defined as:

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K \tag{14}$$

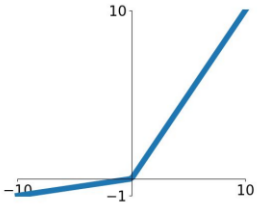
ReLU activation function is one of the most commonly used activation function. It is developed to solve the vanishing gradient problem. This activation function returns the positive values as is and returns zero for negative values. The formula of the ReLU activation function is defined as:

$$ReLU(x) = max(0, x) \tag{15}$$



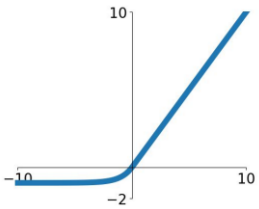
Leaky ReLU is the improved version of ReLU. For the negative values, this function returns minor values instead zero. Formula of the leaky ReLU is defined as:

$$LeakyReLU(x) = max(0.1x, x) \tag{16}$$



Exponential linear unit (ELU) shares all the benefits of ReLU, and negative side of this function is more robust than leaky ReLU. Formula of this function is defined as:

$$ELU(x) = \begin{cases} \alpha(e^x - 1) & \text{for} \quad x \leq 0 \\ x & \text{for} \quad x > 0 \end{cases} \tag{17}$$



On Keras framework, there is also an option not to set any activation functions in the last layer. Linear results are obtained when the activation function is not set. However, so by adding the "from_logits" parameter to the loss function, sigmoid transformed results can be obtained.

2.6 The Proposed Framework

In order to eliminate limited data size transfer learning was used in this study to fine-tune pre-trained deep neural networks. Network models are trained using images from train and validation sets. At the end of the training, prediction results are evaluated using images from the test set. Prediction results are explained in detail in the next chapter. Weights are saved for the model with the best results at the end of the training process. Thus, it can be reused later without the need to retrain that model.

2.6.1 Optimization

Adam optimizer is used in this study due to convergence speed. Only learning rate parameter of the Adam optimizer is set as 1e-5. Since Adam optimizer do well using default values for the other

parameters, Adam optimizer's other parameters are used as default values. As mentioned in the study of Kingma et al., the advantage of two popular optimization methods are combined in Adam optimizer, AdaGrad's ability to deal with sparse gradients and RMSProp's ability to deal with non-stationary objectives[34]. Adam optimizer is simple to implement and does not need high memory.

2.6.2 Hyperparameter Selection

As described in Section 1.5, epoch count and batch size need to be set before training phase. Number of epochs is set as 1000 for the runs in this study. Since early stopping callback function stops training before all epochs are finished, models are trained for 60 to 400 epochs on several runs. Different batch sizes are experimented in this study. Larger batches require more memory, so batch sizes smaller than 64 are used in batches.

Also there are other hyperparameters used in the runs such as trainable, weights, and include top parameters of base models. Firstly, trainable parameter takes true or false as value to determine whether layers of base models are trainable or frozen. Some of the runs in this study contains trainable layers, some others have done using frozen layers on base models. Usage of the trainable parameter is described in details in the next sections. Then weights parameter is used as imagenet to load weights of the neurons as previously trained using ImageNet dataset. The use case of the include top parameter is whether to include the last dense layer of the model. In this study, include top parameter is set as False due to a dense layer is already used on the last layer of the complete model.

2.6.3 Transfer Learning

Transfer learning is widely used in studies where there are not enough training samples [35]. With transfer learning, the deep learning process starts with initial weights that already good. After fine-tuning the model, satisfying results can be achievable.

Model spesific preprocess layers are added before functional layer of the base model. Input image or array is prepared by the preprocess layers to feed the next neurons with appropriate tensors. Details of the preprocess layers differ for the experimented architectures, and described in the following sections in this chapter.

Initially, base model was freezed and only input and output layers were fit for only one epoch. The purpose of doing that single run is updating initial weights with the train data, previously weights are set using ImageNet dataset. After this single run, the base model was unfreezed and fine-tuning of the entire model started. In this study the performance of five pre-trained models are evaluated, Xception[7], VGG 16[17], ResNet 50[5], Inception v3 and Inception ResNet v2[6]. The following sections contain detailed explanations.

Output of the functional layer feeds the global average pooling layer. Global average pooling layer takes the averages of the input feature maps. The operations on this layer does not require a parameter to optimize. This layer helps to avoid overfitting.

Another layer used to avoid overfitting is dropout layer. Dropout means dropping out randomly selected neurons on the layer. With dropout operation applied in every iteration, different model is trained

to avoid overfitting. Dropout rate can be adjustable depending on the study. In this study dropout rate is set to 0.2. 20% of the neurons are randomly dropped out in every epoch.

The last layer is dense layer. Dense layer transforms previous layer's output into desired dimension. Output of the dense layer represents the number of classes used in classification result. In this study, due to the binary classification (COVID-19 positive or not) dense layer outputs a tensor with one value. Dense layer's output is the output of the entire model.

Plotted models for every architecture experimented in this study are stated on the following sections.

2.6.4 Model Specific Layers of Xception

The Xception architecture used in this study was trained under ImageNet with the input pixels between -1 and 1. The original image data used in this study contains data range between 0 and 256. For the preprocessing step, rescaling layer is added before functional layer of the Xception architecture. Rescaling layer scales the image data between -1 and 1. This procedure is required to get the better output from the functional layer. Rescaling formula is given in Equation 18.

$$outputs = inputs * (1/127.5) - 1 \tag{18}$$

Rescaling Formula

The summary of the layers using Xception architecture are described in the Figure 10. To sum up, an input layer, two preprocessing layers, a functional layer, a pooling layer, a dropout layer and a dense layer are processed to get the output for every input.

2.6.5 Model Specific Layers of VGG 16

Model specific preprocessing steps of VGG 16 are converting input values from RGB to BGR, and zero-centering each color channel without scaling. The pre-trained VGG network is trained with this data shape on ImageNet dataset. In this study, grayscale CT images are used in RGB format due to the weights of the pre-trained models are previously trained using RGB formatted images with three dimensional channels. Keras's utility lib-rary supports means for importing grayscale images in RGB format with simply copying the grayscale channel into second and third channels. Since the number of parameters of the preprocessing layers is very low compared to all layers, the pre-process steps in the Keras library are used as they are. For the VGG 16 and ResNet 50 architectures, the pre-process step that converts images from RGB format to BGR format was used as defined in the Keras library and does not have any impact on the performance of these architectures. As the Keras library did not separate the pre-process steps that involve zero-centering and converting the color channels for these architectures, these two pre-process steps had to be used together.

The network layers using VGG 16 architecture are described in the Figure 11. In summary, an input layer, two preprocessing layers, a functional layer, a pooling layer, a dropout layer and finally a dense layer are processed to get the output for every input.

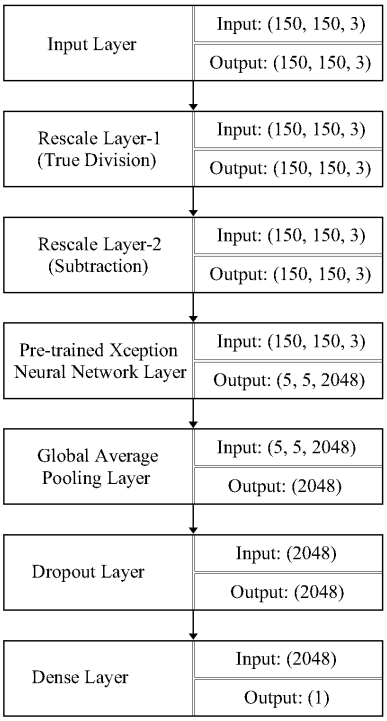


Figure 10: Plotted model using Xception architecture

2.6.6 Model Specific Layers of ResNet 50

Preprocessing steps of ResNet 50 architecture are the same as VGG 16 architecture. Input values are converted RGB to BGR, then each color channel is zero-centered without scaling.

Figure 12 describes the network layers containing an input layer, two preprocessing layers, a functional layer, a pooling layer, a dropout layer and a dense layer.

2.6.7 Model Specific Layers of Inception v3 and Inception ResNet v2

Preprocessing steps of Inception v3 and Inception ResNet v2 architectures are the same as the preprocessing steps of Xception architecture. It expects input pixels between -1 and 1, and a rescaling layer is implemented before functional layer as described in Equation 18.

Network layers of Inception v3 and Inception ResNet v2 models are shown in Figure 13 and Figure 14 respectively.

