CLASSIFICATION OF COVID-19 FROM DIFFERENT COMPUTED TOMOGRAPHY IMAGE DATASETS WITH MACHINE LEARNING AND DEEP LEARNING METHODS

A MASTER THESIS SUBMITTED TO

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

OF

KOCAELI UNIVERSITY

BY

FARID AL-AREQI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF MASTER OF SCIENCE

IN

BIOMEDICAL ENGINEERING

KOCAELİ 2022

CLASSIFICATION OF COVID-19 FROM DIFFERENT COMPUTED TOMOGRAPHY IMAGE DATASETS WITH MACHINE LEARNING AND DEEP LEARNING METHODS

A THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
KOCAELİ UNIVERSITY

BY

FARID AL-AREQI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
BIOMEDICAL ENGINEERING

Assist. Prof. Dr. Mehmet Zeki KONYAR
Supervisor, Kocaeli University ....
Assist. Prof. Dr. Faruk AKTAŞ
Jury member, Kocaeli University ....
Assist. Prof. Dr. Mücahid BARSTUĞAN
Jury member, Konya Technical University ....

Thesis Defense Date: 22.06.2022

ETIK BEYAN VE ARASTIRMA FONU DESTEGI

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez/proje çalışmasında,

- Bu tezin/projenin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmanın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin/Projenin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez/proje çalışması olarak sunmadığımı,

beyan ederim.

☐ Bu tez/proje çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

☒ Bu tez/proje çalışması kapsamında üretilen veri ve bilgiler Kocaeli Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi tarafından 2021-2751 no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Farid AL-AREQI

YAYIMLAMA VE FIKRI MÜLKIYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin/projemin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin/projemin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanımı bana ait olacaktır.

Tezin/projenin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin/projenin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim kurulu tarafından yayınlanan "***Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge***" kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

☐ Enstitü yönetim kurulu kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.

☐ Enstitü yönetim kurulu gerekçeli kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.

☒ Tezim/projem ile ilgili gizlilik kararı verilmemiştir.

Farid AL-AREQI

PREFACE AND ACKNOWLEDGEMENTS

The thesis's main objective is to classify COVID-19 from CT scan images using machine learning and deep learning techniques. The study also included a comparison between machine learning and deep learning techniques on different public COVID-19 datasets.

During the work of this dissertation, both support and guidance were provided by my beloved supervisor Assist. Prof. Dr. Mehmet Zeki KONYAR despite his restricted schedule. I would like to thank him for all the support and effort he has put in.

I equally thank all the Biomedical and Mechatronics departments staff who in one way or the other assisted me in the process of my courses and thesis. Besides, I would like to express my sincere gratitude to Turkey's Presidency for Turks Abroad and Related Communities (YTB) for providing me with educational opportunities. And finally, I want to express my gratitude to my beloved Father FUAD and my beloved mother WAFAA for all the support they gave me throughout my entire life.
This study was supported by Kocaeli University Research Fund. Project Number: 2021-2751.

June – 2022 Farid AL-AREQI

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTS OF SYMBOLS AND ABBREVIATIONS

Abbreviations

+ssRNA	: Positive Single-Stranded Ribonucleic Acid
AI	: Artificial Intelligence
ANN	: Artificial Neural Networks
AUC	: Area Under the Curve
CAP	: Community-Acquired Pneumonia
CNN	: Convolutional Neural Networks
COVID-19	: Corona Virus Disease 2019
CT	: Computed Tomography
CV	: Cross-Validation
DL	: Deep Learning
DT	: Decision Tree
ELISA	: Enzyme-Linked Immunosorbent Assay
FC	: Fully Connected
FOS	: First-Order Statistics
GLCM	: Gray Level Co-Occurrence Matrix
GLDM	: Gray Level Dependence Matrix
GLRLM	: Gray Level Run Length Matrix
GLSZM	: Gray Level Size Zone Matrix
IAVP	: Influenza-A Viral Pneumonia
k-NN	: k-Nearest Neighbors
LR	: Logistic Regression
ML	: Machine Learning
MRI	: Magnetic Resonance Imaging
NGTDM	: Neighboring Gray Tone Difference Matrix
NN	: Neural Networks
PET	: Positron Emission Tomography
RBF	: Radial Basis Function
ReLU	: Rectified Linear Unit
RF	: Random Forest
ROC	: Receiver Operating Characteristic
ROI	: Region Of Interest
RT-PCR	: Reverse Transcription Polymerase Chain Reaction
SARS	: Severe Acute Respiratory Syndrome
SARS-CoV-2	: Severe Acute Respiratory Syndrome Corona Virus 2
SVM	: Support Vector Machines
WHO	: World Health Organization
XGBoost	: eXtreme Gradient Boosting

FARKLI BİLGİSAYARLI TOMOGRAFİ GÖRÜNTÜ VERİ KÜMELERİNDEN MAKİNE ÖĞRENME VE DERİN ÖĞRENME YÖNTEMLERİYLE COVID-19 SINIFLANDIRILMASI

ÖZET

Covid-19 hastalığının hızlı teşhisi, enfeksiyonu önlemek için kritik öneme sahiptir. Yapay zekâ tabanlı birçok yöntemin radyologlara önemli bir yardım sağladığı kanıtlanmıştır. Bu tezde, bilgisayarlı tomografi (BT) tarama görüntülerinden COVID-19'u sınıflandırmak için makine öğrenmesi yöntemleri ve transfer öğrenme modelleri önerilmiş ve karşılaştırılmıştır. Makine öğrenmesi yöntemleri ile akciğere ait 746 eksenel BT görüntüsü ile COVID-CT veri seti; 349 COVID-19 (pozitif) ve 397 Covid-19 olmayan (negatif) kullanılmıştır. Şekil öznitelikleri, birinci dereceden istatistiksel öznitelikler ve gri seviye doku öznitelikleri gibi radiomiks özellikler BT görüntülerinden çıkarılmıştır. Model eğitimi için öznitelik vektörü, bir öznitelik grubu veya birden fazla grubun bir kombinasyonu ile oluşturulmuştur. Daha sonra Destek Vektör Makinesi, Rastgele Orman, k-en yakın komşular ve XGBoost sınıflandırıcı modellerini kullanarak sınıflandırılmıştır. Modellerin en iyi hiperparametreleri, GridSearchCV tekniği kullanılarak tuning modelleri ile seçilmiştir. Transfer öğrenme modelleri ile iki farklı COVID-19 CT veri seti kullanılmıştır. Sınıflandırma için, VGG-19, ResNet-50, DenseNet-201, Xception ve EfficientNet-B0 olmak üzere beş tane önceden eğitilmiş transfer öğrenme modeli kullanılmıştır. Modeller, Adam optimize yöntemi ile 50 tur için iki farklı COVID-19 CT veri kümesinde (Large COVID-19 CT scan slice dataset and CT Scans for COVID-19 Classification) ince ayarlanıp eğitilmiştir. Makine öğrenimi ve derin öğrenme modellerini değerlendirmek için her veri kümesinden bağımsız bir test seti kullanılmıştır. Birinci dereceden istatistik özelliklerine sahip XGBoost sınıflandırıcı ile makine öğrenme yöntemlerinin en iyi doğruluğu %98,65 iken, VGG-19 ve ResNet-50 sırasıyla %98,04 ve %99,62 ile birinci ve ikinci veri setlerinde en iyi doğruluğu elde etmiştir.

Anahtar Kelimeler: BT Taraması, COVID-19, Makine Öğrenimi, Radiomiks, Transfer Öğrenimi.

CLASSIFICATION OF COVID-19 FROM DIFFERENT COMPUTED TOMOGRAPHY IMAGE DATASETS WITH MACHINE LEARNING AND DEEP LEARNING METHODS

ABSTRACT

Rapid diagnosis of the Covid-19 disease is critical to prevent infection. Many artificial intelligence-based methods have proved to provide an important assist to radiologists. In this thesis, machine learning methods and transfer learning models are proposed and compared to classify COVID-19 from CT scan images. With the machine learning methods, COVID-CT dataset with 746 axial computed tomography (CT) images of the lung; 349 COVID-19 (positives) and 397 non-Covid-19 (negative) was used. Radiomics features such as shape, first-order statistical features, and Gray-level texture features were extracted from the CT images. The feature vector for model training is constructed with one feature group or a combination of more than one group. We then classified using Support Vector Machine, Random Forest, k-nearest neighbors, and XGBoost classifier models. The best hyperparameters of the models were chosen by the tuning models using the GridSearchCV technique. With Transfer learning models, we used two different COVID-19 CT datasets. For classification, five transfer learning pre-trained models namely VGG-19, ResNet-50, DenseNet-201, Xception, and EfficientNet-B0 were used. The models were fine-tuned and trained on two different COVID-19 CT datasets (Large COVID-19 CT scan slice dataset and CT Scans for COVID-19 Classification) for 50 epochs with an Adam optimizer. An independent test set from each dataset was used to evaluate the machine learning and deep learning models. The best accuracy of machine learning methods was 98.65% by XGBoost classifier with first-order statistics features, while VGG-19 and ResNet-50 achieved the best accuracy on the first and the second datasets of 98.04% and 99.62% respectively.

Keywords: CT Scan, COVID-19, Machine Learning, Radiomics, Transfer Learning.

1. INTRODUCTION

Severe Acute Respiratory Syndrome (SARS) or coronavirus is a virus from the RNA virus's family that affect the respiratory system and can be transmitted by droplets through the air when someone coughs, sneezes, or talks at a close distance with another person, or through touching the eyes or nose after touching an infected surface(Tang et al., 2020). The virus first emerged from Wuhan China in December 2019 and spread quickly to the rest of the world causing an outbreak. The World Health Organization (WHO) have declared COVID-19 as a pandemic on 2020 11th of March after it hit deaths of 4,291(WHO, 2020). Governments worldwide started to make critical actions and procedures to limit the propagation of the virus and protect people from getting infected. The full lockdown was the first procedure to be executed since the virus spread through the air. People also committed to wearing medical masks and keeping a social distance of 2 meters to decrease the rate of infections. Washing hands for 20 seconds and using hand sanitizers that contain at least 60% alcohol have been shown to kill the virus(UNICEF, 2021). Fever was reported in 90% of cases, weakness and dry cough in 80% of cases, shortness of breath in 20%, and respiratory distress in 15%, according to (Hui et al., 2020). Every country was racing to develop an effective viral vaccine, as well as faster, more efficient, and more reliable techniques and tests for diagnosing the virus. Traditional procedures, such as antigen-based ELISA and nucleic acid-based RT-PCR tests are the most common tests for COVID-19 detection. However, they are time and money-consuming and have a significant potential for false positives or false negatives(Tahamtan & Ardebili, 2020). Furthermore, lateral flow assays are quick and inexpensive to perform but have low sensitivity(Xu et al., 2020). Although medical images have long been considered a source of information used in the diagnosis and treatment of conditions and diseases, particularly respiratory illnesses, non-expert radiologists can sometimes experience difficulties in diagnosing the infection (S. Wang et al., 2021). Artificial intelligence techniques have significantly assisted radiologists by showing promising results in detection and classification of COVID-19 from radiological images such as x-ray and computed tomography (CT)(Jin et al., 2020). According to recent studies, sensitivity and specificity for ELISA are in the range of 74-100 and 78-100 respectively. While for RT-PCR it was 77.7 and 98.8 for sensitivity and specificity respectively(La Marca et al., 2020). In contrast, many studies that used machine learning (ML) and deep learning (DL) algorithms to classify COVID-19 from CT images have reported higher sensitivity and specificity in the range of 80-100(Ahamed et al., 2021; Sharma, 2020; S. Wang et al., 2021).

1.1. Literature Review

1.1.1. Machine Learning Studies

A CAD system was proposed by (Abbasian Ardakani et al., 2021) to differentiate COVID-19 pneumonia patients from patients who did not have COVID-19 pneumonia. They used a data set that contained 612 CT images, with 306 for COVID-19 patients and 306 belonging to non-COVID-19 patients. They extracted 20 radiological features and then fed those features into 5 distinct classifiers, namely Decision Tree (DT), k- nearest neighbor (k-NN), Naive Bayes, Support Vector Machine (SVM), and Ensemble. They were able to obtain the best possible result by utilizing the suggested ensemble classifier, which had an accuracy of 91.94 %.

(Al-Karawi et al., 2020) suggested a technique to auto assess the COVID-19 in CT images. The data set that they used includes 275 cases that tested positive for COVID-19 and 195 cases that tested negative. They have applied the Gabor filter to the CT images after performing the Fast Fourier Transform on them. SVM was utilized to carry out the classification procedure, and the results showed a 95.37% accuracy.

(Barstugan et al., 2020) used a total of 150 CT images and divided them into 4 different subsets with patch regions of 16x16, 32x32, 48x48, and 64x64 respectively. Radiomics features such as FOS, GLCM, GLRLM, and GLSZM along with DWT features have been extracted and then classified using SVM. Subset 3 with 255 non-infected and 306 infected 48x48 patches gave the best accuracy of 99.64% with DWT features using 10-fold cross-validation.

(Dey et al., 2020) used a total of 400 CT images, including 200 for patients with COVID-19 and another 200 for patients who did not have COVID-19. A CAD system that first segments the infected regions of COVID-19 CT images and then extracts features from those segments was proposed. They utilized Random Forest, k-Nearest Neighbors (k-

NN), Support Vector Machine, and Decision Tree as their four different classifiers for the classification process. They achieved an 87 % accuracy by using the k-NN classifier.

In their study, (Liu et al., 2020) extracted 34 statistical texture features from a total of 61 COVID-19 CT images as well as 27 general pneumonia CT images. They utilized an ensemble of bagged trees and compared it with four different classifiers, including LR, SVM, DT, and k-NN. They achieved the highest score of classification accuracy with the ensemble of bagged trees classifier, which was 94.16 %.

(Özkaya et al., 2020) chose the same dataset that (Barstugan et al., 2020) used and divided it into two Subsets with patch regions of (16x16 and 32x32). They utilized three pre-trained models such as VGG-16, GoogleNet, and ResNet-50 to extract features using the convolutional layers within them. The extracted features were fused and ranked and then classified using SVM. They achieved 98.27% accuracy with the proposed method on Subset-2.

(Kassania et al., 2021) as well extracted features using pre-trained models. A dataset with a total of 274 x-ray and CT images (117 x-rays and 20 CT as positive, 117 x-rays and 20 CT as negative) was used. Methods such as Decision Tree, Random Forest, XGBoost, AdaBoost, Bagging, and LightGBM were utilized to classify the features. Bagging tree as a classifier was able to achieve an accuracy of 99 % on the features extracted by DenseNet121.

There were 1658 cases of COVID-19-positive pneumonia and 1027 cases of community-acquired pneumonia in the dataset used by (Shi et al., 2021). They segmented the infected regions and extracted handcrafted location-specific features, conventional CT severity scores, and radiomics features. Several classification methods, including (SVM), (LR), (NN), and (RF) have been tested against RF and LightGBM-based method that was proposed. On handcrafted features, the proposed method obtained the best accuracy score of 89.4%.

(Saeedi et al., 2020) used many transfer learning models to extract features from the COVID-CT dataset. The models utilized for the feature extraction are DenseNet121, ResNet-50 V1&V2, InceptionV3, and MobileNet V1&V2. The features were classified using Nu-SVM classifier that achieved a 90.61% accuracy on the features extracted by DenseNet121.

1.1.2. Deep Learning Studies

(Alsharman & Jawarneh, 2020) have used a public dataset, namely COVID-CT, consisting of 746 CT images; 349 of which are COVID-19 cases and 397 are non-COVID-19 cases. They used GoogleNet model to automatically extract the features and classify them. The model achieved an accuracy of 82.14%.

(Pham, 2020) has also used COVID-CT dataset to train 16 pre-trained CNN model which are SqueezeNet, GoogLeNet, Inception-v3, DenseNet-201, MobileNet-v2, ResNet-18, ResNet-50, ResNet-101, Xception, Inception-ResNet-v2, ShuffleNet, NasNet-Mobile, NasNet-Large, AlexNet, VGG-16, and VGG-19. The best result was 96.20% achieved by DenseNet-201.