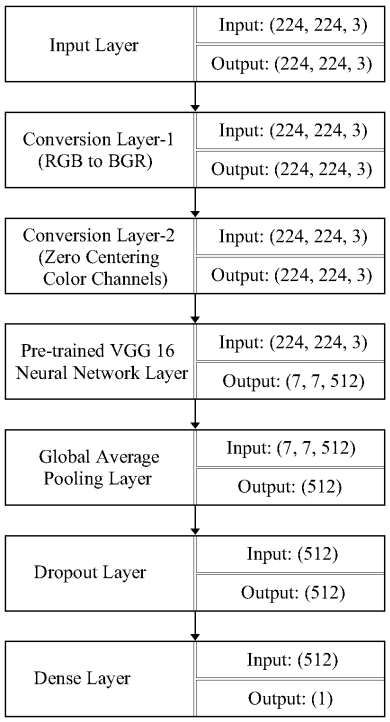


Figure 11: Plotted model using VGG 16 architecture

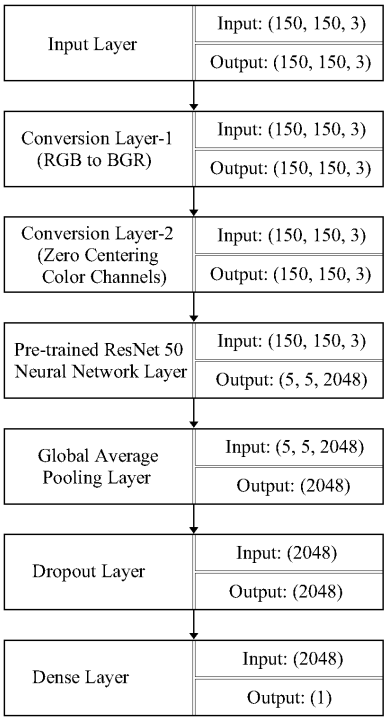


Figure 12: Plotted model using ResNet 50 architecture

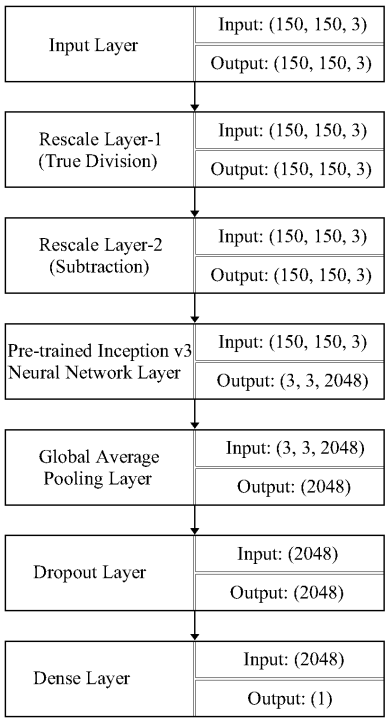


Figure 13: Plotted model using Inception v3 architecture

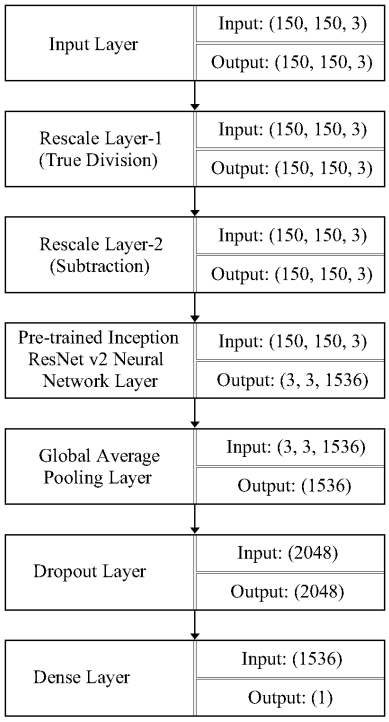


Figure 14: Plotted model using Inception ResNet v2 architecture

CHAPTER 3

RESULTS

In this chapter, comparisons and experiment results are presented. The augmented data specified in the tables in this section contain 20.176 augmented images, and 2.522 images are used to train the network without augmentation. Test results are evaluated using 160 COVID-19 positive CT scan images and 156 COVID-19 negative CT scan images from test set.

The implementation of the codes used in this study was made on python. The Keras framework was chosen to use pre-trained network models as functional layer [36]. Written python scripts are run on 64 bit Ubuntu 20.04 operating system with 32 gigabytes of memory, Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60GHz CPU, GeForce GTX 1080 Ti GPU.

Network performances that are explained in this chapter are calculated as follows. For each network, 6 training processes are performed. Network models are trained using both augmented data and original data. Different batch sizes (32, 48, 64) are experimented in the training phase. Then, the metrics explained in details of this chapter are calculated from the confusion matrices. A confusion matrix is calculated for each training result with the combination of augmentation status and batch size.

3.1 Evaluation of Xception Network

In this section performance results of Xception network model are provided. Confusion matrices of the evaluated results on Xception network can be seen in Table A.1-A.6. According to Table 4, highest sensitivity, specificity, precision, and accuracy scores are recorded with augmented data and batch size of 48 as 95.63%, 95.65%, 98.71%, 97.15% respectively. Using same batch size, trained Xception network with augmented data results higher accuracy with batch sizes of 48 and 64, however the trained network without augmentation shows higher accuracy with using batch size of 32.

3.2 Evaluation of VGG 16 Network

This section contains evaluated performance results for VGG 16 network model. For the evaluated results of VGG 16 network model, confusion matrices are shown in Table A.7-A.12. Using the VGG 16 network model, the highest sensitivity score is recorded as 97.50% and specificity score of 97.26%, but the lowest precision score is 91.76% and accuracy score is 94.40% without data increase and batch size of 48.The highest accuracy of 97.15% was achieved using data augmentation with a batch size of

Table 4: Xception evaluation results. Best scores are bold.

Model	Augmentation	Batch Size	Sensitivity	Specificity	Precision	Accuracy
Xception	Yes	32	93.75	93.83	97.40	95.57
Xception	Yes	48	**95.63**	**95.65**	**98.71**	**97.15**
Xception	Yes	64	93.75	93.87	98.04	95.89
Xception	No	32	94.38	94.44	98.05	96.20
Xception	No	48	94.38	94.30	95.57	94.94
Xception	No	64	94.38	94.34	96.18	95.25

48. According to the Table 5, the VGG 16 network model gives higher sensitivity and specificity scores with augmented data, but results without augmentation have higher precision and accuracy scores.

Table 5: VGG 16 evaluation results. Best scores are bold.

Model	Augmentation	Batch Size	Sensitivity	Specificity	Precision	Accuracy
VGG 16	Yes	32	96.25	96.18	96.86	96.52
VGG 16	Yes	48	96.25	96.23	98.09	**97.15**
VGG 16	Yes	64	94.38	94.48	**98.69**	96.52
VGG 16	No	32	96.25	96.23	98.09	**97.15**
VGG 16	No	48	**97.50**	**97.26**	91.76	94.30
VGG 16	No	64	95.63	95.60	97.45	96.52

3.3 Evaluation of ResNet 50 Network

Performance results of ResNet 50 network model are provided in this section. Confusion matrices of the provided results on ResNet 50 network can be found in Table A.13-A.18. As stated in Table 6, with the augmented data, the same sensitivity score is recorded with batch sizes of 48 and 64, however specificity, precision and accuracy scores are higher with the batch size of 48. Without data augmentation best result is recorded with the batch size of 48. The highest scores are equal with or without data augmentation as 95.63% of sensitivity, 95.64% of specificity, 98.08% of precision and 96.84% of accuracy scores. All accuracy scores recorded with augmented data are higher than the accuracy scores recorded without augmentation for the same batch sizes.

Table 6: ResNet 50 evaluation results. Best scores are bold.

Model	Augmentation	Batch Size	Sensitivity	Specificity	Precision	Accuracy
ResNet 50	Yes	32	95.00	95.00	97.44	96.20
ResNet 50	Yes	48	**95.63**	**95.63**	**98.08**	**96.84**
ResNet 50	Yes	64	**95.63**	95.57	96.84	96.20
ResNet 50	No	32	94.38	94.16	93.21	93.67
ResNet 50	No	48	**95.63**	**95.63**	**98.08**	**96.84**
ResNet 50	No	64	93.13	93.21	96.75	94.94

3.4 Evaluation of Inception v3 Network

Test results with the Inception v3 network are explained in this section. Confusion matrices for the evaluated results of this network are detailed in Table A.19-A.24. Training scores with data augmentation are significantly higher than the scores without using augmented data for all of the experimented batch sizes. The highest sensitivity, specificity, precision and accuracy scores are achieved with the batch size of 32 and using augmented data as 98.75%, 98.72%, 98.75% and 98.73% respectively. According to the Table 7, among tests without augmented data, highest scores are achieved with batch size of 32 for all of the metrics except precision.

Table 7: Inception v3 evaluation results. Best scores are bold.

Model	Augmentation	Batch Size	Sensitivity	Specificity	Precision	Accuracy
Inception v3	Yes	32	**98.75**	**98.72**	**98.75**	**98.73**
Inception v3	Yes	48	94.38	94.30	95.57	94.94
Inception v3	Yes	64	95.63	95.63	98.08	96.84
Inception v3	No	32	95.00	95.06	98.70	96.84
Inception v3	No	48	94.38	94.27	94.97	94.62
Inception v3	No	64	95.00	94.74	92.68	93.67

3.5 Evaluation of Inception ResNet v2 Network

In this section, test results using Inception ResNet v2 network are explained. In Table A.25-A.30, confusion matrices used to evaluate the results in this section are stated. Evaluated scores of this network are similar to the Inception v3 network. Unlike Inception v3 model, batch size of 48 shows better scores than 32. The highest scores of sensitivity, specificity, precision and accuracy are achieved by using augmented data with the batch size of 32 as following 96.88%, 96.79%, 96.88% and 96.84%, respectively. As shown in Table 8, test results with augmented data are higher than the achieved results without using augmented data.

Table 8: Inception ResNet v2 evaluation results. Best scores are bold.

Model	Augmentation	Batch Size	Sensitivity	Specificity	Precision	Accuracy
Inc. ResNet v2	Yes	32	95.63	95.54	96.23	95.89
Inc. ResNet v2	Yes	48	**96.88**	**96.79**	**96.88**	**96.84**
Inc. ResNet v2	Yes	64	96.25	96.13	95.65	95.89
Inc. ResNet v2	No	32	95.63	95.39	93.29	94.30
Inc. ResNet v2	No	48	92.50	92.64	96.73	94.62
Inc. ResNet v2	No	64	93.75	93.67	94.94	94.30

3.6 Evaluation of Network Models Using Seperated Datasets

For all of the runs in the previous subsections, the training accuracy scores reached 100% while loss values decreased to the zero. Considering the high prediction rates of previous runs, the effects of the data on the results are investigated in more depth with the runs described in this subsection. After recording previous results using the combination of the two datasets, the models are trained using both datasets separately. Since the effect of batch size ws not detected in previous runs, batch size is used by default at 32 for the training process of seperated datasets. Network models are trained using images from one of the datasets. However, test results described in Table 9 are saved from the predictions made using the same test set used in previous runs. This means that the test set used in the runs explained in this section contains the same set of images as the previous sections.

Table 9: Training performances of the network models when images in the training set are used only from one of the data sets. Batch size is used by default at size 32.

Network	Train data set	Sensitivity	Specificity	Precision	Accuracy
Xception	Zhao's	73.13	60.91	56.80	58.23
ResNet 50	Zhao's	70.63	59.83	56.78	57.91
Inception v3	Zhao's	91.88	75.00	55.68	58.86
Inception ResNet v2	Zhao's	68.13	61.07	58.92	59.81
VGG 16	Zhao's	69.38	55.05	53.62	54.11
Xception	SARS-CoV-2	81.88	83.80	95.62	88.92
ResNet 50	SARS-CoV-2	78.13	81.38	97.66	87.97
Inception v3	SARS-CoV-2	80.00	82.22	94.12	87.34
Inception ResNet v2	SARS-CoV-2	80.63	82.97	96.27	88.61
VGG 16	SARS-CoV-2	83.75	85.06	94.37	89.24

Since the number of images in Zhao's dataset and the SARS-CoV-2 CT scan dataset is not equal, the test set contains a different number of images from these two datasets. When we look at the datasets from which the image is obtained, we see an unbalanced distribution. Data distribution of the test set can be shown in Table 10. Accuracy scores of the test results appear to be in different ranges depending on the dataset. While the accuracy scores of the models trained using the Zhao dataset vary between 54% and 60%, the accuracy scores of the models trained using the SARS-CoV-2 dataset are between 87% and 90%.

Table 10: Data distribution in the test set.

Dataset	COVID-19	non-COVID-19	Total
SARS-CoV-2	131	140	271
Zhao's	25	20	45

3.7 Evaluation of Network Models Without Using Trainable Layers

When the high accuracy scores obtained in the previous runs and the relatively lower training accuracy scores obtained with separated datasets are examined, it is seen that the models overfit the data. All

the results were evaluated together, it was determined that high accuracy scores were achieved after less than 20 iterations for all of the runs. At the same time, it was observed that the training accuracy value increased constantly, reaching 100%.

In this section, it is useful to mention some details about making the base model trainable. The neural network architectures used in this study contain millions of parameters. These parameters represent the weight values of neurons that can be updated in each iteration during training. In other words, while training the model, millions of weight values are updated according to the input values and the best estimation ratio is tried to be achieved. The fact that all parameters are updated causes overfitting in a short time with the effect of working with a small data set. In order to prevent this, it was preferred to train by freezing the base model. With this change, the weights of the pre-trained base model are not updated, only the existing parameters in the input and output layers are updated when the models are trained. Number of epochs is set to 1000 for these training runs. Iterations were terminated manually and the threshold value controlled for early stop was determined as 1×10^{-2}. When the trained models were terminated in accordance with the parameters in the early stop function, it was observed that the training and validation accuracy values did not reach 100%.

Table 11 shows the results of the latest runs when layers of the base models are frozen. Since there is no significant effect of batch size is detected, results of the trained models are compared for default batch size of 32. According to the table, ResNet 50 base model makes more successful predictions than other models. Using ResNet 50 architecture 89.38% of sensitivity, 88.67% of specificity, 86.14% precision and 86.14% of accuracy scores are achieved.

Table 11: Training performances of the network models using freezed base models. Batch size is used by default at size 32.

Network	Augmentation	Sensitivity	Specificity	Precision	Accuracy
Xception	Yes	81.25	79.87	77.84	78.80
VGG 16	Yes	79.38	79.88	83.55	81.65
ResNet 50	Yes	**89.38**	**88.67**	**86.14**	**87.34**
Inception v3	Yes	75.63	75.16	76.10	75.63
Inception ResNet v2	Yes	87.50	83.74	72.54	76.90
Xception	No	65.00	67.82	73.24	70.25
VGG 16	No	79.38	72.95	65.46	68.35
ResNet 50	No	75.63	75.47	77.07	76.27
Inception v3	No	78.75	68.22	60.29	62.97
Inception ResNet v2	No	68.13	66.45	66.46	66.46

Validation accuracy and loss curves for the runs with augmented data are shown in Figure 15 and Figure 16 respectively. Accuracy scores are very low in early epochs, but performance rapidly improved in the very first epochs. The accuracy score for the ResNet 50 architecture rises above 0.80 in the first 10 epochs, then gradually reaches the highest accuracy. According to the Figure 16, the minimum validation loss value is seen below 0.45 using ResNet 50 model. As shown in the graphs, runs were stopped after trained with different number of epochs by early stop callback function.

3.8 Evaluation of Network Models With Optimized Parameters

The results of the runs using lower minimum delta value for the early stopping function are described in this section. Since the slope in the validation loss is still trending downwards in Figure 16, the idea of increasing the number of epochs was evaluated. The trained models does not achieve maximum epoch count due to early stopping function. Even if number of epochs was set a higher value, performance of the model was not affected as the training was stopped by the early stopping function. Models are trained again with optimizing early stopping function parameters. Minimum delta parameter of the early stopping was set as 1×10^{-2} on all of the previous runs. The next runs were made using minimum delta value as 1×10^{-3} and validation accuracy and loss curves for this runs are shown in Figure 17 and Figure 18, respectively. As shown in Figure 18, lower validation loss scores than previous runs were achieved.

The slope of the validation loss graph still looks bearish. Ignoring the time constraint, it was decided to train the models during the maximum epoch number without using the early stopping function. After early stopping function was excluded, models were trained for 1000 epochs as defined on all of the runs in this study. Without early stopping, it was seen that the decline in the slope of the loss curve had come to an end.

According to the Figure 19, maximum validation accuracy score was achieved by ResNet 50 architecture as 0.85. Figure 20 shows a significant improvement in the minimum validation loss score of 0.35. Table 12 shows the results of the runs without using early stopping function. Highest scores are achieved by ResNet 50 model with 90.63% of sensitivity, 89.80% of specificity, 92.26% precision and 90.82% of accuracy scores.

Table 12: Training performances of the network models using freezed base models without early stopping. Batch size is used by default at size 32.

Network	Augmentation	Sensitivity	Specificity	Precision	Accuracy
Xception	Yes	79.38	79.88	83.55	81.65
VGG 16	Yes	83.75	84.15	88.16	86.08
ResNet 50	Yes	89.38	89.44	**92.26**	**90.82**
Inception v3	Yes	83.13	81.51	78.24	79.75
Inception ResNet v2	Yes	86.25	84.29	78.41	81.01
Xception	No	86.25	84.51	79.31	81.65
VGG 16	No	84.38	84.57	87.66	86.08
ResNet 50	No	**90.63**	**89.80**	85.80	87.66
Inception v3	No	77.50	77.50	79.49	78.48
Inception ResNet v2	No	80.00	79.22	79.01	79.11

3.9 Comparison of Experimented Network Models

Since sensitivity (recall) is one of the significant measures in medical research area, sensitivity scores are compared in Table 13. Higher sensitivity scores are recorded using augmented data for Inception

v3, and Inception ResNet v2 models. However sensitivity score of the Xception, VGG 16 and ResNet 50 network models without data augmentation are higher than using augmented data.

Table 13: Comparison of sensitivity scores of the proposed network models. Best scores are bold.