In (Polsinelli et al., 2020) study, they proposed a light CNN model based on SqueezeNet to classify COVID-19 from the COVID-CT dataset. The proposed model achieved an accuracy of 85.03%

(Santa Cruz, 2021) has used an ensemble method to combine many pre-trained models such as VGG-16, ResNet-50, Wide ResNet-50-2, DenseNet161, DenseNet169, and Inception v3 and build a more robust model. Their proposed method obtained an 86.7% accuracy on the COVID-CT dataset.

(Abd Elaziz et al., 2021) used two datasets; COVID-CT and x-ray images dataset. A model based on MobileNetV3 with Aquila Optimizer was proposed. The proposed model achieved a 78.3% accuracy.

(Sakagianni et al., 2020) have designed a NAS-based model using Google Cloud AutoML Vision. The model was trained and tested on the COVID-CT dataset. It achieved an 88.31% accuracy.

(Shaik & Cherukuri, 2022) have created a strong model by utilizing an ensemble method. The model was built from VGG-16, VGG-19, InceptionV3, ResNet-50, ResNet-50V2, InceptionResNetV2, Xception, and MobileNet models. The proposed model achieved a 93.33% accuracy on the COVID-CT dataset.

(JavadiMoghaddam & Gholamalinejad, 2021) proposed a deep learning model named (Wavelet CNN-4) that has Wavelet and four convolution layers; pooling and the Squeeze Excitation Block (SE-block) layer in its pooling layer. They compared the proposed model with models like VGG11, ResNet18, ResNet-50, and Inception-v3. The proposed model achieved an accuracy of 99.03%.

(Balaha et al., 2020) proposed a pre-trained models-based hybrid learning and optimization approach to detect COVID-19. They combined three publicly available datasets and applied data augmentation. Weighted Sum Method (WSM) was used as an evaluation metric to compare the combinations of the models. The best accuracy obtained was 99.33% by VGG-19.

(Neha et al., 2021) proposed a CNN-based model and logistic regression-based model. They used two datasets; chest X-ray and CT scan images datasets. The proposed CNN model achieved 99.10% on the CT dataset.

(Ahamed et al., 2021) used in their study chest X-ray and CT-scan datasets to train their proposed model. The images were preprocessed and augmented before they were fed to the proposed ResNet-50V2 model. They classified the images based on two-class, three-class, and four-class categories with and without preprocessing. The model achieved an accuracy of 99.01% and 83.6% for the 3-class category with and without preprocessing respectively.

(Baghdadi et al., 2022) introduced a Sparrow search algorithm (SSA) to optimize the pre-trained CNN models' hyperparameters via finding the best configuration. The pre-trained models used were SeresNext50, SeresNext101, SeNet154, MobileNet, MobileNetV2, MobileNetV3Small, and MobileNetV3Large. The authors used three publicly available datasets two of them with two classes and the other with three classes. They combined the datasets with the binary class and used it as one binary dataset. Both the combined and the three-class datasets were balanced using augmentation techniques. The best accuracy achieved was 99.74% for two classes and 98% for three classes with MobileNetV3Large and SeNet154 models respectively.

(S. Wang et al., 2021) proposed a 3D deep convolutional neural network (DeCoVNet) for automatic detection of COVID-19 from CT images. A total of 630 3D CT images were used from 540 patients (313 with COVID-19 and 229 without COVID-19). Firstly, they used a simple 2D UNet to segment the lung regions from the 3D volumes. The obtained segments were used as masks to train the (DeCoVNet) model. The training images were augmented to improve the model performance. The classification process was held many times by varying the probability thresholds and the best accuracy was 90.8% with a threshold of 0.3.

(Xu et al., 2020) proposed a classification model of three classes that was based on DL. These classes included 618 CT images of 219 COVID-19, 175 healthy individuals, and 224 cases of influenza-A viral pneumonia (IAVP). 3D CNN was used for segmentation. The segmented regions were then augmented for better performance. ResNet-18 transfer learning-based model was used for feature extraction. For the classification, they used a location-attention classification model. The proposed method obtained an accuracy rate of 86.7%.

Another DL study of COVID-19 classification was carried out by (Song et al., 2021). They collected 708 CT images of 86 healthy people, 505 CT images of 100 patients infected with bacterial pneumonia, and 777 CT images of 88 patients infected with COVID-19. They achieved a 93% accuracy rate for three-class classification by using a pre-trained DL model called Details Relation Extraction neural network. This model is based on the ResNet-50 model.

(S. Wang et al., 2021) managed to collect CT images from 259 patients, including 180 cases of typical viral pneumonia and 79 cases of COVID-19. To differentiate COVID-19 patients from patients with typical viral pneumonia, they made use of a transfer learning CNN-based model, which was the GoogleNet inception model. There were 320 images used to train the model: 160 negative COVID-19 images, and 160 positive COVID-19 images. Validation was carried out using 455 images (360 COVID-19 negative images and 95 COVID-19 positive images). The model achieved an 89.5 % accuracy.

1.2. Aims

This thesis aims to evaluate the effectiveness of the classification of COVID-19 from CT images by carrying out a comparative analysis of ML algorithms and DL-based transfer learning models. To achieve that, the experiment is carried out in two stages; machine learning stage and deep learning stage. In the machine learning stage, we investigate the impact of different feature extraction methods such as Shape-based, First-Order, and texture-based features on the performance of the classification process using different ML algorithms and COVID-CT dataset. In the deep learning stage, we used several transfer learning pre-trained models for the feature extraction and classification process with two different public COVID-19 CT datasets. Finally, a comparison between ML and DL methods for COVID-19 classification is carried out as the last goal.

In the literature, various ML algorithms such as SVM, logistic regression, k-NN, decision tree, RF, and NN as well as DL CNN-based pre-trained models such as VGG, ResNet, Inception, MobileNet, etc. have been applied to different private and public COVID-19 CT datasets. Additionally, several types of data preprocessing techniques such as resizing, normalization, filtering, segmentation, and feature extraction have been implemented. The applications made during the thesis study showed that the proposed diagnosis system gives successful results for COVID-19 classification. The proposed models perform well even with limited data preprocessing. Classification results using the proposed methods show a significant improvement in the performance compared to the state-of-art.

2. THEORETICAL BACKGROUND

2.1. COVID-19

Coronavirus Disease 2019 (COVID-19), caused by the new coronavirus (SARS-CoV-2), has quickly become a medical emergency and a worldwide crisis since its discovery in December 2019. The World Health Organization has declared it a pandemic on March 11, 2020. According to WHO statistics, by the 6th of April 2020, there have been more than 1.2 million recorded incidents of COVID-19 and 67,000 fatalities worldwide in 209 nations, regions, or territories(Lin et al., 2021). Individuals, families, healthcare institutions, food production, and economies are all feeling the pressure. COVID-19 has turned many people's lives upside down, and for some, it has resulted in a humanitarian catastrophe. Despite the imposition of strict travel bans, international travel by a significant number of people infected with COVID-19 has led to the virus spreading globally(Gwee et al., 2021). Universally, scientific societies, including diverse healthcare experts, are constantly attempting to study the virological, epidemiological, and clinical aspects of COVID-19 in order to discover viable treatments and vaccines(ElBagoury et al., 2021).

Coronaviruses, often known as CoVs, are a large viral family that is responsible for a wide variety of ailments, from the common cold to more severe diseases. Two members of the Coronaviruses family include the Middle East respiratory syndrome coronavirus (MERS-CoV) and the severe acute respiratory syndrome coronavirus (SARS-CoV).(H. Wang et al., 2020). SARS-CoV-2 is a novel coronavirus that contains positive-sense single-stranded RNA (+ssRNA). It has not been observed in humans, but it is extremely similar to the original SARS-CoV. (M. Y. Wang et al., 2020). Because of the virus's similarities to the bat coronavirus, it's believed to be animal origin. (Lin et al., 2021). The virus is infectious in humans, mammals, and birds. Spike (S), membrane (M), envelope (E), and nucleocapsid (N) proteins compose the majority of the viral structure. The viral envelope, which is a lipid bilayer produced from the host cell membrane, contains the S, M, and E proteins. On the other hand, the N protein interacts with the viral RNA in the virus core(Santos et al., 2020). Figure 2.1 shows the structure of SARS-CoV-2.

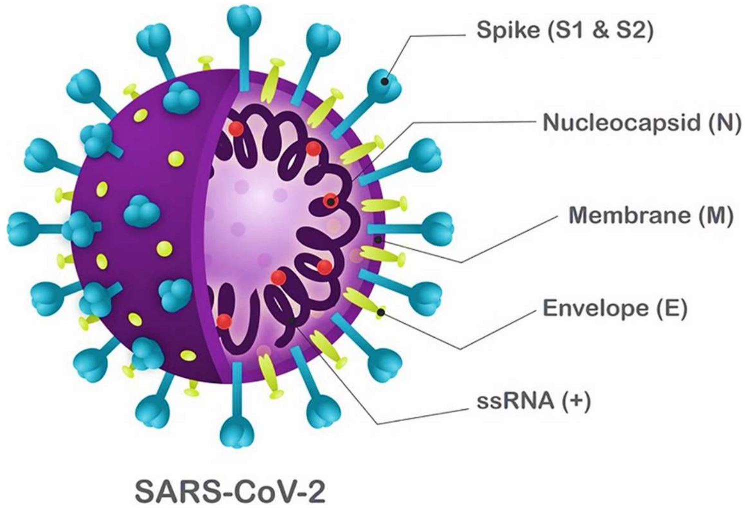


Figure 2.1. The structure of SARS-CoV-2 (Santos et al., 2020)

Chest CT is strongly suggested for both initial assessment and follow-up in suspected COVID-19 cases since the infection is involved in the pulmonary system. In early stages, chest images are of little clinical relevance, although CT abnormalities may be present even before symptoms appear. Furthermore, CT findings have been shown to be diagnostic in many cases following the RT-PCR assay(Carotti et al., 2020).

2.2. Computed Tomography (CT) Scan

CT scan is a modality that uses x-ray to capture the internal structure of the human body from different angles and generate slices. An associated computer called "image reconstruction unit" generates a three-dimensional image from the two-dimensional images obtained from the scanning process. The first CT scan was used to image the brain by its developer Dr. G.N. Hounsfield in 1971 in England. CT scanning was the first to obtain a high-resolution three-dimensional image(Bronzino, 2006). Figure 2.2 shows a typical CT scan modality and its component. A typical CT scan system consists of a gantry, a patient table, a control console, and a computer.

Configurations of data acquisition have changed through generations based on the scanning motion and detector arrangement. Each generation has its advantages and disadvantages. The first four generations of data acquisition configuration are illustrated in Figure 2.3 and we can see the differences in beam projection, scanning motion, and detector array arrangement.

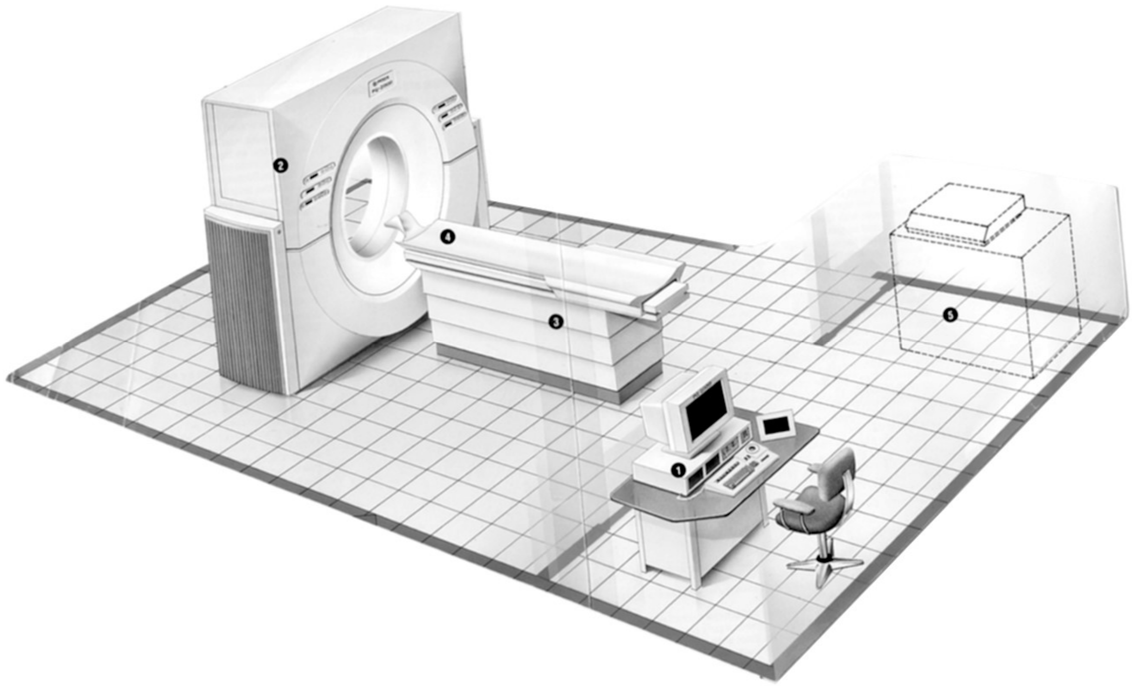


Figure 2.2. A typical CT scan system(Bronzino & Peterson, 2018)

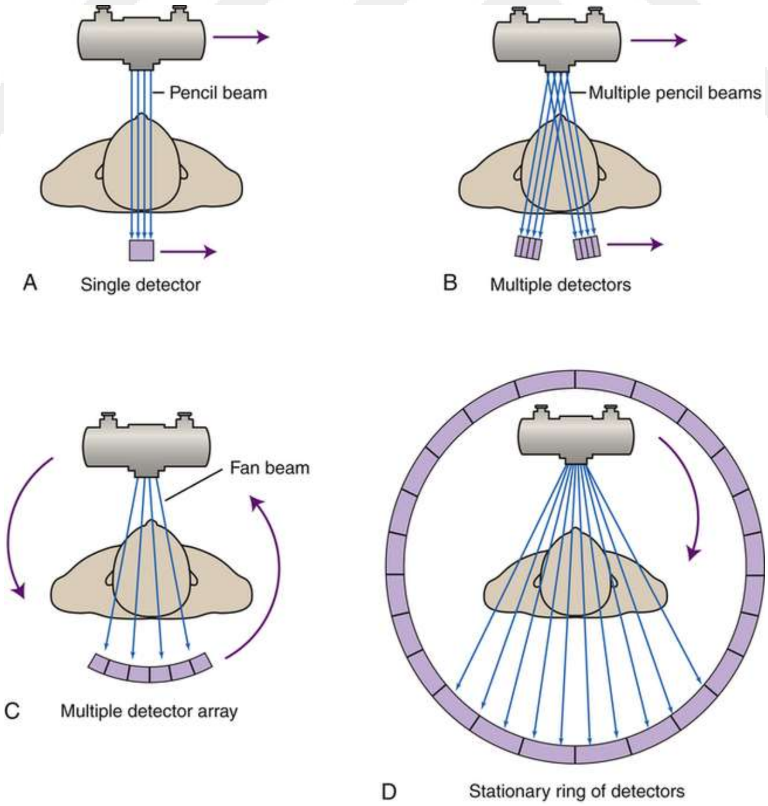


Figure 2.3. The four generations of data acquisition configuration in CT scan system A) first generation, B) second generation, C) third generation, and D) fourth generation (Radiology Key, n.d.)

Spiral/helical scanning methods were developed in response to the need for faster scan times, particularly for rapid sequential scans for three-dimensional imaging. Slip-ring technology that is self-lubricating is used to link electrical components to rotating components in third and fourth generation systems. In this approach, the x-ray fan beam can rotate constantly, eliminating the need for power and signal connections that would have to be unwound after each scan. Instead of stopping for each image, many images are taken as the patient moves through the gantry in a continuous motion as seen in Figure 2.4. Because the x-ray source follows a spiral or helical path around the patient, the reconstruction techniques are sophisticated(Bronzino, 2006).