Model	With Augmented Data	Without Augmentation
Xception	79.38	86.25
VGG 16	83.75	84.38
ResNet 50	**89.38**	**90.63**
Inception v3	83.13	77.50
Inception ResNet v2	86.25	80.00

According to Table 14, highest accuracy score is achieved with augmented data and ResNet 50 network model. The accuracy values of experiments with augmented data for all network models are higher than those without augmentation when there is no difference recorded on accuracy scores for the Xception and VGG 16 architectures compared to the augmentation. According to the accuracy scores, data augmentation has a positive effect on diagnosing COVID-19 positive cases for most of the experimented network models in this study. Compared to the previous run, the maximum accuracy score achieved in the final runs has been improved by 4 percent.

Table 14: Comparison of accuracy scores of the proposed network models. Best scores are bold.

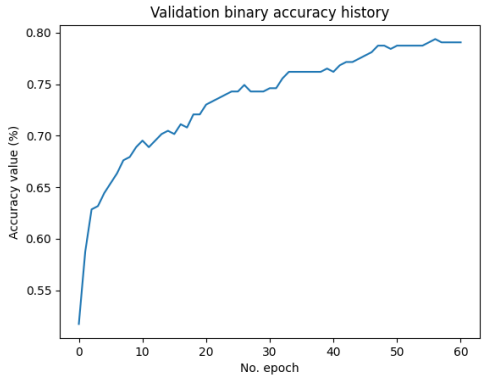
Model	With Augmented Data	Without Augmentation
Xception	81.65	81.65
VGG 16	86.08	86.08
ResNet 50	**90.82**	**87.66**
Inception v3	79.75	78.48
Inception ResNet v2	81.01	79.11

3.10 Evaluation of Results

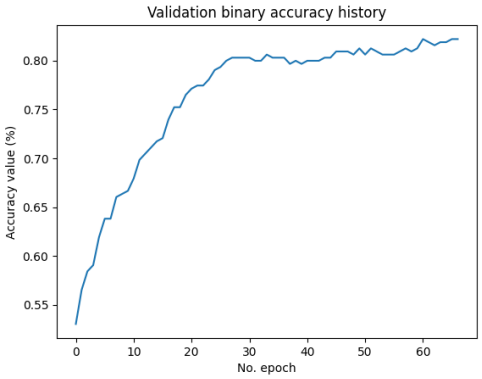
In this section results evaluated in this study are summarized and compared to other studies.

As stated in the previous chapters, sensitivity is one of the critical metrics used in medical studies. Early diagnosis plays a very important role in overcoming the disease. Incorrect predictions about the diagnosis of the disease lead to the detection of the disease process after it has worsened. Radiologists and doctors request more than one examination to diagnose the disease. As mentioned in the study of Bai et al, radiologists had high specificity scores however average sensitivity scores diagnosing COVID-19 patients using chest CT scan [13]. Radiologists stated by Bai et al. have sensitivity scores ranged from 70% to 94%. Most of the experimented deep learning models have higher sensitivity scores than the radiologists stated on Bai et al..

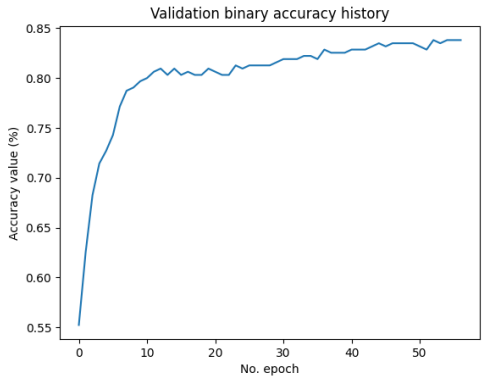
CT scan images from two data sets, Zhao's data set and SARS-CoV-2 CT Scan data set, are used in this study. Studies using these two data sets are compared in the Table 15. Among these studies, Soares et al. proposed an xDNN classification approach using SARS-CoV-2 CT Scan data set that



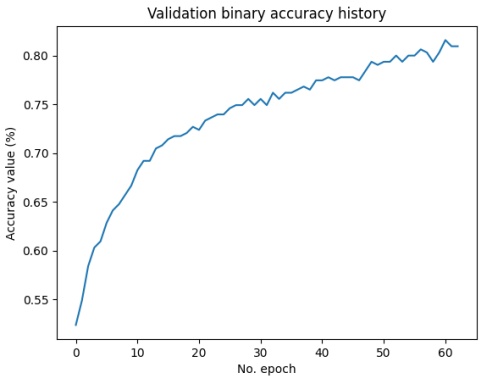
(a) Xception



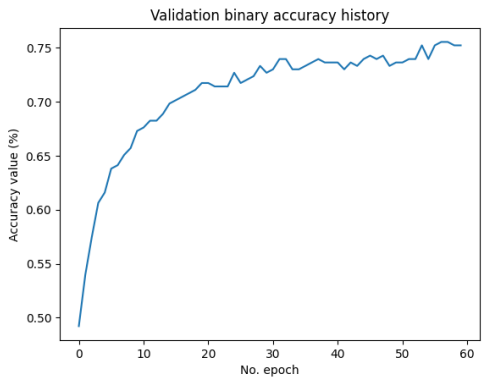
(b) VGG 16



(c) ResNet 50

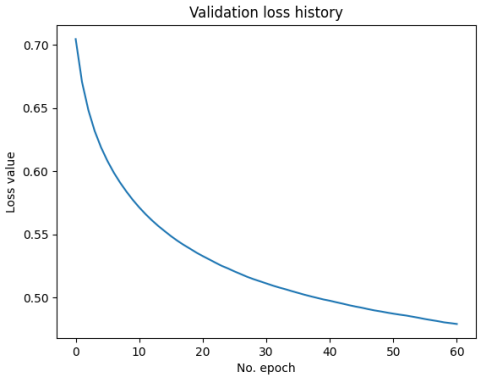


(d) Inception v3

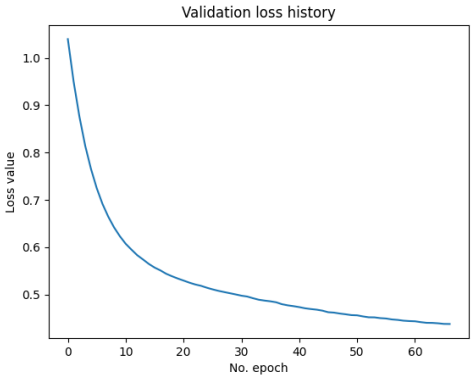


(e) Inception ResNet v2

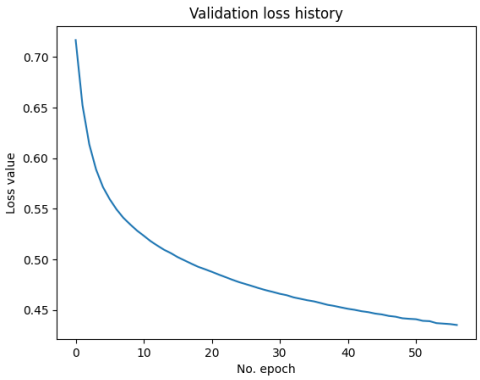
Figure 15: Validation accuracies of trained models using augmented data without trainable layers



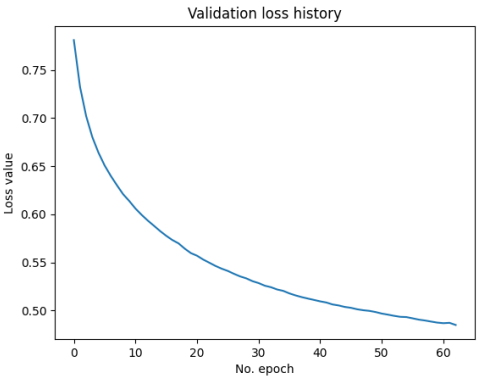
(a) Xception



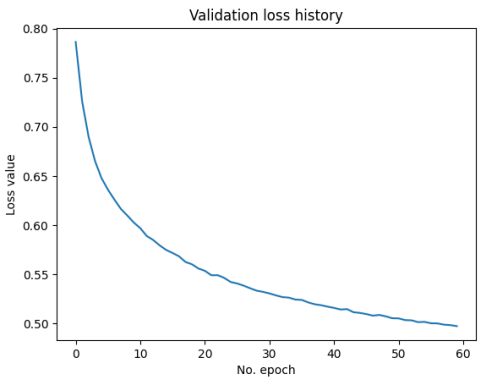
(b) VGG 16



(c) ResNet 50

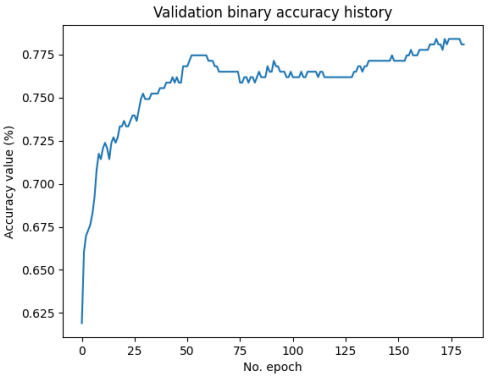


(d) Inception v3

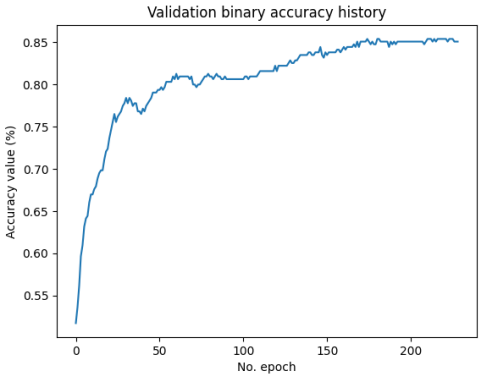


(e) Inception ResNet v2

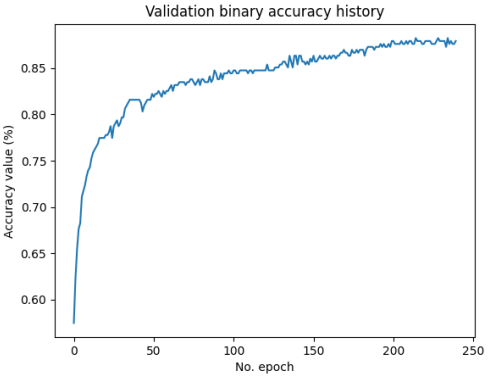
Figure 16: Validation losses of trained models using augmented data without trainable layers



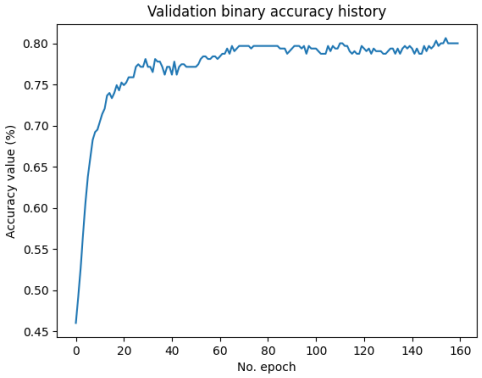
(a) Xception



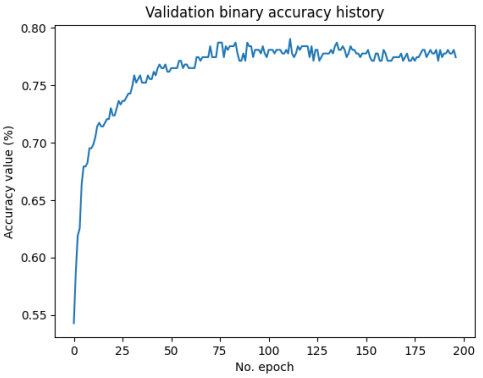
(b) VGG 16



(c) ResNet 50

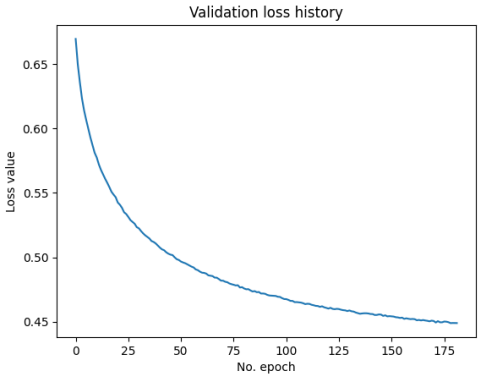


(d) Inception v3

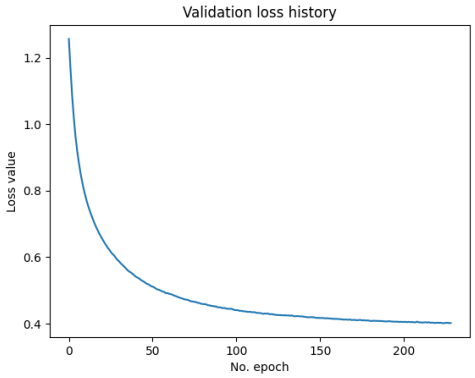


(e) Inception ResNet v2

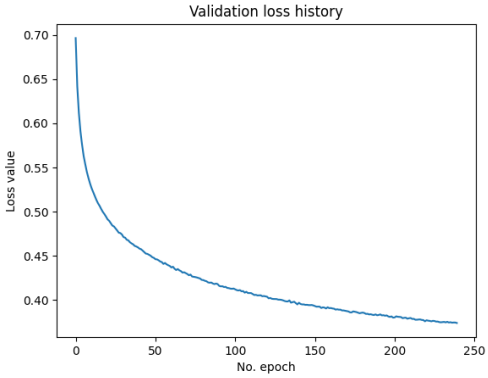
Figure 17: Validation accuracies of trained models using augmented data without trainable layers and with lower minimum delta value for early stopping callback function



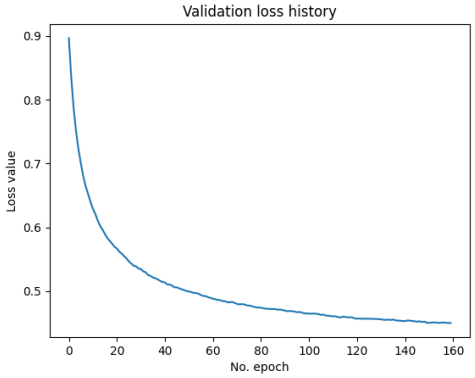
(a) Xception



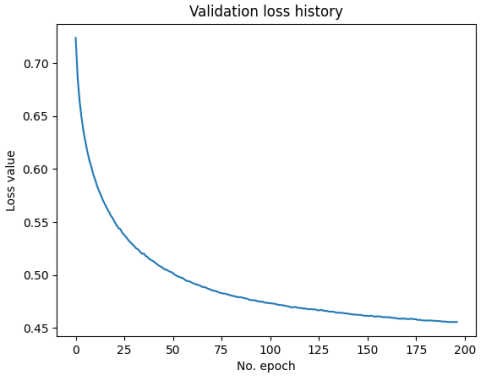
(b) VGG 16



(c) ResNet 50

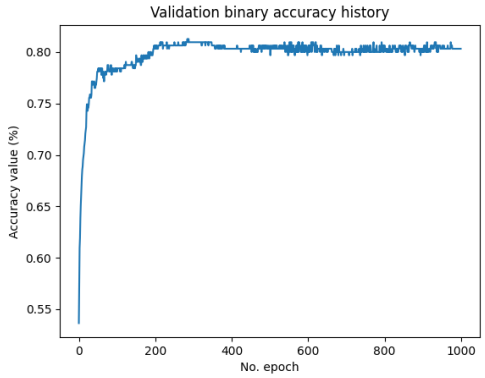


(d) Inception v3

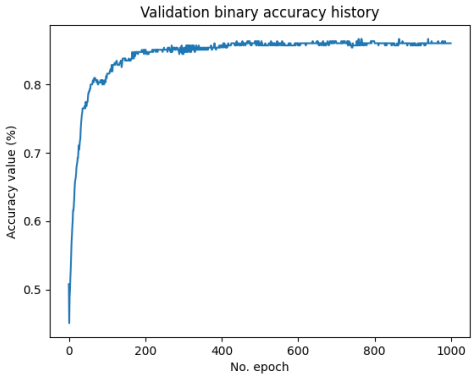


(e) Inception ResNet v2

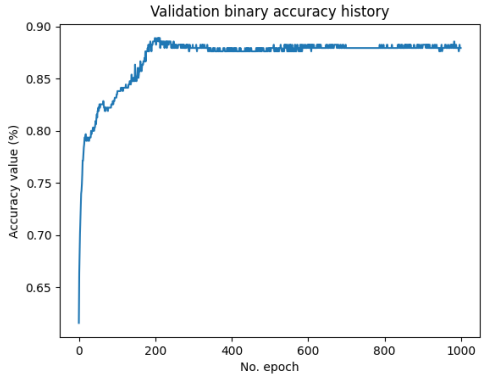
Figure 18: Validation losses of trained models using augmented data without trainable layers and with lower minimum delta value for early stopping callback function