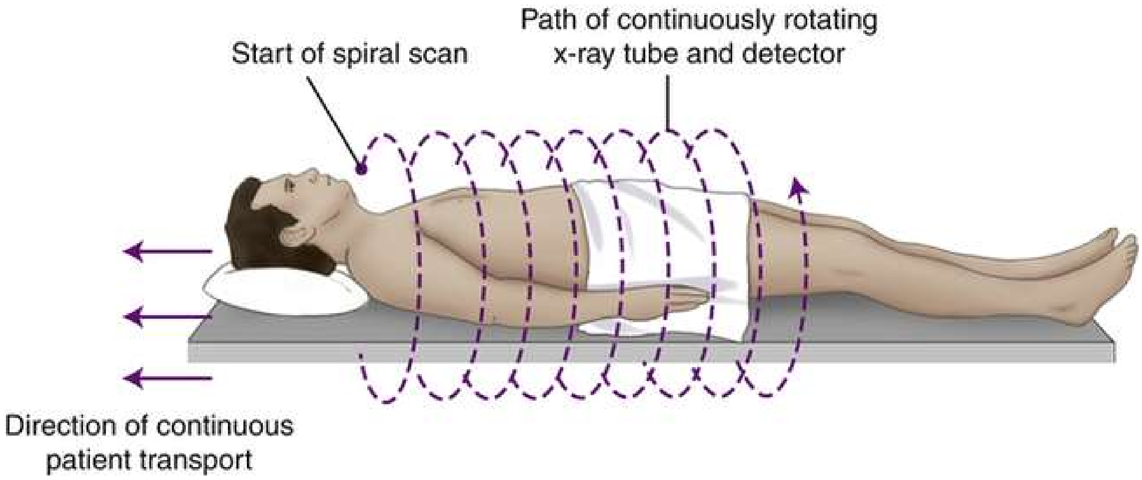


Figure 2.4. The spiral rotation of the x-ray tube around the patient as he is driven through the gantry aperture (Radiology Key, n.d.)

2.2.1. Gantry

Gantry is the largest part of the CT scan system and it houses the x-ray source as well as the x-ray detectors, the data-acquisition system, and the mechanical components responsible for the rotational motion. The gantry can also help inFigure 2.5 shows an uncovered gantry of a fourth generation CT scan with the internal components.

2.2.2. X-Ray Source

This part is responsible for emitting x-rays. It consists of a cathode and anode in a vacuum tube. The cathode produces an electron beam from its filament when heated through a passing current in (mA). The electron beam is accelerated by a high voltage (kV) applied between the cathode and the anode. When the electron beam hit the anode, it emits x-rays. The area of the anode at which the x-ray is generated is called the focal spot.

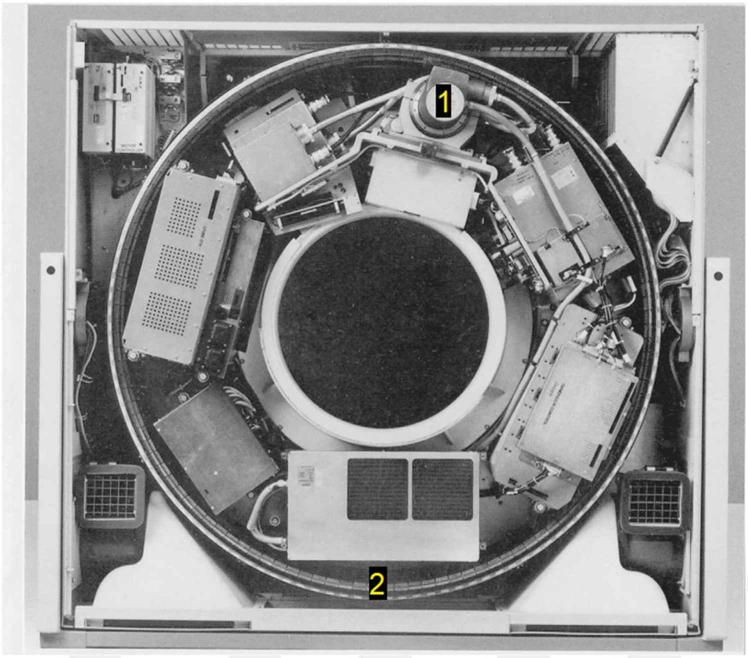


Figure 2.5. An uncovered gantry of a fourth generation CT scan showing the internal components 1) x-ray source, 2) x-ray detectors (Bronzino & Peterson, 2018)

It has a size of 0.5 × 1.5 mm and 1.0 × 2.5 mm in most systems. Because the electron beam hits the anode in a small focal spot, it produces heat that can melt the anode. For that reason, a rotating anode was developed to overcome that problem. The energy of the emitted x-ray beam depends on the material of the anode(General Electric Healthcare, 2014). Figure 2.6 depicts a view of a rotating-anode x-ray tube viewed through its cross-sectional axis.

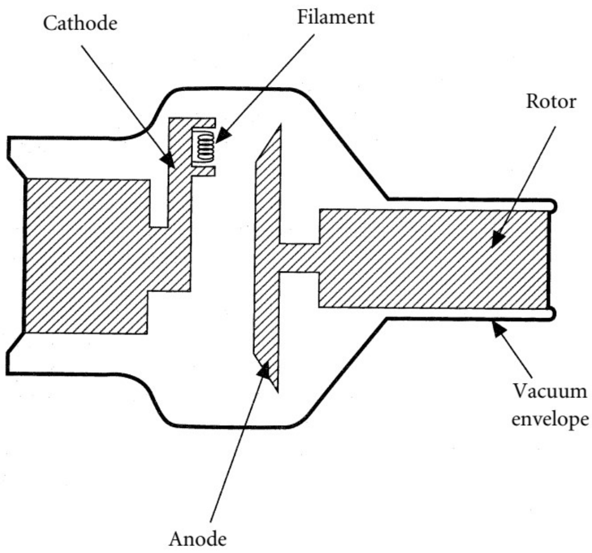


Figure 2.6. A basic rotating-anode x-ray tube (Bronzino, 2006)

2.2.3. X-Ray Detectors

The x-ray detectors in a CT scan system are an array of detector elements which could be solid-state detectors or gas ionization detectors. The solid-state detector is a combination of scintillating crystal and photodiode. The scintillating crystal in the detectors converts the x-ray beam into light, then a photodiode converts the light into an electrical signal. The gas ionization detector has a compressed gas inside them, usually Xenon, and a high voltage is applied to metal plates to collect the ions produced from the ionized gas by x-ray. The detector array could be rotary as in the third generation or stationery as in the fourth generation. The rotary array of detectors is arranged as an arc faces the x-ray tube and rotates along with it in the gantry. The stationary detector array, on the other hand, is an array that is circularly arranged along the gantry(Bronzino, 2006). Figure 2.7 demonstrate the two types of detectors used in most CT scan systems.

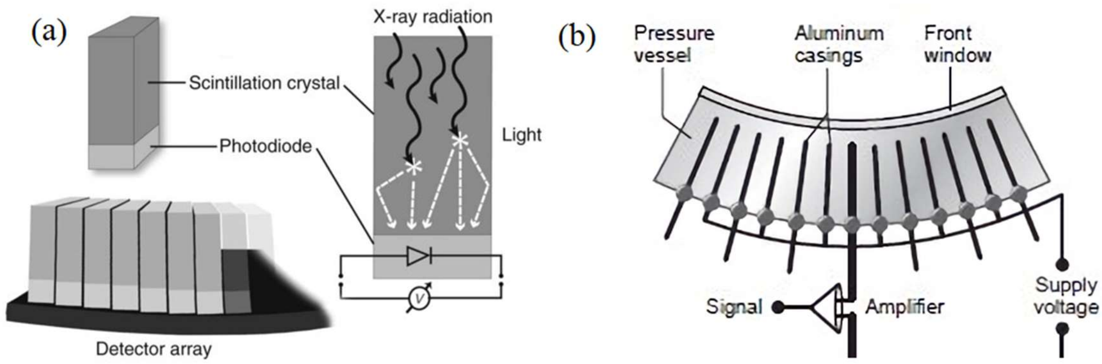


Figure 2.7. The detector types used in CT scan systems a) solid-state detector, b) gas ionization detector (Rad Tech On Duty, n.d.; Veterian Key, n.d.)

2.2.4. Data-Acquisition System

It's the system that receives, filters, and amplifies the electrical signal produced by the detectors then converts it into a digital signal and sends it to the computer system.

2.2.5. Patient Table

The patient table provides support and vertical/longitudinal motion of the patient relative to the CT scanner.

2.2.6. Control Console

It's a console used by a CT operator to control the CT system. The operator controls the dose of the x-ray and reconstructs and displays the images using a graphical user interface.

2.2.7. Computer System

This system is a combination of hardware and software that links the CT scan system components. The computer system is used to acquire the data from the data acquisition system, reconstruct, display, and store image data.

2.3. Artificial Intelligence

Systems or machines that are able to replicate human intelligence in order to carry out tasks and improve themselves based on the information that they acquire are referred to as having artificial intelligence (AI). In AI, the ability to understand and interpret data is more important than any particular form or function that it might take. The purpose of AI is to greatly improve human capabilities and contributions. (Oracle, n.d.). ML is a subcategory of AI as Figure 2.8 shows the relationship between them.

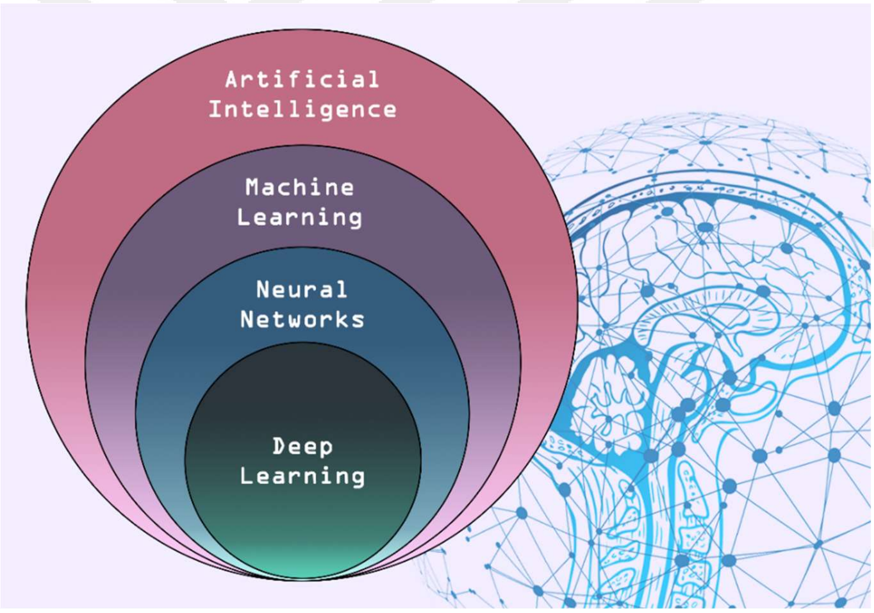


Figure 2.8. The relationship between AI and ML

2.3.1. Machine Learning

ML is a subfield of AI that builds models from algorithms to solve a practical problem based on a problem-related dataset (Expert, n.d.). Machine learning models can learn from data by identifying patterns and then automatically make decisions based on what they have learned without being explicitly programmed (Burkov, 2019). ML is classified into four types: supervised learning, semi-supervised learning, unsupervised learning, and reinforcement learning.

2.3.1.1 Supervised Learning

Supervised learning is a learning method that builds a model using a labeled input dataset and tests this model over the test set. The dataset contains many examples/samples and we extract features from each sample. Each sample has features that describe it as well as a label to which the sample belongs. All the samples in the dataset have the same number and features in the feature vector but are different in value. The label could be discrete value {0,1,2, 3, …., C} or a continuous value. Accordingly, supervised learning methods were divided into two types Regression and Classification (Burkov, 2019). Figure 2.9 shows the workflow of supervised machine learning.

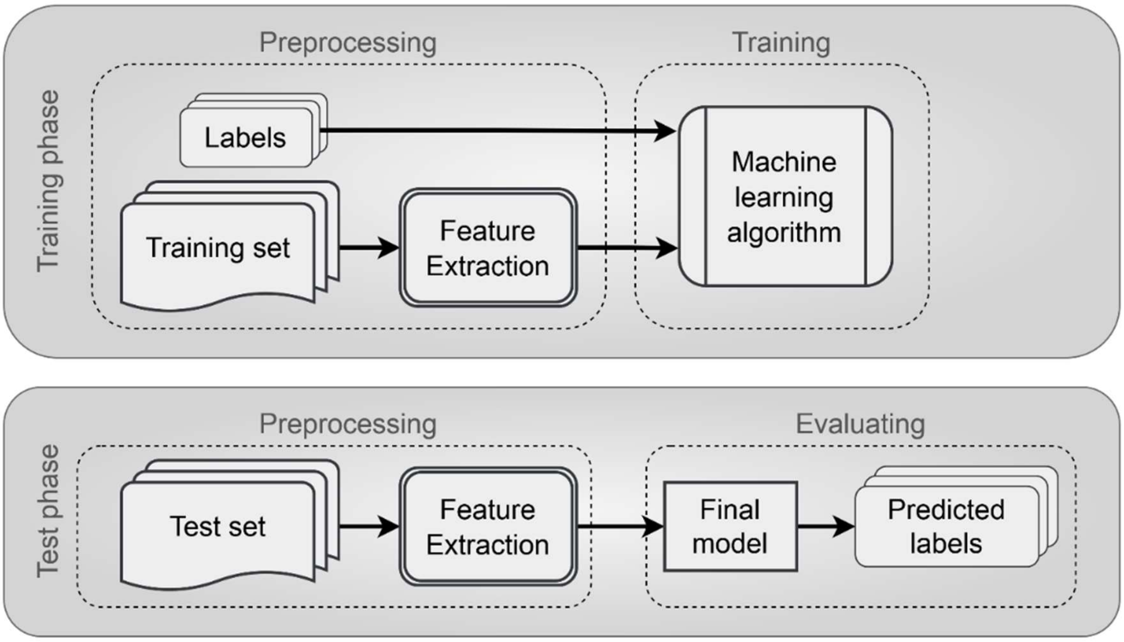


Figure 2.9. A simple workflow of supervised machine learning in which the model is trained in the training phase and evaluated in the test phase

Regression is the problem in which the model predicts a continuous value label for a given unlabeled sample. The model will predict based on the labeled dataset it learned from. An example of regression is predicting a house price using house features such as house area, number of rooms, and house age.

Classification works on predicting which class a new test sample will belong to, by using a labeled training set in which each sample is known to which class it belongs. An example of a classification problem is predicting if it’s going to rain or not based on features like temperature, presence of clouds, and wind speed.

The most widely used supervised ML algorithms are support vector machines (SVM), k-nearest neighborhood (k-NN), ensemble learning, and neural network algorithms.

2.3.1.1.1. Data Preprocessing

Data Preprocessing is a preparation step that makes the raw data more suitable for algorithms to increase their learning pace and enhance their performance (Ketkar & Moolayil, 2021). Resizing, normalization, standardization, image filtering, segmentation, augmentation, and feature extraction are all included in data preprocessing.

Resizing is used with image datasets and means reducing the size of the image. Resizing is common in DL because large images require large computational power and more time.

Normalization is the process of changing a numerical feature's real range of values into a standard range of values, mostly in the interval [-1, 1] or [0, 1]. Normalization helps the algorithms learn faster (Burkov, 2019). Normalization is used in machine and deep learning. Normalization Formula is in Equation (2.1),

$$x_{normalized} = \frac{x - min(x)}{max(x) - min(x)} \tag{2.1}$$

Standardization, unlike normalization, does not limit values to a certain range; rather, it rescales the feature to a standard normal distribution with a mean value of 0 and a standard deviation of 1. Standardization is used in machine learning to prevent the algorithm from biasing towards features with high values and thus affecting the learning process (Burkov, 2019). The Formula for standardization in Equation (2.2),

$$x_{standardized} = \frac{x - mean\ (x)}{standard\ deviation(x)} \tag{2.2}$$

Image filtering is the process of altering or improving an image to emphasize particular features and removing others (Mathworks, n.d.). Smoothing, sharpening, and edge enhancement are examples of filtering and are illustrated in Figure 2.10. The filtering process is accomplished by applying a convolutional filter/kernel (usually 3x3 or 5x5) to the input image and sliding the filter over the image.