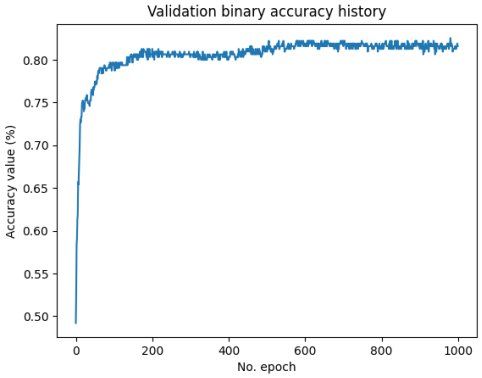
(a) Xception



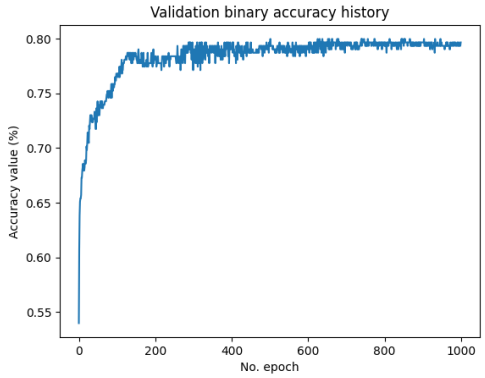
(b) VGG 16



(c) ResNet 50

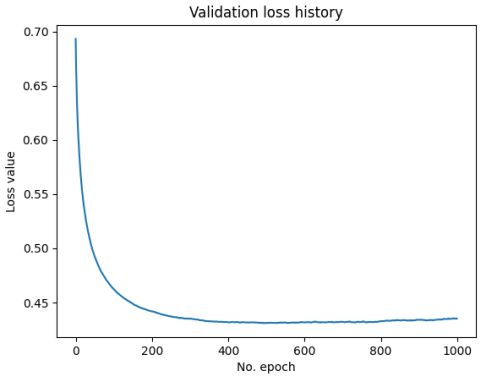


(d) Inception v3

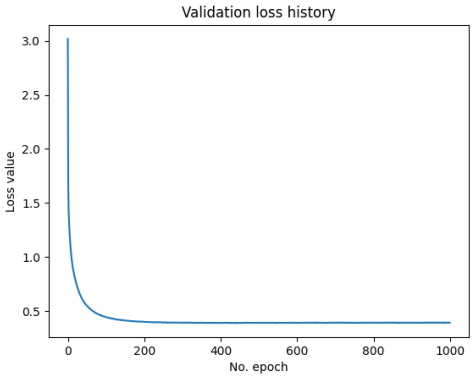


(e) Inception ResNet v2

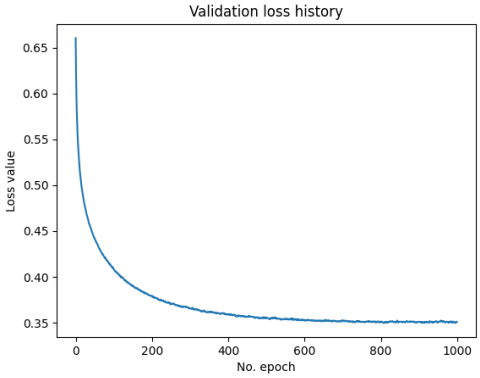
Figure 19: Validation accuracies of trained models using augmented data without trainable layers and without early stopping callback function (1000 epochs)



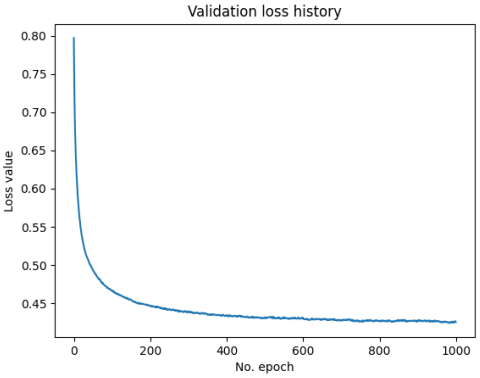
(a) Xception



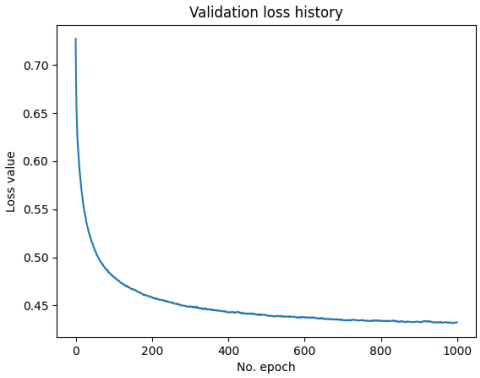
(b) VGG 16



(c) ResNet 50



(d) Inception v3



(e) Inception ResNet v2

Figure 20: Validation losses of trained models using augmented data without trainable layers and without early stopping callback function (1000 epochs)

shows highest performance with more than 97% of accuracy and sensitivity scores [30]. Using same data set, DenseNet201 based COVID-19 classification model is proposed by Jaiswal et al.[37]. This study performs 96.25% accuracy and 96.29% sensitivity scores. In studies using Zhao's data set, performance results are not as good as SARS-CoV-2 CT-scan data set. Among these studies, highest accuracy as 89% is performed by the study of Yang et al and the highest sensitivity score as 95% is reported in the study of Wu et al [29, 38].

Table 15: Studies in the literature including the data sets used in this study. Scores not specified in cited articles are shown as "-". **A**) Zhao's data set, **B**) SARS-CoV-2 CT scan data set.

Authors	Dataset	Method	Accuracy	Sensitivity
Soares et al. [30]	B	eXplainable Deep Learning classification approach, xDNN	97.38%	97.15%
Jaiswal et al. [37]	B	DenseNet201 based COVID-19 classification model	96.25%	96.29%
Yang et al. [29]	A	Multi-task learning and self-supervised learning	89.00%	-
Wu et al. [38]	A	Joint Classification and Segmentation System	-	95%
Mobiny et al. [39]	A	Detail-Oriented Capsule Networks	87.6	91.50%
Loey et al. [40]	A	CGAN based on a deep transfer learning model	82.91%	77.66%

CHAPTER 4

CONCLUSIONS

4.1 Discussion

In this work, comparison of five deep neural network architectures are presented. Discussion of the results is outlined in this section.

This study includes the comparison of the performances of different neural network models on COVID-19 detection using CT scan images. The effect of augmented data on performance is also noted among the comparisons made in this study. Since data is the most valuable element of studies in medical research, this study contains evaluation of multiple models between different data groups. As described in previous chapters, more accurate estimates are obtained in this study by using the multiple datasets combined. The models trained after the combined dataset was expanded with data augmentation techniques show the highest percentage results among all of the comparisons included in this study. As stated about the first series of runs using trainable layers on base models, significantly high scores are achieved. Accuracy scores of around 95% were obtained in all trials. When the accuracy and loss values of training and validation were examined, it was important to catch the overfitting situation. Overfitting is one of the major problems on deep learning studies, and the models are compared using frozen layers to overcome overfitting. As a result of this study, the evaluations of the runs using frozen layers in the basic models were examined. Overall, ResNet 50 network model achieved up to the COVID-19 detection results in obtaining sensitivity score of 89.38%, specificity score of 89.44%, precision score of 92.26% and accuracy score of 90.82%.

There is no significant finding explained in this study about the effect of batch size on performance of the models. Due to the large neural network models and limited hardware capacity on the working computer, the network models proposed in this study cannot be trained using higher batch size.

The reliability of the results is impacted by containing images extracted from previous studies. Images in Zhao's dataset may have been subject to possible distortion during extraction from pdf files. Image sizes are quite small compared to original CT images with high resolution.

The last runs in this study show the significant effect of limiting the number of iterations. When the number of iterations or epochs in a training model does not allow the model to converge sufficiently, iterations should continue. The same happens when the early stop function stops training before the model converges.

Since this study was completed in a limited time period, it was preferred to use the early stopping function in order to quickly conclude the runs. When this study reached a certain maturity, stretching or even removing the early stop function was tried. In order to increase the scores achieved after these trials, different values can be used for other hyperparameters. Loss and activation functions do not differ between runs, changing these parameters may affect training performance. Binary cross-entropy is mostly preferred in binary classification studies and diagnosis of COVID-19 from other diseases can be examined by using categorical classification. When training the deep learning model for a categorical classification problem, categorical cross-entropy loss function can be used as loss function.

4.2 Future Work

As described in previous chapters, neural network models trained using multiple datasets have more accurate predictions. Better test results can be achieved by working with more than two datasets. Since access to personal health data is limited, the number of public datasets studied is limited. Future works can also benefit from the datasets referenced in this study. The datasets combined in this study are publicly accessed from [29] and [30].

Considering that better performance results are obtained by using augmented data, further studies can be made by feeding network models with more augmented data. Apart from enlarging the data set, machine learning techniques can be studied on using the patient's symptoms such as fever, cough and other findings as input. The number of days passed through the RT-PCR test result and the time period of the disease can also be important inputs. Since different variations are encountered in different time periods, the research area can be expanded by examining the periodic differences in CT findings.

The results in this study show that ResNet 50 deep learning model has the best performance among the five models used in this study. As future work, other deep learning models can be tried to get higher performance measurements. Batches with lower than the size of 64 is used in this study. Since computation power is limited with the studied computer, network models can be re-trained with a better computer. With an increased video ram in GPU higher batch sizes can be examined in the future works.

With the widespread use of machine learning approaches, its use in the interdisciplinary field is becoming more and more popular. It is important to show the reasons for the outputs in areas such as economy, health, and education. At the forefront of the areas studied on this subject is explainable artificial intelligence (XAI). XAI aims to explain the reasons for the decisions made in the predictions made. In Phillips' study, it was mentioned that different types of explanations are required for different users and that in addition to the explanation, how accurate this explanation should be stated [41]. The machine learning approaches used in this study and many other approaches do not explain the reasons for the outputs, the focus is on achieving higher scores. In addition to aiming to reach higher scores in the studies, it is also very important to conduct studies that provide better explanations. For this need, which is not addressed in most of the machine learning and artificial intelligence approaches, a solution is sought with approaches such as explainable artificial intelligence.

In this study, CT scan images were used in RGB format. Although the CT images used in this study are grayscale, they are used in RGB format to feed the pre-trained models with three-channel data. Kanan et al. highlighted the importance of the RGB to grayscale conversion method [42]. As stated in their study, the method of converting color to grayscale affects the performance of image recognition

approaches, as demonstrated by multiple experiments using different datasets. CT images are originally scanned in grayscale format and there is no need to convert images from RGB. Since Kanan et al.'s study highlights the effect of multiple grayscale formatted versions of the same images in deep learning approaches, different deep learning approaches can be studied using CT images with single channels considering the effect of RGB to grayscale conversion.

Since the rapid increase of COVID-19 cases reaching pandemic size, computed tomography scanning has been used more and more. It was not predictable that such a large amount of data would be produced. The fact that so much data are being produced provides the appropriate conditions for increasing the popularity of studies on CT scans. All of the pre-trained convolutional neural network models included in this study emerged with the ImageNet dataset in 2015 and later. With the huge number of images to be collected from COVID-19 and other diseases, a CT dataset with a high number of images can be created. By working with a large number of images, new CNN models trained on these images can be created. The similarity of the images in the dataset used in transfer learning studies and the images in the dataset where the model was previously trained can be examined.

4.3 Conclusion

The performances of deep neural networks is compared in this study.

SARS-CoV-2-CT Scan dataset and Zhao's dataset are combined together and data augmentation techniques are applied on all images. As a pre-processing step, images are resized to provide the required dimensions to the input layers.

Performance for classification of the COVID-19 disease are investigated. The performance of the five deep neural network architectures are compared, Xception, VGG 16, ResNet 50, Inception v3, and Inception ResNet v2. ResNet 50 shows the best result for all of the metrics described in the results chapter. Since sensitivity is previously explained as one of the most valuable metrics on COVID-19 diagnosis results, the performances are compared using sensitivity and accuracy scores.

As mentioned in the previous chapter, this study shows the desired results when compared with similar studies. Since machine learning becomes more popular day by day on solving classification problems, the best accuracy score of 91% in this study shows about 2% of improvement on diagnosing COVID-19 disease on CT images compared to machine learning studies using Zhao's dataset in the literature. This tiny performance improvement in diagnosing COVID-19 disease, which has reached very large masses, is invaluable due to the number of people the virus has reached all over the world.

Training studies in deep neural network structures can be improved by changing many variables such as batch size, number of epochs, optimizer, loss function, activation function, and early stopping callback function. In this study, several runs were performed changing the training dataset, batch size, and early stopping function.

In summary, in this study comparisons are made about solving the binary classification problem on diagnosing COVID-19 using CT images. Since the data used in this study is limited, more accurate training performances are achieved on most of the architectures with data augmentation. Performance tests were performed on both augmented and non-augmented data. The results indicate that there is a significant performance improvement on using augmented data. However, batch sizes of more than

64 have not been examined, as the computation power is limited by the GPU of the studied computer. For the batch sizes lower than 64, experimented network models do not show any significant change. Among all of the test results, the highest sensitivity, specificity, precision, and accuracy scores are collected using ResNet 50 architecture.

REFERENCES

[1] "Worldometer information." `https://www.worldometers.info/coronavirus`. Reported on 1 July 2022.

[2] K. M. Das, J. A. Alkoteesh, J. Al Kaabi, T. Al Mansoori, A. J. Winant, R. Singh, R. Paraswani, R. Syed, E. M. Sharif, G. B. Balhaj, and E. Y. Lee, "Comparison of chest radiography and chest CT for evaluation of pediatric COVID-19 pneumonia: Does CT add diagnostic value?," *Pediatric Pulmonology*, vol. 56, pp. 1409–1418, June 2021.

[3] J. Song, H. Wang, Y. Liu, W. Wu, G. Dai, Z. Wu, P. Zhu, W. Zhang, K. W. Yeom, and K. Deng, "End-to-end automatic differentiation of the coronavirus disease 2019 (COVID-19) from viral pneumonia based on chest CT," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 47, pp. 2516–2524, Oct. 2020.

[4] M. Ferguson, R. ak, Y.-T. Lee, and K. Law, "Automatic localization of casting defects with convolutional neural networks," pp. 1726–1735, 12 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, Feb. 2017.

[7] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, July 2017.

[8] "World health organization. coronavirus disease 2019 (covid-19) situation report – 51. geneva (switzerland); 2020."

[9] J. Bedford, D. Enria, J. Giesecke, D. L. Heymann, C. Ihekweazu, G. Kobinger, H. C. Lane, Z. Memish, M.-d. Oh, A. A. Sall, A. Schuchat, K. Ungchusak, and L. H. Wieler, "COVID-19: towards controlling of a pandemic," *The Lancet*, vol. 395, pp. 1015–1018, Mar. 2020.

[10] W. Wang, Y. Xu, R. Gao, R. Lu, K. Han, G. Wu, and W. Tan, "Detection of SARS-CoV-2 in Different Types of Clinical Specimens," *JAMA*, vol. 323, pp. 1843–1844, May 2020.

[11] A. Bhargava, A. Bansal, and V. Goyal, "Machine learning-based automatic detection of novel coronavirus (COVID-19) disease," *Multimedia Tools and Applications*, vol. 81, no. 10, pp. 13731–13750, 2022.

[12] S. A. Duzgun, G. Durhan, F. B. Demirkazik, M. G. Akpinar, and O. M. Ariyurek, "COVID-19 pneumonia: the great radiological mimicker," *Insights into Imaging*, vol. 11, p. 118, Dec. 2020.

[13] H. X. Bai, B. Hsieh, Z. Xiong, K. Halsey, J. W. Choi, T. M. L. Tran, I. Pan, L.-B. Shi, D.-C. Wang, J. Mei, X.-L. Jiang, Q.-H. Zeng, T. K. Egglin, P.-F. Hu, S. Agarwal, F.-F. Xie, S. Li, T. Healey, M. K. Atalay, and W.-H. Liao, "Performance of radiologists in differentiating covid-19 from non-covid-19 viral pneumonia at chest ct," *Radiology*, vol. 296, no. 2, pp. E46–E54, 2020. PMID: 32155105.

[14] K. Fukushima, "Neocognitron," *Scholarpedia*, vol. 2, no. 1, p. 1717, 2007. revision #91558.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," p. 30.

[16] M. D. Bloice, P. M. Roth, and A. Holzinger, "Biomedical image augmentation using Augmentor," *Bioinformatics*, vol. 35, pp. 4522–4524, Nov. 2019.

[17] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Apr. 2015. arXiv:1409.1556 [cs].

[18] S. Basodi, C. Ji, H. Zhang, and Y. Pan, "Gradient amplification: An efficient way to train deep neural networks," *Big Data Mining and Analytics*, vol. 3, pp. 196–207, Sep. 2020.

[19] 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*, vol. 56, p. 2000775, Aug. 2020.

[20] M. Abd Elaziz, M. A. A. Al-qaness, E. O. Abo Zaid, S. Lu, R. Ali Ibrahim, and A. A. Ewees, "Automatic clustering method to segment COVID-19 CT images," *PLOS ONE*, vol. 16, p. e0244416, Jan. 2021.

[21] J. Ma, Z. Nie, C. Wang, G. Dong, Q. Zhu, J. He, L. Gui, and X. Yang, "Active contour regularized semi-supervised learning for COVID-19 CT infection segmentation with limited annotations," *Physics in Medicine & Biology*, vol. 65, p. 225034, Nov. 2020.

[22] Z. Yu, X. Li, H. Sun, J. Wang, T. Zhao, H. Chen, Y. Ma, S. Zhu, and Z. Xie, "Rapid identification of COVID-19 severity in CT scans through classification of deep features," *BioMedical Engineering OnLine*, vol. 19, p. 63, Dec. 2020.

[23] L. Berta, C. De Mattia, F. Rizzetto, S. Carrazza, P. Colombo, R. Fumagalli, T. Langer, D. Lizio, A. Vanzulli, and A. Torresin, "A patient-specific approach for quantitative and automatic analysis of computed tomography images in lung disease: Application to COVID-19 patients," *Physica Medica*, vol. 82, pp. 28–39, Feb. 2021.

[24] F. Xiong, Y. Wang, T. You, H. h. Li, T. t. Fu, H. Tan, W. Huang, and Y. Jiang, "The clinical classification of patients with COVID-19 pneumonia was predicted by Radiomics using chest CT," *Medicine*, vol. 100, p. e25307, Mar. 2021.

[25] R. Wilson and A. Devaraj, "Radiomics of pulmonary nodules and lung cancer," *Translational Lung Cancer Research*, vol. 6, pp. 86–91, Feb. 2017.