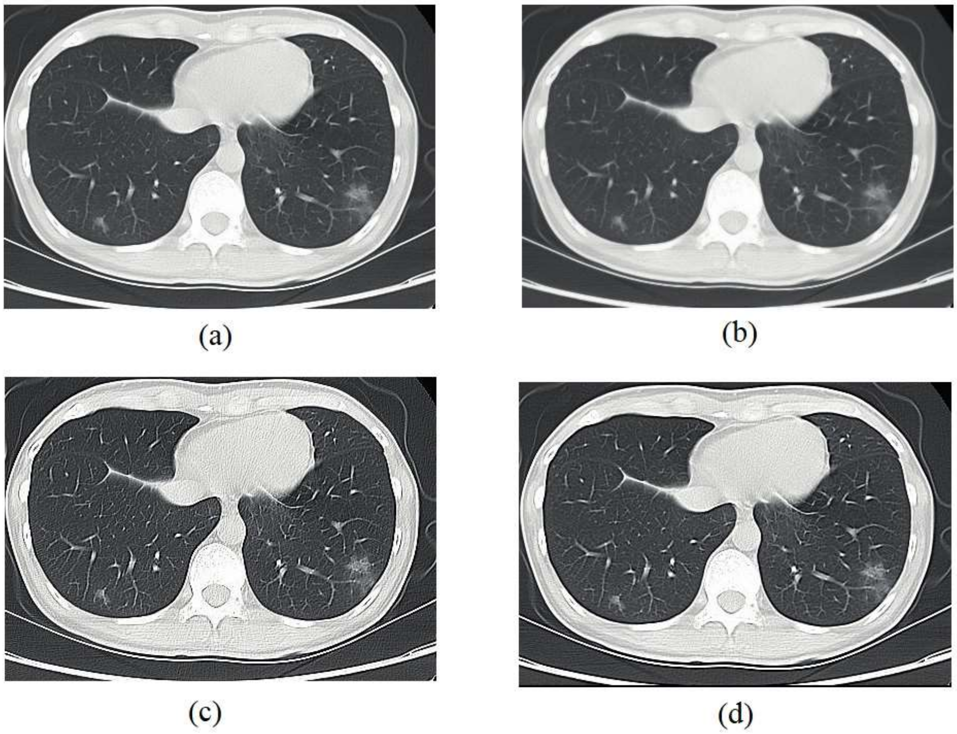


Figure 2.10. Examples of image filtering on CT images a) original image, b) smoothed image, c) sharpened image, and d) edge enhanced image

Segmentation means separating image contents into parts or segments according to their pixel values, shape, or position. Region of interest (ROI) segmentation is a type of segmentation where a specific region of the image is selected and the rest are removed as in Figure 2.11.

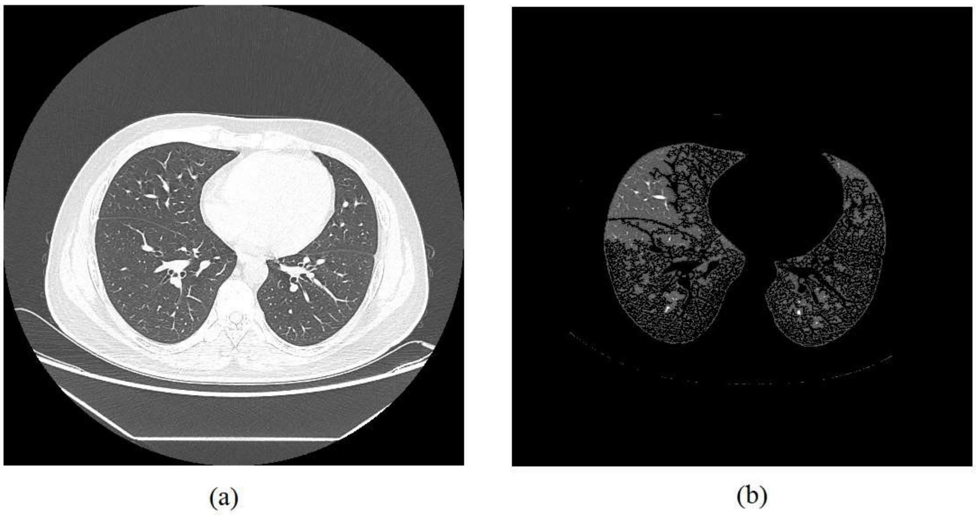


Figure 2.11. An example of ROI segmentation

Augmentation artificially expands the dataset by generating a large number of realistic versions of each sample(Géron, 2019). Augmentation is used with small datasets when the algorithms perform poorly. It could also be used with unbalanced datasets that have a small number of samples in a specific class(Ahamed et al., 2021). Cropping, zooming in or out, shearing, rotating, flipping, and changing the brightness are some techniques of augmentation. Figure 2.12 shows some image augmentation techniques.

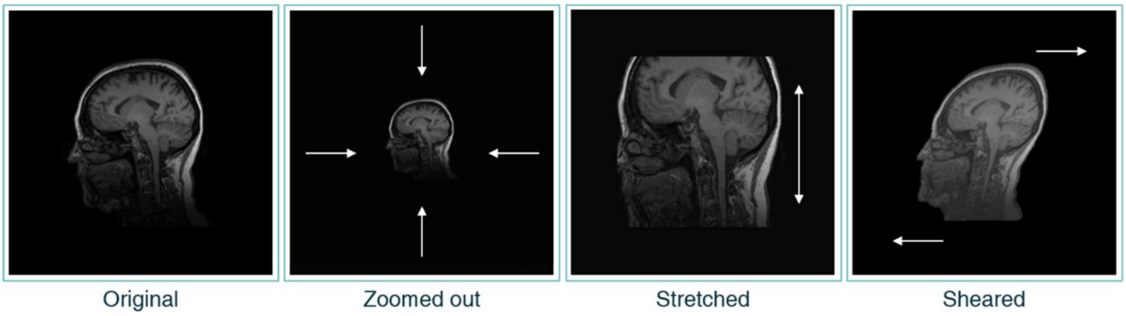


Figure 2.12. Some examples of augmentation techniques

Feature extraction is the process of removing unnecessary information by extracting only the features that describe data the most based on your knowledge of the data. In many situations, expecting a machine learning model to learn from fully arbitrary data is impractical. The data must be provided to the model in such a way that the model's task is facilitated (Goodfellow et al., 2016).

2.3.1.1.2. Support Vector Machine

SVM is one of the most widely used supervised learning algorithms, and it is most commonly applied to problems involving binary classification or regression. It was developed by Vladimir Vapnik and Corinna Cortes in 1995(Cortes & Vapnik, 1995). Based on the number of features, SVM examines the data in an n-dimensional space, where the value of each feature is the value of a certain coordinate. By drawing a line between them, the algorithm divides the sample into two classes. This line, which divides the good examples from the bad ones, is called the decision boundary. It is just a line in 2D, a plane in 3D, and a hyperplane in higher dimensions when viewed from such perspectives. Let $\{x_i, y_i\}$, $i=1, 2, ..., N$ be a dataset with N samples, where $x_i \in R^n$ feature vector for sample I with n features, $y_i \in \{-1,1\}$ is the label for sample i. The linear hyperplane that will separate the data is given by Equation (2.3),

$$w^T x_i + b = 0 \tag{2.3}$$

where w^T is the weights vector, x_i is the feature vector, and b is a bias. To find whether a class is positive or negative, SVM takes the input vector, multiply it with $w*$, add it to $b*$, then the sign of the outputs is the predicted class. So, $w^T x_i + b \geq +1$ if $y_i = +1$, and $w^T x_i + b \leq -1$ if $y_i = -1$. A uniform equation that donates a sample x_i for both classes can be written as $y_i(w^T x_i + b) \geq +1$. SVM finds $w*$ and $b*$ which are the optimal values for w and b parameters by using the optimization function. Figure 2.13 visualize the hyperplane along with the support vectors.

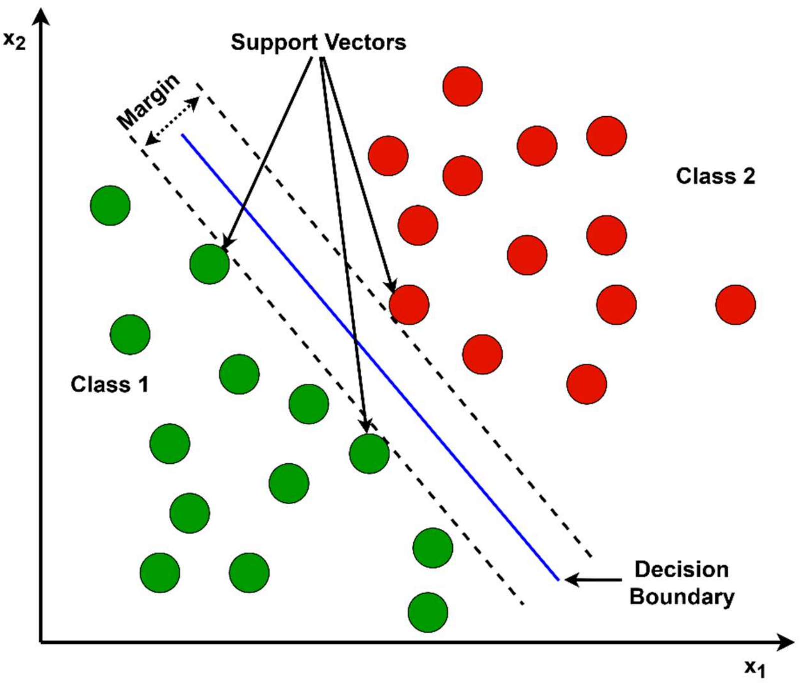


Figure 2.13. A two-class (binary) dataset as red and green points with two features x_1 and x_2 is drawn in 2-dimensional space and separated by a linear decision boundary

The distance between the decision boundary and the closest examples is called the margin m and can be found by the formula $m = \frac{2}{||w||}$. Minimizing $||w||$ will maximize the margin and therefore gives better generalization, making the model classify the new examples well. Support vectors shown in Figure 2.13 are used by SVM to minimize $||w||$. The term $\frac{1}{2}||w||^2$ is used for the quadratic programming optimization and minimizing it is equivalent to minimizing $||w||$. The quadratic programming problem is given by Equation (2.4),

$$min\frac{1}{2}||w||^2 \;\; such\ that \;\; y_i(w^T x_i + b) \geq +1, \forall i \tag{2.4}$$

To solve the quadratic programming problem the Lagrange Multiplier method can be used as in Equation (2.5),

$$\mathcal{L}(w,b,\alpha) = min\frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1) \tag{2.5}$$

where α is the Lagrange Multiplier and $\alpha_i > 0$. The partial derivation of the Lagrange Multiplier function (2.5) gives us the KKT conditions in Equations (2.6) and (2.7),

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{2.6}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow 0 = \sum_{i=1}^{n} \alpha_i y_i \tag{2.7}$$

plugging into the Lagrange Multiplier function will reduce it to a dual optimization function as in (2.8),

$$\mathcal{L}_d(w,b,\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j\,(x_i x_j) \tag{2.8}$$

After solving the dual optimization function, we can get a vector α that contains the Lagrange Multiplier values for every sample. Then, we can find *w* and *b* from Equations (2.9) and (2.10),

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{2.9}$$

$$b = \frac{1}{S}\sum_{i=1}^{S} (y_i - w^T x_i) \tag{2.10}$$

where *S* refers to the number of support vectors.

SVM can separate the data by linear decision boundary as we saw in Figure 2.13 or non-linear decision boundary using kernels. Kernels are functions that transform the data such that the hyperplane can separate them. The kernel performs inner product to its input(Flach, 2012). The most popular kernels used in non-linear SVM are polynomial, radial basis function (RBF), and sigmoid kernels, and their Formulas are given in Equations (2.11) (2.12) (2.13) respectively.

$$K(x_i, x_j) = (x_i \cdot x_j + C)^d \tag{2.11}$$

$$K(x_i, x_j) = e^{-\gamma \left|\left|x_i - x_j\right|\right|^2 + C} \tag{2.12}$$

$$K(x_i, x_j) = \tanh\left(\gamma\left(x_i x_j\right) + C\right) \tag{2.13}$$

where *d* is the polynomial's degree parameter, *C* is the regularization parameter, and γ is the gamma parameter $\gamma > 0$.

2.3.1.1.3. k-Nearest Neighbors

K-NN is a simple algorithm that can be used to solve classification and regression problems in machine learning. This algorithm puts the examples in a feature space and once an unseen example x comes it finds the k closest neighbor examples and assigns the average label to the new example. For that it doesn't involve any training or learning process(Burkov, 2019). When k is equal to 1, the k-NN algorithm is in its most basic version. This means that the test example will be classified based on the information provided by its first nearest neighbor in the data used for training. The k-NN algorithm's basic principle is illustrated in Figure 2.14.

For two points *x* and *y* in a plane the distance between them can be calculated using one of the distance Equations (2.14), (2.15), and (2.16);

$$Euclidean\ Distance\ d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.14}$$

$$\text{Manhattan Distance } d(x,y) = \sum_{i=1}^{n} |x - y_i| \tag{2.15}$$

$$\text{Minkowski Distance } d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{2.16}$$

where p is the power parameter and it defines the distance, when $p=1$ it returns the Manhattan distance and when $p=2$ it returns the Euclidean distance.

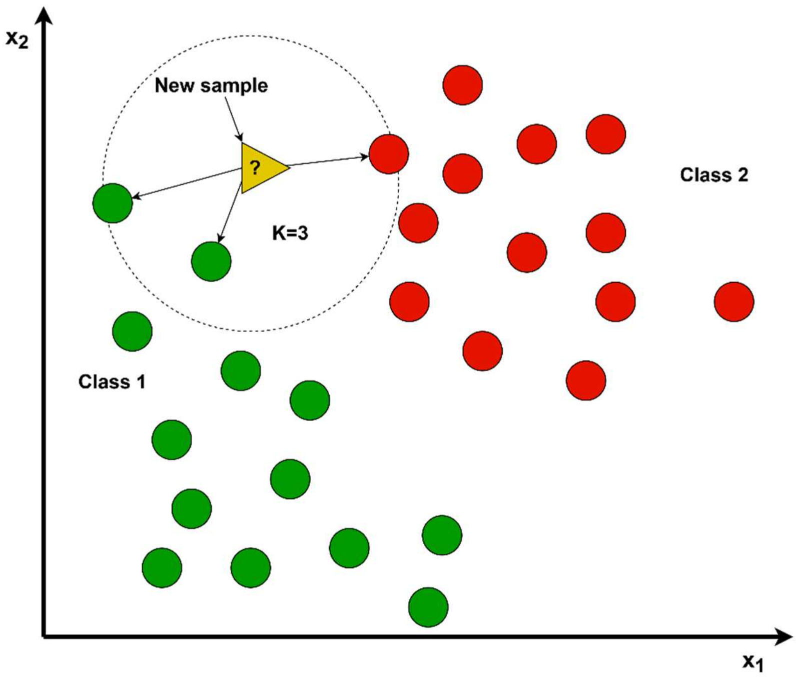


Figure 2.14. A two-class (binary) dataset with two features x_1 and x_2 is drawn in 2-dimensional space as red and green points (features) and a new sample as a yellow triangle is placed inside the space. The new sample takes the label of the average of its three closest neighbors which is class 1 in this case

2.3.1.1.4. Ensemble Learning

Ensemble learning focuses on training many weak learners which are low-accuracy models like decision trees and takes the average of their predictions. There are two modes of algorithms in ensemble learning; bagging and boosting(Burkov, 2019).

Random forest is one of the most widely used supervised ensemble-based ML algorithms. RF is a bagging algorithm that consists of many decision trees, which are fast at training and prediction. The training set is split into different subsets where each subset is partially different(Géron, 2019). Then, we train each decision tree on each subset to obtain multiple weak models and take the average of all their predictions as the answer. Figure 2.15 illustrates how RF work.

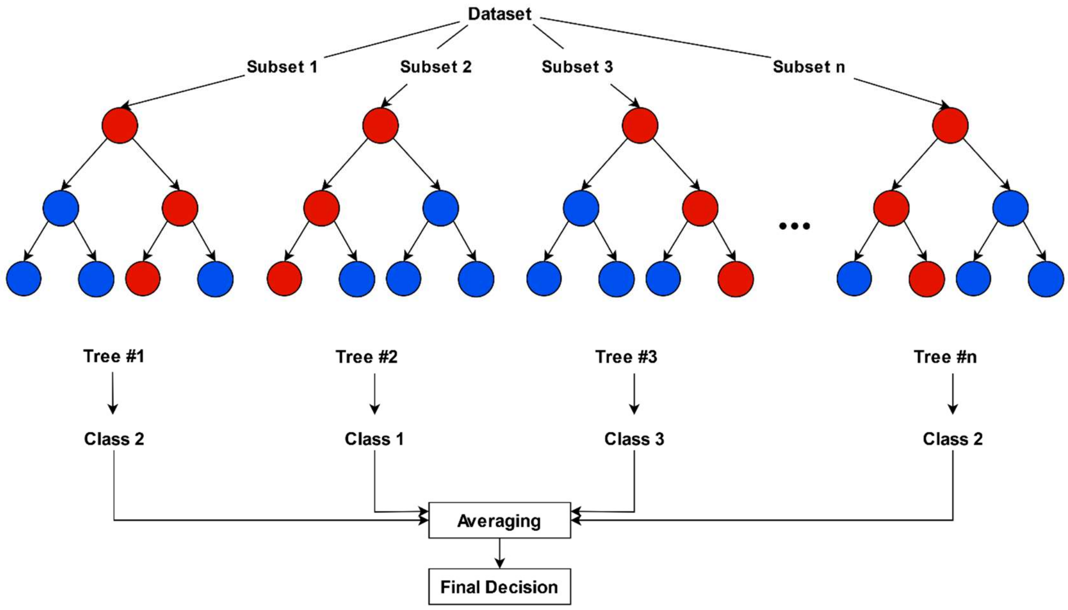


Figure 2.15. Random forest working diagram

XGBoost (eXtreme Gradient Boosting) is a modified, Gradient Boosting-based ensemble learning algorithm that has high performance over traditional Gradient Boosting algorithms. The concept behind gradient boosting is to construct multiple decision tree-based weak learners in a sequential manner, with each model attempting to learn from the error of the previous one and thus building an improved and more accurate final version of the model (Burkov, 2019) as can be seen in Figure 2.16.

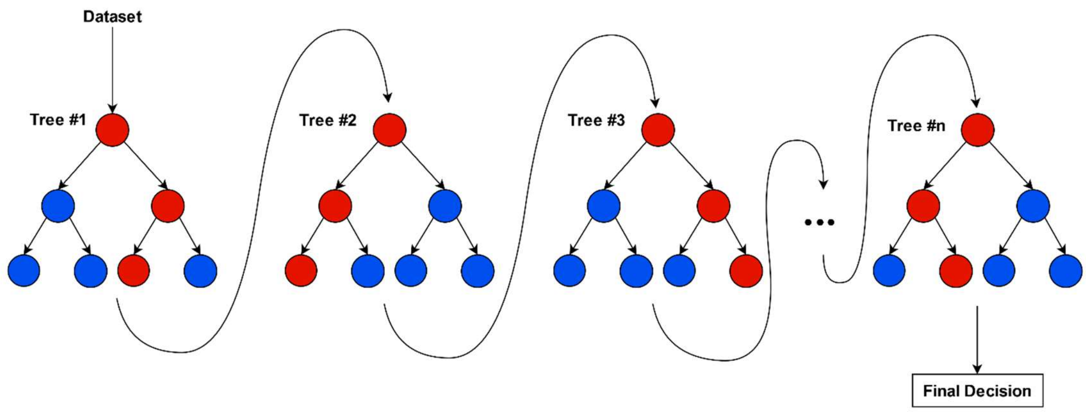


Figure 2.16. A general structure of the Gradient Boosting algorithm

This algorithm's most key advantages are its high predictive power, ability to prevent over-learning, management of empty data, and speed.

2.3.1.1.5. Neural Networks

NN, also called artificial neural networks (ANN),are supervised ML algorithms that is used for classification or regression problems and are the core of DL algorithms(IBM Cloud Education, 2020). Neurons make up each layer of the neural network, and each neuron in one layer is connected to every other neuron in the layer before it and the layer after it. They are connected through a connection called weight/parameter. Each weight has a value that defines how much impact the neuron has on the next neuron. Each neuron takes the sum of weights multiplied by the outputs of the previous neurons, applies an activation function, and sends it as an input to the neurons in the next layer as shown in Figure 2.17. Based on the input and the weights, the activation function decides whether to activate the neuron or not.

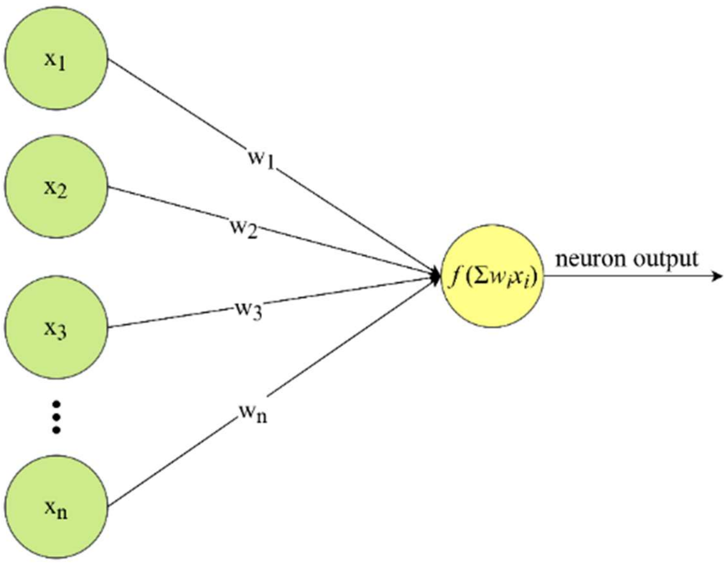


Figure 2.17. A neuron structure in the NN. x_i is the input features vector, w_i is the weights vector, and f is the activation function.

The first layer in NN is the input layer which has as many neurons as the number of features in the feature vector. The last layer in this network is the output layer which has as many neurons as the number of classes in the input data except for a two-class dataset where the output has only one neuron that produces 1 or 0. Consider the simple neural network in Figure 2.18 as an example. This neural network has three layers; input layer, hidden layer, and output layer. The neural network in Figure 2.18 is used with a dataset that has two classes and each sample has three features. Thus, the input to this network is a three-dimensional feature vector, and the output is a scalar (0 or 1).

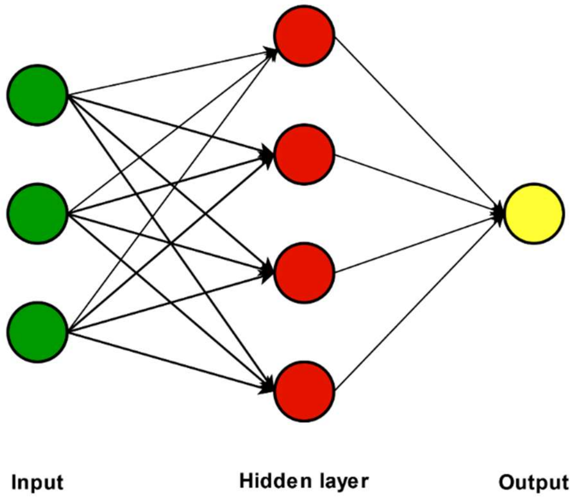


Figure 2.18. A simple MLP with three layers

2.3.1.2 Semi-Supervised Learning

The process of enabling machines to classify samples based on both labeled and unlabeled data is referred to as semi-supervised learning. The number of unlabeled samples is usually substantially more than the number of labeled ones.

2.3.1.3 Unsupervised Learning

Unsupervised learning uses an unlabeled dataset and creates a model that describes the samples by either grouping them into clusters based on their features as in Clustering problems, reducing the number of features as in Dimensionality Reduction problems, or determining the difference between a sample from a typical sample in the dataset as in Outlier Detection.

2.3.1.4 Reinforcement Learning

Reinforcement learning is a branch of ML in which an agent is put in an interactive environment and learn to make decision and actions and get rewards or penalties according to them. The reinforcement learning agent is an algorithm designed to learn a policy which is a function that takes a feature vector as input and returns the best action to do to maximize the total reward.

2.3.2. Radiomics

Radiomics is a growing field in radiology that extracts several quantitative properties from a digital radiological image(Mayerhoefer et al., 2020). Radiomics is a combination of the two terms 'radio' meaning medical images and 'omics' representing fields such as genomics and proteomics that help us understand various medical conditions(Koçak et al., 2019). Radiomics can be applied to various imaging techniques such as CT, MRI, PET, X-ray, and ultrasonography(Peper & Dahl, 2016; Rizzo et al., 2018). Radiomics features are mainly divided into 'semantic' and 'agnostic'. Semantic features are the quantitative counterparts of morphological features such as size and shape used by radiologists to describe lesions. Agnostic features, on the other hand, are quantitative texture parameters that analyze the distribution of gray levels in the image space and their relationship with each other, which the human eye cannot distinguish (Gillies et al., 2016). Pyradiomics library for python contains shape-based features, first-order features, and texture-based features. The texture-based method uses matrices to extract texture features and then calculate statistical features from those matrices(Saleh et al., 2021).

2.3.2.1 Shape-Based Features

Shape features use descriptors that describe the size and the shape of the image and its geometric properties. There are 10 Shape-based features extracted using the Pyradiomics library.

2.3.2.2 First-Order Statistics Features

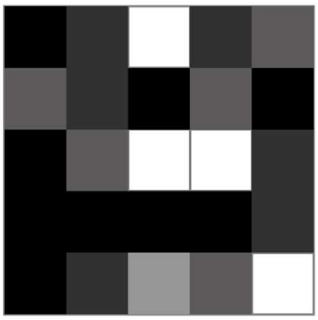
FOS features are histogram-based features that measure the gray level distribution in the image. It does not contain information about the spatial relationship of pixels with surrounding pixels. There are 14 features that can be extracted by this method using the Pyradiomics library.

2.3.2.3 Texture-Based Features

Texture-based features examine pixels' relationship to surrounding pixels, that is, they describe statistical relationships between pixels. There are five methods use matrices in Pyradiomics to extract texture-based features and they are GLCM, GLRLM, GLSZM, NGTDM, and GLDM.

2.3.2.3.1. Gray level co-occurrence matrix (GLCM)

The GLCM method analyzes the image to determine the frequency of pixel pairs. Pairs separated by a predefined distance δ are counted for different angles θ such as 0°, 45°, 90°, and 135° separately then the mean value of them is returned. Each value in the matrix is expressed as $(i,j)^{th}$ which is the number of i and j pixel pairs that are in the image. The distance should be determined in accordance with the application, as a greater distance will enable to detect rough areas. Figure 2.19 illustrate how GLCM is extracted from the image.



a

1	2	5	2	3
3	2	1	3	1
1	3	5	5	2
1	1	1	1	2
1	2	4	3	5

b

	1	2	3	4	5
1	6	4	3	0	0
2	4	0	2	1	3
3	3	2	0	1	2
4	0	1	1	0	0
5	0	3	2	0	2

c

Figure 2.19. Working principle of GLCM a) A 5x5 gray image with 5 gray level values, b) the equivalent numerical matrix of the left image, c) the GLCM matrix of the image

The GLCM in Figure 2.19 was created by calculating the pixel pairs horizontally i.e., 0° and 1 distance. Rows represent the first pixel in the pairs and columns represent the second pixel. For 1 and 3 pair of pixels, they occurred three times the in the image as shown in c. Thus, the number 3 is written in the third row of the first column as well as in the first row of the third column because pairs are considered for both forward and backward directions. This GLCM is diagonally symmetric and is called “the symmetrical horizontal GLCM”. The following Equation (2.17) is used to compute the GLCM;

$$P_{\delta,\theta}(i,j) = \sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\begin{cases}1, & if\ I(x,y) = i\ \ and\ \ I(\Delta x,\Delta y) = j \\ 0, & otherwise\end{cases} \tag{2.17}$$

where: $P_{\delta,\theta}(i,j)$ is the GLC matrix of an $N_x \times N_y$ image, $I(x,y)$ the value of the pixel at the position (x,y), $(\Delta x\,,\Delta y)$ is the offset of the pixel pair for δ distance on angle θ given by Equation (2.18);

$$I(\Delta x\,,\Delta y) = \begin{cases} I(x+\delta\,,y) & if\ \theta = 0° \\ I(x+\delta\,,y+\delta) & if\ \theta = 45° \\ I(x\,,y+\delta) & if\ \theta = 90° \\ I(x-\delta\,,y+\delta) & if\ \theta = 135° \end{cases} \tag{2.18}$$

From the GLCM, 24 features can be extracted using the Pyradiomics library.

2.3.2.3.2. Gray level run length matrix (GLRLM)

GLRLM describes the distribution of consecutive pixels (run) with the same gray level values in the image in a specific direction. Runs of the same pixels are counted for different angles such as 0, 45, 90, and 135 separately then the mean value of them is returned. Each value in the matrix is expressed as *(i,j)*th which is the number of pixels with the value i and its run length j that are in the image. Figure 2.20 illustrate how GLRLM is extracted from the image.

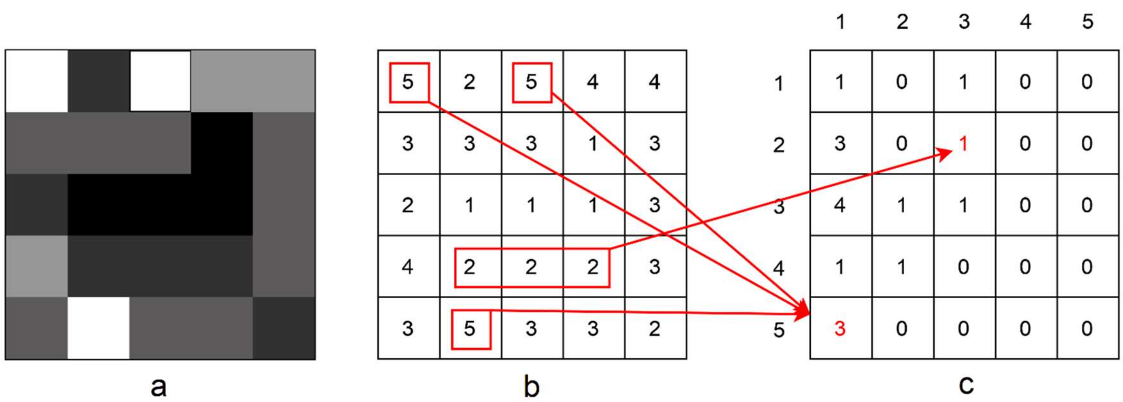


Figure 2.20. Working principle of GLRLM a) A 5x5 gray image with 5 gray level values, b) the equivalent numerical matrix of the left image, c) the GLRLM matrix of the image

The GLRLM in Figure 2.20 was created by calculating the size zones of pixels in the image. Rows represent the pixel gray values and columns represent their run length. For pixels with value 5, three of them occurred separately in the image which means they have a run length of 1 as shown in Figure 2.20. Thus, number 3 was written in the fifth row of the first column. The following Equation (2.19) is used to compute the GLRLM;

$$P_\theta(i,j) = |\{(x, y): |\{(k,l) \in Nb(x, y, j, \theta): I(k, l) = i\}| = j\}| \tag{2.19}$$

where: $P_\theta(i,j)$ is the GLRL matrix of an $N_x \times N_y$image, $I(x,y)$ the value of the pixel at the coordinate (x,y), $|\cdot|$ denotes the cardinality of a set, $Nb(x,\ y,\ j,\ \theta)$ is the set of the pixel for j run length on angle θ given in Equation (2.20);

$$Nb(x, y, j, \theta) = \begin{cases} I(x+1, y) + I(x+2, y) + \cdots + I(x+j, y) & if\ \theta = 0° \\ I(x+1, y+1) + I(x+2, y+2) + \cdots + I(x+j, y+j) & if\ \theta = 45° \\ I(x, y+1) + I(x, y+2) + \cdots + I(x, y+j) & if\ \theta = 90° \\ I(x-1, y+1) + I(x-2, y+2) + \cdots + I(x-j, y+j) & if\ \theta = 135° \end{cases} \tag{2.20}$$

From the GLRLM, 16 features can be extracted using the Pyradiomics library.

2.3.2.3.3. Gray level size zone matrix (GLSZM)

GLSZM describes the distribution of adjacent pixels (zone) with the same gray level values in the image in a specific direction. Zones of connected pixels with the same gray level values are counted and allocated to the GLSZM. Each value in the matrix is expressed as *(i,j)*th which is the number of connected pixels with the value *i* and its zone size *j* that are in the image. Figure 2.21 illustrate how GLSZM is extracted from the image.