[26] M. Turkoglu, "COVID-19 Detection System Using Chest CT Images and Multiple Kernels-Extreme Learning Machine Based on Deep Neural Network," *IRBM*, vol. 42, pp. 207–214, Aug. 2021.

[27] D.-N. Le, V. S. Parvathy, D. Gupta, A. Khanna, J. J. P. C. Rodrigues, and K. Shankar, "IoT enabled depthwise separable convolution neural network with deep support vector machine for COVID-19 diagnosis and classification," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 3235–3248, Nov. 2021.

[28] D. Sharifrazi, R. Alizadehsani, M. Roshanzamir, J. H. Joloudari, A. Shoeibi, M. Jafari, S. Hussain, Z. A. Sani, F. Hasanzadeh, F. Khozeimeh, A. Khosravi, S. Nahavandi, M. Panahiazar, A. Zare, S. M. S. Islam, and U. R. Acharya, "Fusion of convolution neural network, support vector machine and Sobel filter for accurate detection of COVID-19 patients using X-ray images," *Biomedical Signal Processing and Control*, vol. 68, p. 102622, July 2021.

[29] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang, and P. Xie, "COVID-CT-Dataset: A CT Scan Dataset about COVID-19," June 2020. arXiv:2003.13865 [cs, eess, stat].

[30] E. Soares, P. Angelov, S. Biaso, M. H. Froes, and K. Abe, "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification," p. 8.

[31] "https://www.medrxiv.org."

[32] "https://www.biorxiv.org."

[33] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur, "Classification of the covid-19 infected patients using densenet201 based deep transfer learning," *Journal of Biomolecular Structure and Dynamics*, vol. 39, no. 15, pp. 5682–5689, 2021. PMID: 32619398.

[34] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017. arXiv:1412.6980 [cs].

[35] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, and G. Jamalipour Soufi, "Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning," *Medical Image Analysis*, vol. 65, p. 101794, Oct. 2020.

[36] F. Chollet and others, "Keras," 2015.

[37] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur, "Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning," *Journal of Biomolecular Structure and Dynamics*, vol. 39, pp. 5682–5689, Oct. 2021.

[38] Y.-H. Wu, S.-H. Gao, J. Mei, J. Xu, D.-P. Fan, R.-G. Zhang, and M.-M. Cheng, "JCS: An Explainable COVID-19 Diagnosis System by Joint Classification and Segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 3113–3126, 2021. arXiv:2004.07054 [cs, eess].

[39] A. Mobiny, P. A. Cicalese, S. Zare, P. Yuan, M. Abavisani, C. C. Wu, J. Ahuja, P. M. de Groot, and H. Van Nguyen, "Radiologist-Level COVID-19 Detection Using CT Scans with Detail-Oriented Capsule Networks," Apr. 2020. arXiv:2004.07407 [cs, eess].

[40] M. Loey, G. Manogaran, and N. E. M. Khalifa, "A deep transfer learning model with classical data augmentation and CGAN to detect COVID-19 from chest CT radiography digital images," *Neural Computing and Applications*, Oct. 2020.

[41] P. J. Phillips, C. A. Hahn, P. C. Fontana, A. N. Yates, K. Greene, D. A. Broniatowski, and M. A. Przybocki, "Four Principles of Explainable Artificial Intelligence," tech. rep., National Institute of Standards and Technology, Sept. 2021.

[42] C. Kanan and G. W. Cottrell, “Color-to-Grayscale: Does the Method Matter in Image Recognition?,” *PLoS ONE*, vol. 7, p. e29740, Jan. 2012.

APPENDIX A

CONFUSION MATRICES OF EACH EXPERIMENTED NETWORK

Table A.1: Confusion Matrix of Xception Using Augmented Data with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	150	10
	Negative	4	152

Table A.2: Confusion Matrix of Xception Using Augmented Data with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	153	7
	Negative	2	154

Table A.3: Confusion Matrix of Xception Using Augmented Data with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	150	10
	Negative	3	153

Table A.4: Confusion Matrix of Xception Using Without Augmentation with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	151	9
	Negative	3	153

Table A.5: Confusion Matrix of Xception Without Data Augmentation with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	151	9
	Negative	7	149

Table A.6: Confusion Matrix of Xception Without Data Augmentation with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	151	9
	Negative	6	150

Table A.7: Confusion Matrix of VGG 16 Using Augmented Data with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	154	6
	Negative	5	151

Table A.8: Confusion Matrix of VGG 16 Using Augmented Data with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	154	6
	Negative	3	153

Table A.9: Confusion Matrix of VGG 16 Using Augmented Data with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	151	9
	Negative	2	154

Table A.10: Confusion Matrix of VGG 16 Without Data Augmentation with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	154	6
	Negative	3	153

Table A.11: Confusion Matrix of VGG 16 Without Data Augmentation with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	156	4
	Negative	14	142

Table A.12: Confusion Matrix of VGG 16 Without Data Augmentation with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	153	7
	Negative	4	152

Table A.13: Confusion Matrix of ResNet 50 Using Augmented Data with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	152	8
	Negative	4	152

Table A.14: Confusion Matrix of ResNet 50 Using Augmented Data with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	153	7
	Negative	3	153

Table A.15: Confusion Matrix of ResNet 50 Using Augmented Data with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	153	7
	Negative	5	151

Table A.16: Confusion Matrix of ResNet 50 Without Data Augmentation with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	151	9
	Negative	11	145

Table A.17: Confusion Matrix of ResNet 50 Without Data Augmentation with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	153	7
	Negative	3	153

Table A.18: Confusion Matrix of ResNet 50 Without Data Augmentation with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	149	11
	Negative	5	151

Table A.19: Confusion Matrix of Inception v3 Using Augmented Data with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	158	2
	Negative	2	154

Table A.20: Confusion Matrix of Inception v3 Using Augmented Data with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	151	9
	Negative	7	149

Table A.21: Confusion Matrix of Inception v3 Using Augmented Data with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	153	7
	Negative	3	153

Table A.22: Confusion Matrix of Inception v3 Without Data Augmentation with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	152	8
	Negative	2	154

Table A.23: Confusion Matrix of Inception v3 Without Data Augmentation with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	151	9
	Negative	8	148

Table A.24: Confusion Matrix of Inception v3 Without Data Augmentation with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	152	8
	Negative	12	144

Table A.25: Confusion Matrix of Inception ResNet v2 Using Augmented Data with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	153	7
	Negative	6	150

Table A.26: Confusion Matrix of Inception ResNet v2 Using Augmented Data with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	155	5
	Negative	5	151

Table A.27: Confusion Matrix of Inception ResNet v2 Using Augmented Data with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	154	6
	Negative	7	149

Table A.28: Confusion Matrix of Inception ResNet v2 Without Data Augmentation with 32 Batch Size

		Predicted	
		Positive	Negative
True	Positive	153	7
	Negative	11	145

Table A.29: Confusion Matrix of Inception ResNet v2 Without Data Augmentation with 48 Batch Size

		Predicted	
		Positive	Negative
True	Positive	148	12
	Negative	5	151

Table A.30: Confusion Matrix of Inception ResNet v2 Without Data Augmentation with 64 Batch Size

		Predicted	
		Positive	Negative
True	Positive	150	10
	Negative	8	148

TEZ İZİN FORMU / THESIS PERMISSION FORM

ENSTİTÜ / INSTITUTE

Fen Bilimleri Enstitüsü / Graduate School of Natural and Applied Sciences [ ]

Sosyal Bilimler Enstitüsü / Graduate School of Social Sciences [ ]

Uygulamalı Matematik Enstitüsü / Graduate School of Applied Mathematics [ ]

Enformatik Enstitüsü / Graduate School of Informatics [X]

Deniz Bilimleri Enstitüsü / Graduate School of Marine Sciences [ ]

YAZARIN / AUTHOR

Soyadı / Surname : Sarıoğlu
Adı / Name : Mehmet Tunahan
Bölümü / Department : Sağlık Bilişimi Anabilim Dalı/ Department of Health Informatics

TEZİN ADI / TITLE OF THE THESIS (**İngilizce** / English) : A COMPARISON OF DEEP NEURAL NETWORK ARCHITECTURES FOR COVID-19 DETECTION USING CT CHEST IMAGES

TEZİN TÜRÜ / DEGREE: Yüksek Lisans / Master [X] **Doktora** / PhD [ ]

1. **Tezin tamamı dünya çapında erişime açılacaktır. /** Release the entire work immediately for access worldwide. [X]
2. **Tez iki yıl süreyle erişime kapalı olacaktır.** / Secure the entire work for patent and/or proprietary purposes for a period of **two year. *** [ ]
3. **Tez altı ay süreyle erişime kapalı olacaktır.** / Secure the entire work for period of **six months**. ***** [ ]

***** *Enstitü Yönetim Kurulu Kararının basılı kopyası tezle birlikte kütüphaneye teslim edilecektir.*
A copy of the Decision of the Institute Administrative Committee will be delivered to the library together with the printed thesis.

Yazarın imzası / Signature ..... **Tarih** / Date .....