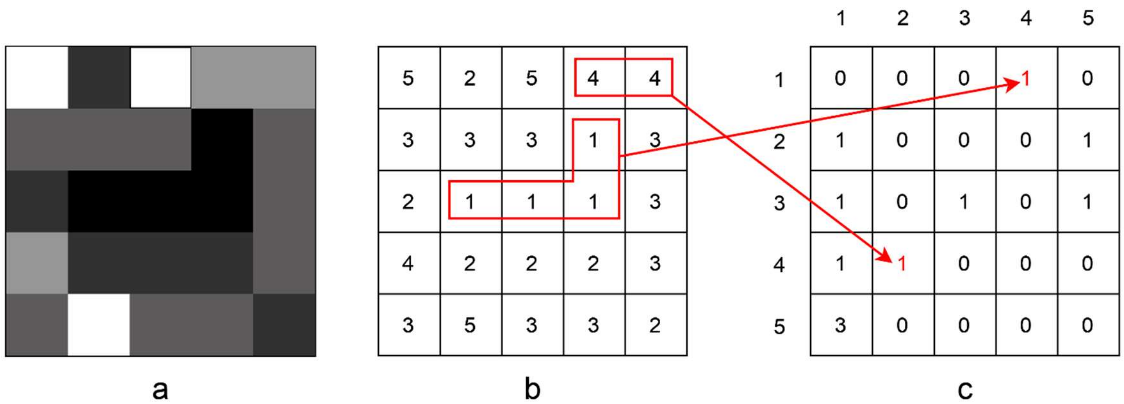
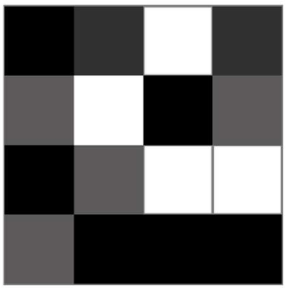


Figure 2.21. Working principle of GLSZM a) A 5x5 gray image with 5 gray level values, b) the equivalent numerical matrix of the left image, c) the GLSZM matrix of the image

The GLSZM in Figure 2.21 was created by calculating the run length horizontally i.e., 0°. Rows represent the pixel gray values and columns represent their zone size. For pixels with value 4, two of them are connected which means they have a zone size of 2 as shown in Figure 2.21 b). Thus, number 1 was written in the fourth row of the second column. From the GLSZM, 16 features can be extracted using the Pyradiomics library.

2.3.2.3.4. Neighboring gray tone difference matrix (NGTDM)

NGTDM measures the difference between the gray level value of a pixel and the average gray level value of the neighboring pixels at a predefined distance. NGTDM is a one-dimensional matrix in which each value S_i is the absolute sum of the differences between a gray level value i and the average value of its neighbors. In Figure 2.22, the working principle of NGTDM is shown.



a

1	2	5	2
3	5	1	3
1	3	5	5
3	1	1	1

b

i	n_i	p_i	s_i
1	6	0.375	13.35
2	2	0.125	2.00
3	4	0.25	2.63
4	0	0.00	0.00
5	4	0.25	10.075

c

Figure 2.22. Working principle of NGTDM a) A 4x4 gray image with 5 gray level values, b) the equivalent numerical matrix of the left image, c) the NGTDM matrix of the image

The NGTDM has only four columns and the number of rows is dependent on the number of gray values in the image. The NGTDM in Figure 2.22 was created by calculating n_i, P_i , and S_i for a neighbor distance of 1. For pixel with value 2, there are two of them in the image, and the corresponding sum of absolute differences S_i was calculated as $S_2 = \left|2-\frac{15}{5}\right| + \left|2-\frac{9}{3}\right| = 2$. Because there isn't any pixel with gray level 4, its corresponding n_i, P_i , and S_i are 0's as shown in Figure 2.22 c). The following equations are used to compute the NGTDM;

Let $I(x, y)$ be the gray tone of any pixel at (x, y) having gray tone value i. Then we find the average gray tone over a neighborhood centered at, but excluding (x, y) with Equation (2.21);

$$\bar{A_i} = \bar{A}(x,y) = \frac{1}{W-1}\left[\sum_{m=-d}^{d}\sum_{n=-d}^{d} I(x+m,\ y+n)\right] (m,n) \neq (0,0) \tag{2.21}$$

where d specifies the neighborhood size and $W = (2d+1)^2$.

Then the ith entry in the NGTDM is $S_i = \begin{cases} \sum|i - \bar{A_i}|, & for\ i \in N_i\ if\ N_i \neq 0 \\ 0, & otherwise \end{cases}$

where N_i is the set of all pixels having gray tone i (except in the peripheral regions of width d). From the NGTD matrix, 5 features can be extracted using the Pyradiomics library.

2.3.2.3.1. Gray level dependence matrix (GLDM)

GLDM describes the distribution of connected pixels (dependency) with a center pixel within a predefined distance. Each value in the matrix is expressed as *(i,j)th* which is the number of times a center pixel with the value *i* that has *j* dependencies occurs in the image. A pixel with gray level *j* is count to be dependent on a center pixel with gray level *i* if $|i-j| \leq \alpha$. Figure 2.23 illustrate how GLDM is extracted from the image.

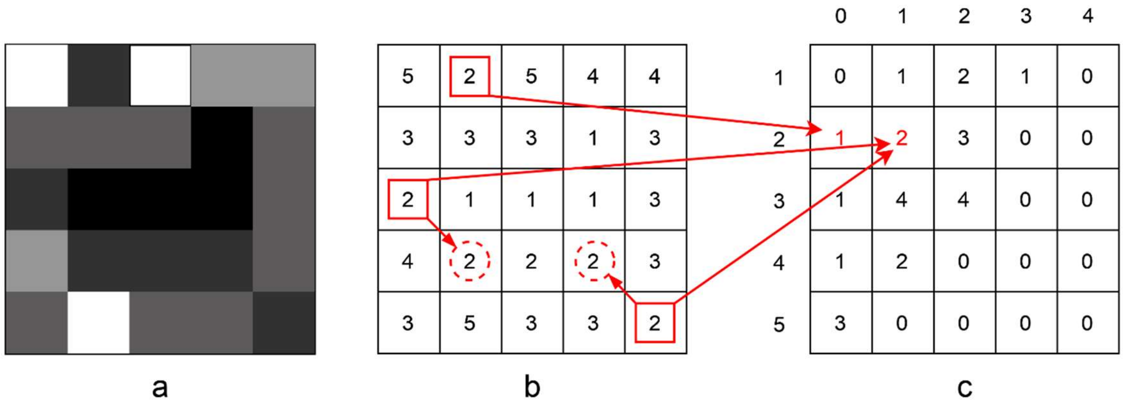


Figure 2.23. Working principle of GLDM a) A 5x5 gray image with 5 gray level values, b) the equivalent numerical matrix of the left image, c) the GLDM matrix of the image

The GLDM in Figure 2.23 was created by calculating the dependencies for $\alpha = 0$ and a neighbor distance of 1. Rows represent the pixel gray values and columns represent their number of dependencies. For pixels with value 2, one of them has 0 dependency and two have 1 dependency as shown in Figure 2.23 b. Thus, number 1 was written in the second row of the first column and 2 was written in the second row of the second column. From the GLDM, 14 features can be extracted using the Pyradiomics library.

2.3.3. Deep Learning

DL is a branch of ML that uses NN to solve complex problems. The term "deep learning" refers not to any particular type of deeper knowledge that is created by the process, but rather to the concept of building layers in a successive manner. The depth of a model is defined as the number of layers that contribute to it (Ketkar & Moolayil, 2021). The NN in DL have more than one hidden layer.

2.3.3.1 Convolutional Neural Networks

CNN are deep neural networks that are used for computer vision and pattern analysis (Goodfellow et al., 2016). CNN are used on images to extract features and patterns to be classified and/or segmented later(Zhou et al., 2018). Figure 2.24 shows a visualization of a CNN architecture.

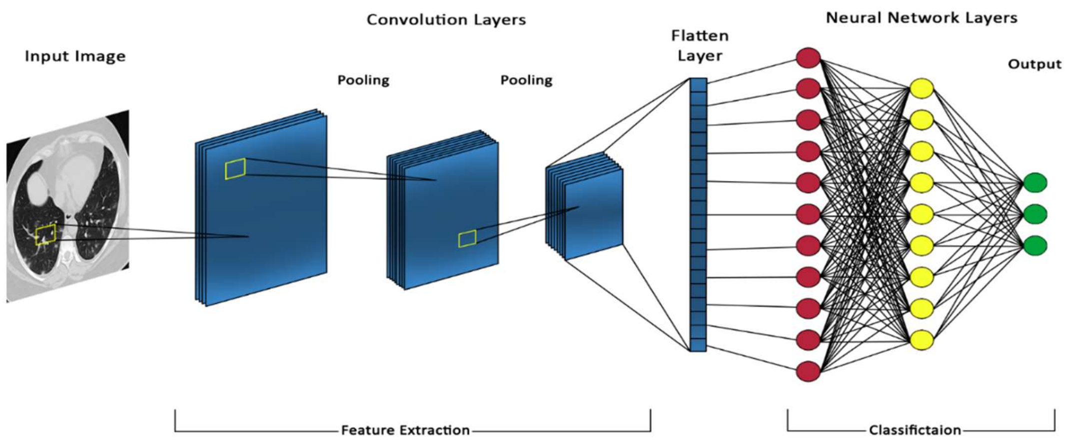


Figure 2.24. A typical CNN architecture scheme shows all hidden layers

The term ‘convolutional’ indicates that a mathematical operation called ‘convolution’ is been employed. Instead of the general matrix multiplication, the images are convolved with specific filters called ‘kernel’ to produce a feature map. Each convolution layer is followed by a non-linear layer called the pooling layer. The filters produced by the convolution layer are pooled by a pooling layer to reduce the dimension of the feature map. The last layers of the CNN are the fully connected layers in which the model learns the features to classify and/or segmented them. The last fully connected layer is the output layer of the network. All the layers between the input and the output layers are called 'hidden layers.

2.3.3.1.1. Convolution layer

The convolution layer is the main part of the CNN and it's where the convolution operation is employed. The purpose of this layer is to find patterns in image regions that mostly can't be noticed by human eyes. The low-level features like edges and circles are extracted in the first convolution layers and the high-level features like eyes, nose, and mouth are extracted in the deepest convolution layers. Most of the heavy operations of the CNN are done in this layer. Equation (2.22) represents the convolution operation;

$$F(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{2.22}$$

where F is the feature map produced by convolving the kernel K with the image I. Therefore, $I*K$ specifies the convolution operation. The indexes I and j are associated with the image matrix, whereas m and n are associated with the kernel matrix. If the kernel size in convolution is 3x3, then the indexes m and n will vary from -1 to 1. Figure 2.25 demonstrate a basic convolution operation.

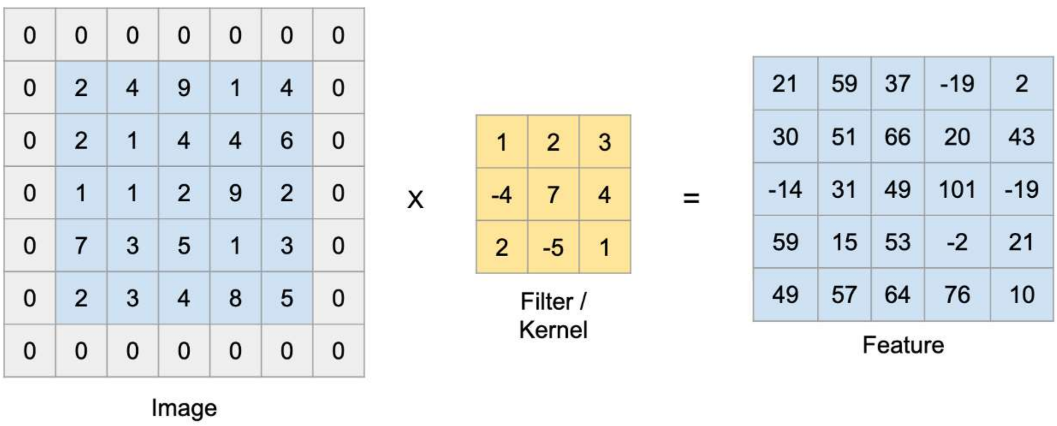


Figure 2.25. Demonstration of a basic convolution operation (Patel, 2019)

The size of the output in a normal convolution operation will be smaller than the size of the input. To prevent the shrinking of the output new rows and columns of zeros are added around the input image. This operation is called padding. The number of rows and columns that should be added depends on the size of the kernel. For a 3x3 kernel, 1 row and 1 column are added, while for a 5x5 kernel 2 rows and 2 columns are added. Padding can be seen in Figure 2.25 as a gray frame of zeros around the input image.

2.3.3.1.2. Pooling Layer

The pooling layer is a non-linear down-sampling layer that is used to reduce the dimensionality of the feature maps(Weidman, 2019). As the convolution layer, the pooling layer is constructed of filters/kernels that are convolved with the output of the convolution layer. There are three different pooling operations used in CNN such as; max-pooling, where the maximum values of pixels are chosen, min-pooling, where the minimum values of pixels are chosen, and average pooling, where the average values of pixels are chosen. Figure 2.26 shows examples of max and average pooling.

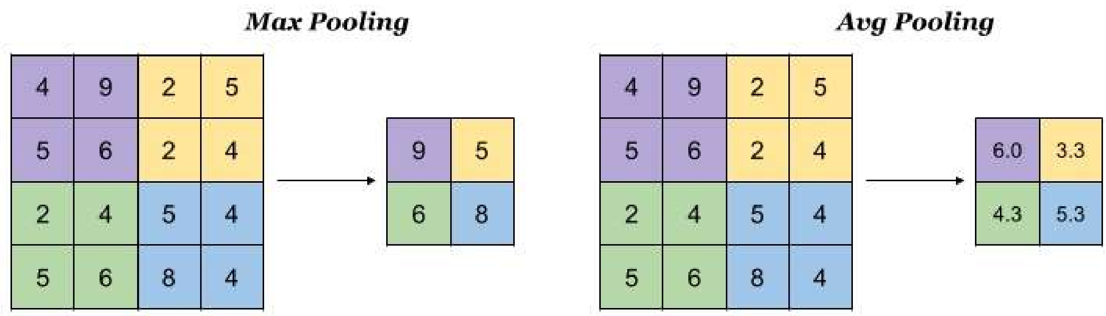


Figure 2.26. Examples of max pooling and average pooling (Reji, 2018)

2.3.3.1.3. Flatten Layer

This layer comes after the pooling layer and is used to convert the 2D image matrix into a 1D vector to be presented to the next fully connected layers.

2.3.3.1.4. Fully Connected Layers

This layer is basically a neural network that has one or more hidden layers. This layer is also called a dense layer because it’s densely connected. FC layers are responsible for the classification process in CNNs. Most of the network’s weights exist in this layer.

2.3.3.1.5. Activation Function

This function is a mathematical function that activates or deactivates a neuron based on its input. There are many activation functions used in the NN and some of them are the Sigmoid function, Rectified Linear Unit function, Hyperbolic Tangent function, and SoftMax function.

Sigmoid function: the sigmoid function is a nonlinear activation function that is used mostly at the output layer when two classes are expected. The function returns a value in the range between 0 and 1. If the input is a very large positive number the output will tend to 1 and if it's a very large negative number the output will tend to 0. This function has a threshold of 0.5 where the input is 0 as in Figure 2.27. The formula used for this function is in Equation (2.23);

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.23}$$

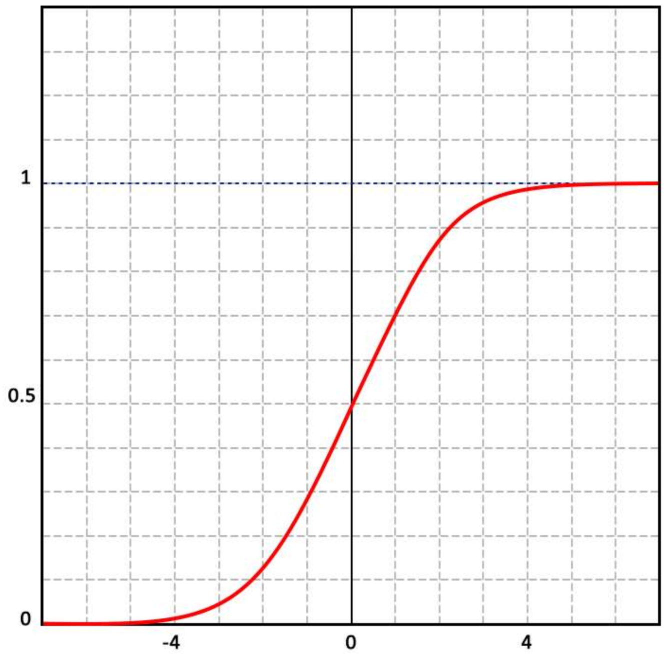


Figure 2.27. Sigmoid activation function graph

Rectified Linear Unit (ReLU) function: ReLU is the most prevalent nonlinear activation function used in NN. The ReLU function is commonly used in the neural network layers and in the convolution layers after each convolution operation to decrease the linearity in the network. This function returns zero if the input is zero or less i.e., negative, and returns the input if it's larger than zero. Figure 2.28 shows a graph of the ReLU function. The formula for this function is in Equation (2.24);

$$f(x) = \max(0, x) = \begin{cases} 0 & for\ x \leq 0 \\ x & for\ x > 0 \end{cases} \tag{2.24}$$

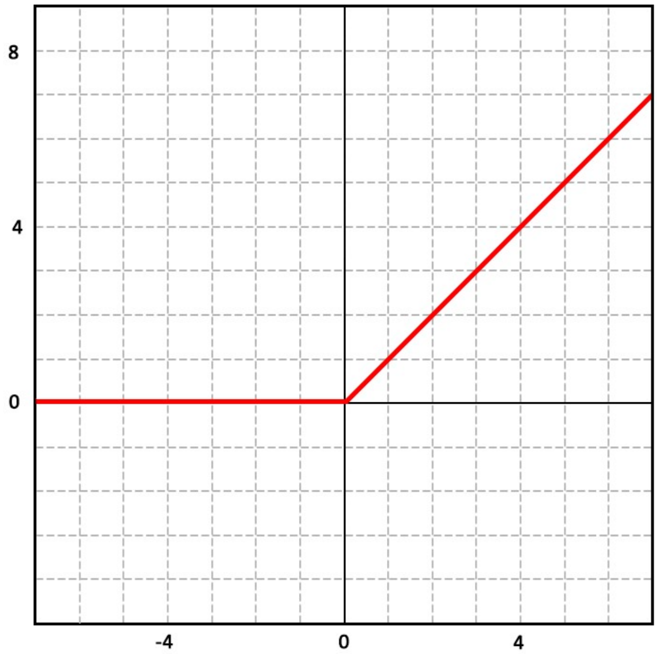


Figure 2.28. ReLU activation function graph

Hyperbolic Tangent (Tanh) function: The Tanh function is another nonlinear activation function used in NN. This function is similar to the sigmoid function and can be used for two classes of output. The output range of this function ranges from -1 to 1 and is symmetric around the origin. This function could be sometimes preferred over sigmoid due to its wider output range. For that reason, it gives better performance and accuracy over the sigmoid function. The graph of this function is shown in Figure 2.29. The formula for this function is in Equation (2.25);

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.25}$$

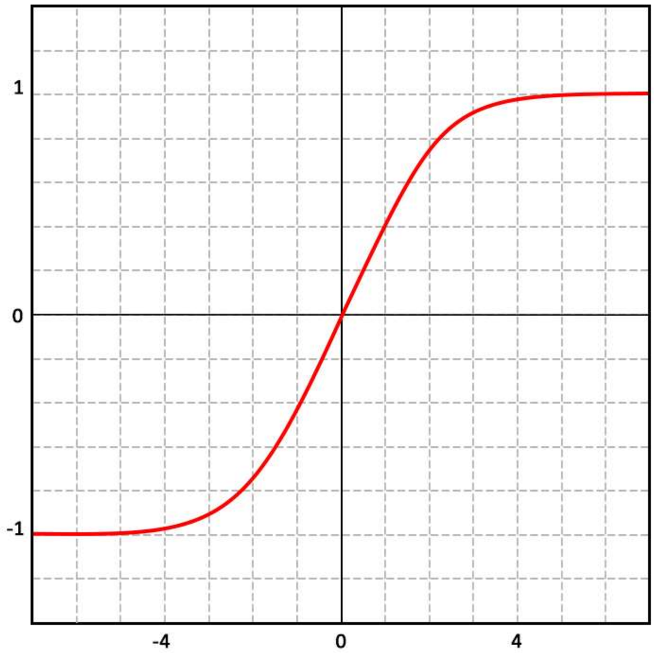


Figure 2.29. Tanh activation function graph

SoftMax function: is a nonlinear activation function that is used at the output layer for a multiclass output. This function takes a vector of values as an input and returns a vector of their probability distribution where the sum of them equals 1. The formula for this function is given in Equation (2.26);

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_i}} \tag{2.26}$$

where $f(x_i)$ is the output vector, x_i is the input vector, i is the position of the value 1 in the one-hot encoded vector, n is the number of classes, $j=1, 2, ..., n$.

2.3.3.2 Transfer Learning

Transfer learning is one branch of AI and was first introduced by Stevo Bozinovski in 1976. Transfer learning has been used in many fields such as industry, agriculture, and medical diagnosis (Dekhtiar et al., 2018; Lih et al., 2020; Rahimzadeh & Attar, 2020). Transfer learning, as the name refers, is known as transferring the knowledge gained by a previously trained model on a specific problem to solve another problem without the need to train a new model from scratch. Although DL models need a lot of data to learn, transfer learning models can be used with small data since they were trained on big data. Convolutional neural network-based models are the most popular pre-trained models used in transfer learning. ImageNet is a big dataset that has over a million images for 1000 classes which developers in the research field use to train their CNN architectures in competitions to show the capability of their architectures in classification and object recognition. VGG-19, ResNet-50, EfficientNet-B0, DenseNet-201, and Xception are some of best the pre-trained models that can be used.

2.3.3.2.1. VGG-19

VGG-19 was proposed in 2015 by the Visual Geometry Group of Oxford University (Simonyan & Zisserman, 2015). It has 19 deep layers which consist of 16 convolutional layers, 5 MaxPool layers, 3 fully connected layers, and 1 SoftMax layer. The model was trained on the ImageNet dataset to classify 1000 classes and has 143.7M parameters. It accepts RGB images as input with a shape of (224,224,3). Its convolution and pooling filters have a size of 3x3 and 2x2 respectively. The first two fully connected layers have 4096 neurons each and the last one, which is the output layer, has 1000 neurons with SoftMax function. All the layers in this model use the ReLU activation function. VGG-19 architecture is shown in Figure2.30.

2.3.3.2.2. ResNet-50

ResNet-50 is a CNN-based model inspired by VGG and was proposed in 2016 by Kaiming He (He et al., 2016). It has 50 deep layers which consist of 48 convolutional layers, 1 MaxPool layer, and 1 layer that has Average Pool layer and one 1000-neurons fully connected output layer with SoftMax. The model was also trained on the ImageNet dataset to classify 1000 classes and has 25.6M parameters. It accepts RGB images as input with a shape of (224,224,3). The first convolution filter has a size of 7x7 while the rest convolution and pooling filters have a size of 3x3 and 2x2 respectively. The filters are grouped in blocks and each block has three filters. This architecture, shown in Figure 2.30, uses residual networks in which a short connection is added to each block of filters. This is done to solve the degradation problem which is the decrease in performance of networks with large number of layers.

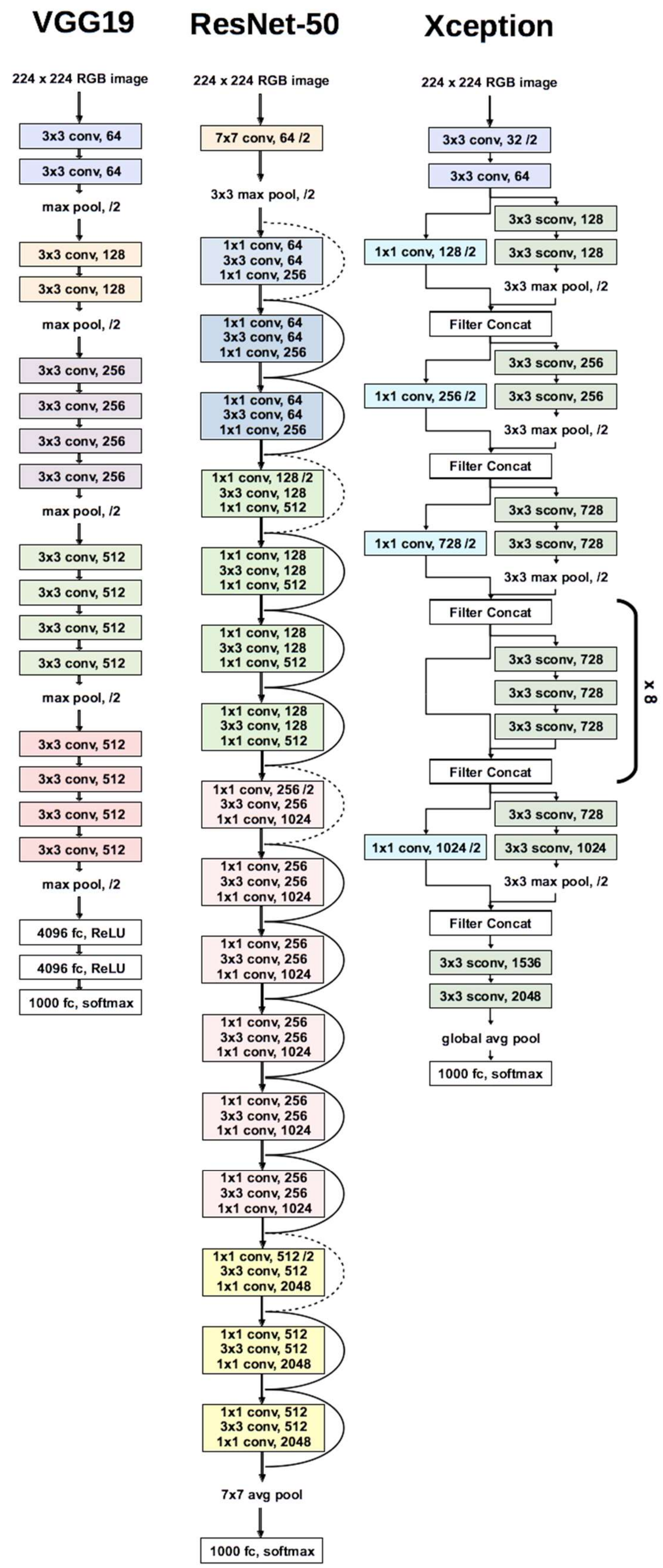


Figure 2.30. Architecture of VGG-19, ResNet-50, and Xception (Leonardo et al., 2019)

2.3.3.2.3. DenseNet-201

Huang et al. were inspired by the ResNet model and proposed DenseNet-201 in 2017 (Huang et al., 2017). DenseNet-201 has 201 layers which consist of a 7x7 convolution filter followed by a 3x3 MaxPool as the first layer, 196 convolutional layers with 3 average pooling layers in between, and 1 output layer with global average pooling and 1000-neurons fully connected layer. This model has 20.2M parameters and accepts RGB images as input with a shape of (224,224,3). The filters are grouped in 4 dense blocks. The layers inside the dense blocks are densely connected in which each layer takes its input from all the preceding layers and passes it to the subsequent layers as in Figure 2.31.

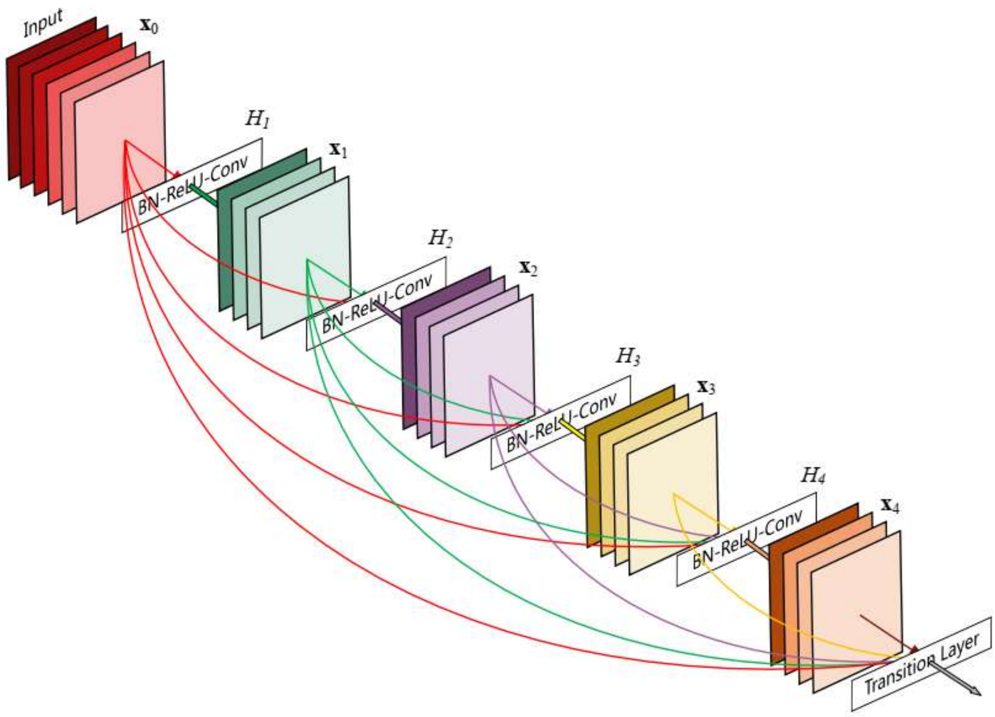


Figure 2.31. DenseNet architecture shows the dense connection between the blocks (Huang et al., 2017)

2.3.3.2.4. Xception

Xception is based on Inception architecture except it involves depth-wise separable convolutions. This model is 36 layers deep and was proposed in 2017 by Francois Chollet at Google, Inc.(Chollet, 2017). The model has 22.9M parameters and was trained on the ImageNet dataset to classify 1000 classes and accepts RGB images as input with a default shape of (299,299,3). It can accept images with the shape of (224,224,3) as well. Depth-wise separable convolutions are convolutions that combine a depthwise convolution (which convolves each filter channel with each input channel) and a pointwise convolution (which combines the produced output channels by convolving them with a 1x1 filter). Figure 2.32 demonstrate the depth-wise separable convolution compared to the normal convolution operation. Figure 2.30 shows the architecture of the Xception model.

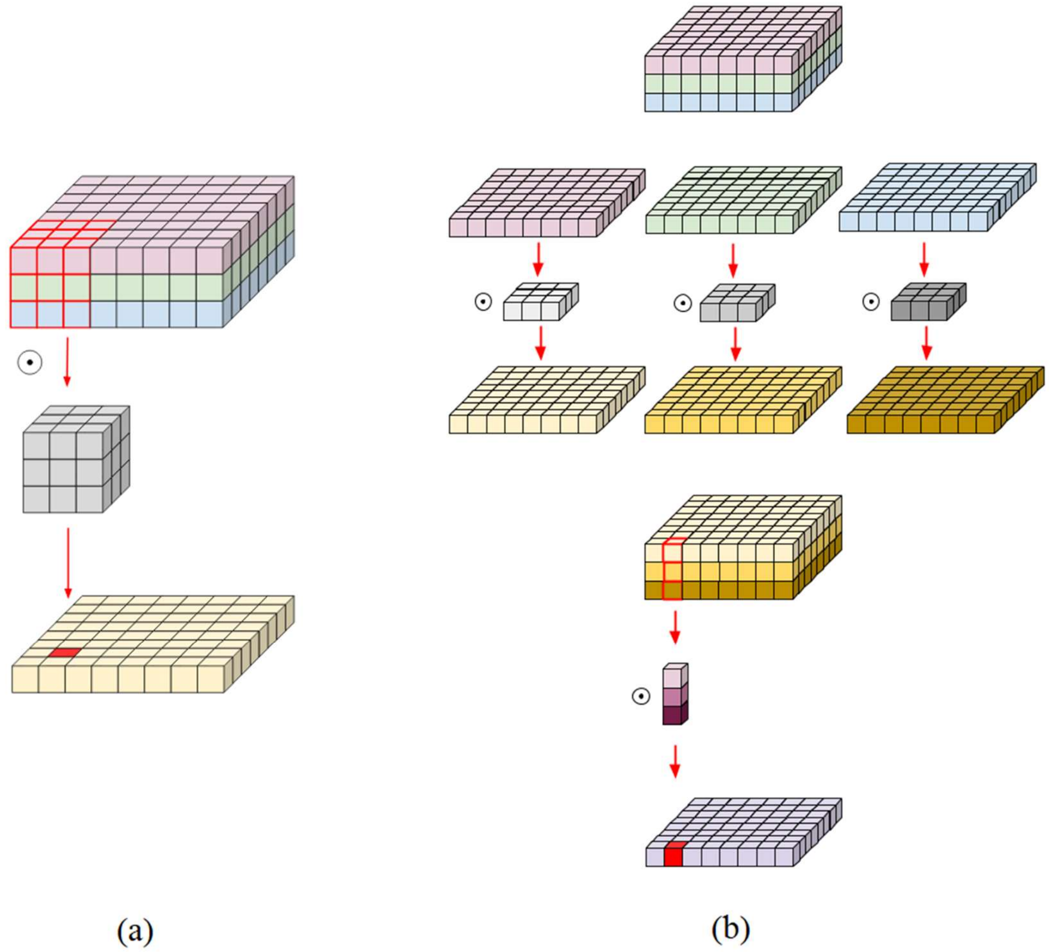


Figure 2.32. Convolution operation in CNN a) normal convolution, b) depth-wise separable convolutions (Eli Bendersky's Website, 2018)

2.3.3.2.5. EfficientNet-B0

EfficientNet, first introduced in 2019 by Tan and Le, is among the most efficient CNN-based models reaching State-of-the-Art accuracy on the ImageNet dataset(Tan & Le, 2019). Although EfficientNet-B0 is 237 layers deep (Agarwal, 2020), it has only 5.3M parameters making it one of the most efficient pre-models. The model accepts RGB images as input with a shape of (224,224,3). In general, a CNN model can be scaled in terms of width, depth, or resolution while in EfficientNet they are all scaled up as shown in Figure 2.33.

The comparison of specifications between the pre-trained models is shown in Table 2.8. The table shows that VGG-19 is the largest model in terms of size and number of parameters yet the shallowest in terms of the number of layers model, while EfficientNet-B0 is the smallest and deepest model.

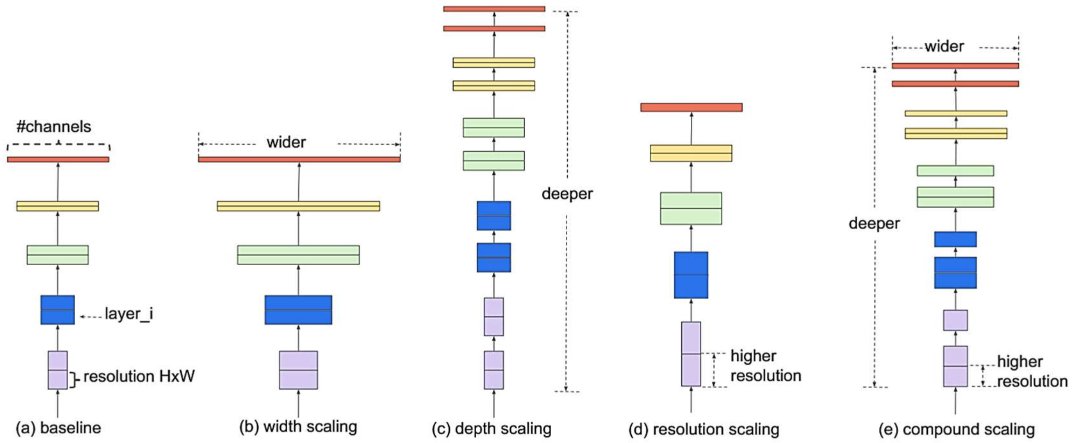


Figure 2.33. CNN model scaling examples (Tan & Le, 2019)

Table 2.1. A comparison between pre-trained models

Model	**Year of publishing**	**Depth**	**Size (MB)**	**Number of parameters (Million)**
VGG-19	2014	19	549	143.7
ResNet-50	2015	50	98	25.6
DenseNet-201	2017	201	80	20.2
Xception	2017	36	88	22.9
EfficientNet-B0	2019	237	29	5.3

2.3.4. Evaluation Methods

Evaluating ML and DL models' performance is a necessary step to measure how well the models learned to classify new data. The outputs of the models are predicted labels and comparing them to the actual/true labels gives us the performance results.

2.3.5. Cross-Validation

CV is a technique used in supervised ML to evaluate ML algorithms on a small portion of the dataset. The concept of this technique is to divide the dataset into two parts one for training the algorithm and the other for testing/validating (Scikit-learn, n.d.). CV estimates how well a ML model generalizes and performs on new or unseen data. The CV technique can also be used to find the best hyperparameters to use with the algorithms (Burkov, 2019).

K-fold CV is a popular type of CV that ensures the model's performance is not related to a certain test subset. Instead, the model is evaluated on different subsets. Figure 2.34 shows how k-fold CV works.

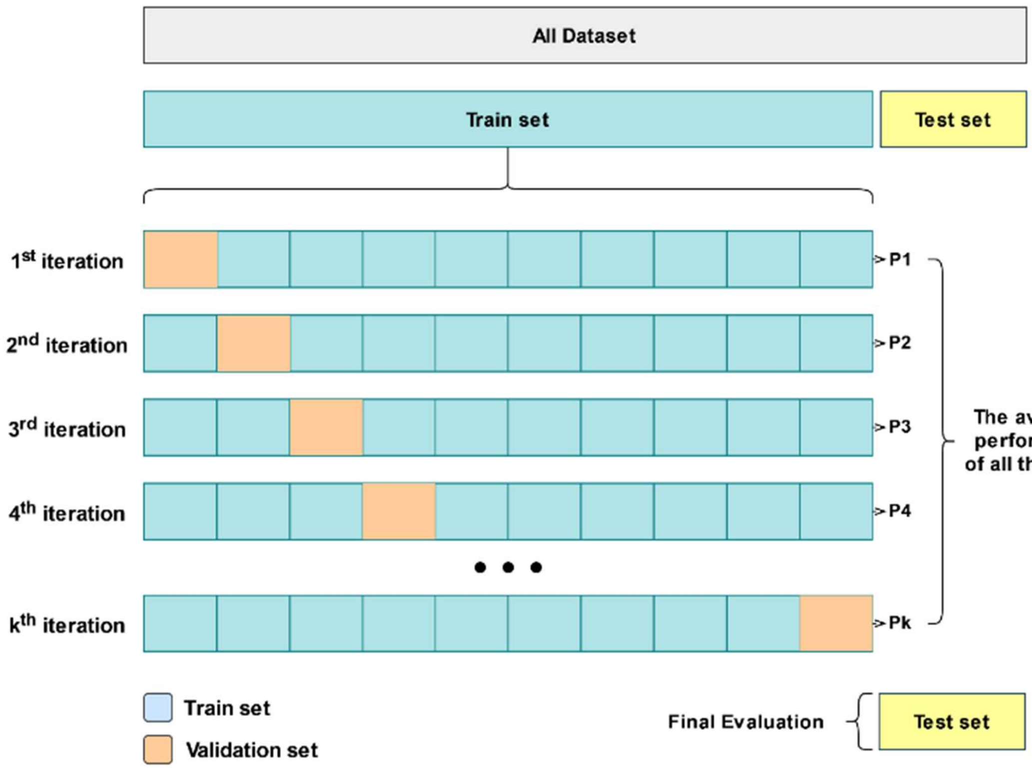


Figure 2.34. k-fold cross-validation working principle

The dataset is split into predefined ratio of training set and a test set in which the test set is left for final evaluation. Then, the train set is split into k different equally parts called folds. The model is trained on k-1 folds and validated on the remaining one. This process is repeated for k iterations such that each fold is used as a validation set. The average performance is taken from the performances of all the folds. The test set left aside is used for the final evaluation of the model's performance.

2.3.6. Confusion Matrix

The comparison of the predicted labels to true labels produces four outcomes; true positive (TP) is the number of positive class samples that were predicted to be positive, true negative (TN) is the number of negative class samples that were predicted to be negative, false positive (FP) is the number of negative class samples that were predicted to be positive, and false negative (FN) is the number of positive class samples that were predicted to be negative. These outcomes are used to generate the confusion matrix. Figure 2.35 shows a confusion matrix for two classes.

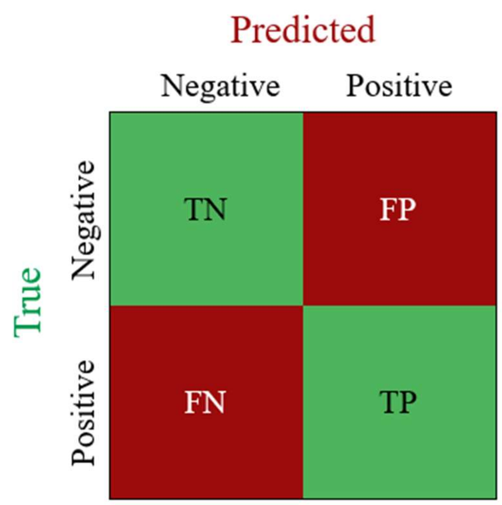


Figure 2.35. A confusion matrix for binary classification

2.3.7. Classification Metrics

Classification metrics are also used as a measurement for models' performance. Metrics such as accuracy, precision, sensitivity, and F1-score are calculated from the confusion matrix as follows:

Accuracy is defined as percentage of true predictions relative to the total number of predictions and is given by equation (2.27);

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.27}$$

Precision is defined as the percentage of true positive class predictions relative to total number of positive class predictions and is given by equation (2.28);

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.28}$$

Sensitivity is defined as the percentage of true positive class predictions relative to the number of true positive class predictions and false negative class predictions and is given by equation (2.29);

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{2.29}$$

The F1-score is defined as the weighted average of precision and sensitivity and is given by equation (2.30);

$$\text{F1} - \text{score} = \frac{2 * \text{Precision} * \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{2.30}$$

Area Under receiver operating characteristics Curve or AUC is another evaluation metric used in classification problems. The ROC curve plots the TP rate against the FP rate. On the Y-axis of the ROC curve is the TP rate and on the X-axis is the FP rate. TP rate is another term for Sensitivity and can be calculated by equation (2.31) and FP rate by equation (2.32),

$$\text{TP rate} = \frac{TP}{TP + FN} \tag{2.31}$$

$$\text{FP rate} = \frac{FP}{FP + TN} \tag{2.32}$$

Figure 2.36 illustrates different possibilities of the ROC curve. AUC calculates the area under the ROC curve. The higher the AUC value the better the performance of the model.

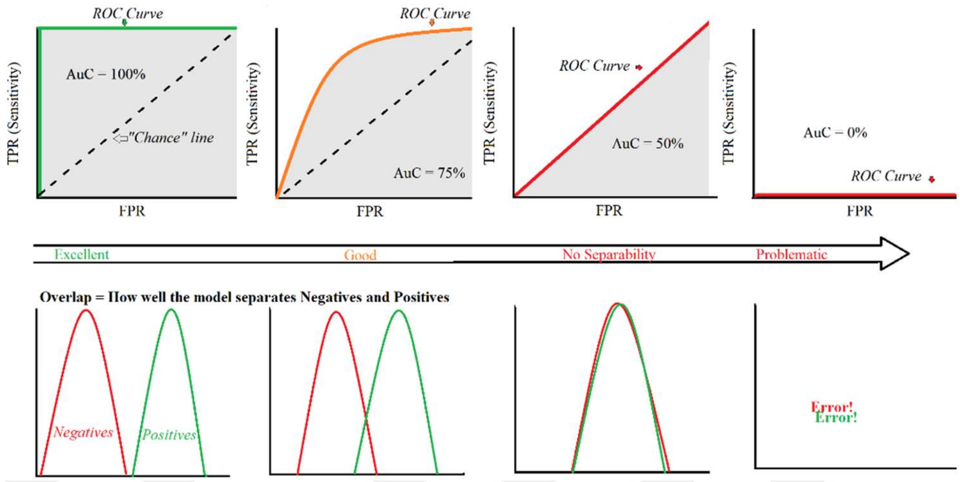


Figure 2.36. Different possibilities of the ROC curve (Glen, 2019)

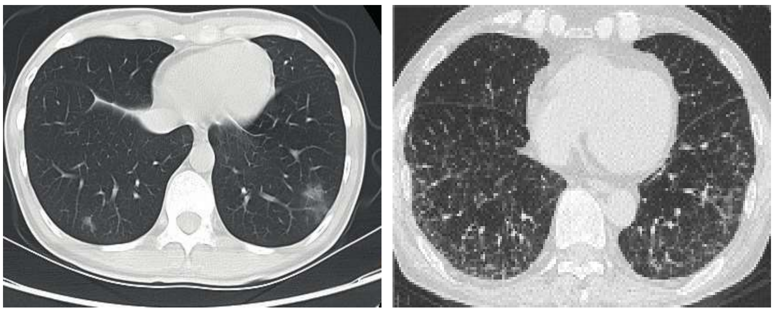
3. MATERIAL AND METHODS

Within the scope of this thesis, three public COVID-19 CT datasets namely “COVID-CT”, “Large COVID-19 CT scan slice dataset”, and “CT Scans for COVID-19 Classification” were used to evaluate the ML and DL algorithms. For the classification process, several ML classifiers and pre-trained DL models were trained and tested on the datasets.

3.1. Datasets

3.1.1. COVID-CT

This dataset contains 349 CT images with COVID-19 of 216 patients and 397 CT images of normal patients. The images were acquired from COVID19-related articles published in medRxiv, bioRxiv, NEJM, JAMA, Lancet, and other archives and journals by (Zhao et al., 2020). A senior radiologist at Tongji Hospital in Wuhan, China has confirmed the usability of this dataset by diagnosing and treating a significant number of people that were infected with COVID-19 during the pandemic. Figure 3.1 shows some samples of the dataset for both classes. In addition, all the details of this dataset are listed in Table 3.1.

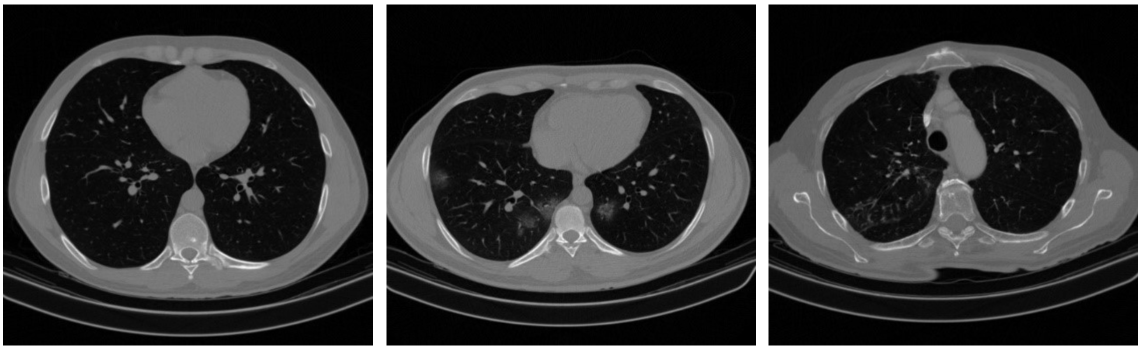


a) b)

Figure 3.1. Image samples of COVID-CT dataset a) COVID-19 infected patient, b) non-COVID-19 patient

3.1.2. Large COVID-19 CT Scan Slice Dataset

This dataset has three classes, 7,593 COVID-19 CT images were obtained from 466 patients, 6,893 normal CT images were obtained from 604 patients, and 2,618 Community-Acquired (CAP) CT images were obtained from 60 patients. The dataset was collected by (Maftouni et al., 2021) from 7 publicly available datasets. Some samples of this dataset for the three classes are shown in Figure 3.2. Furthermore, Table 3.1 shows the details of this dataset.



a) b) c)

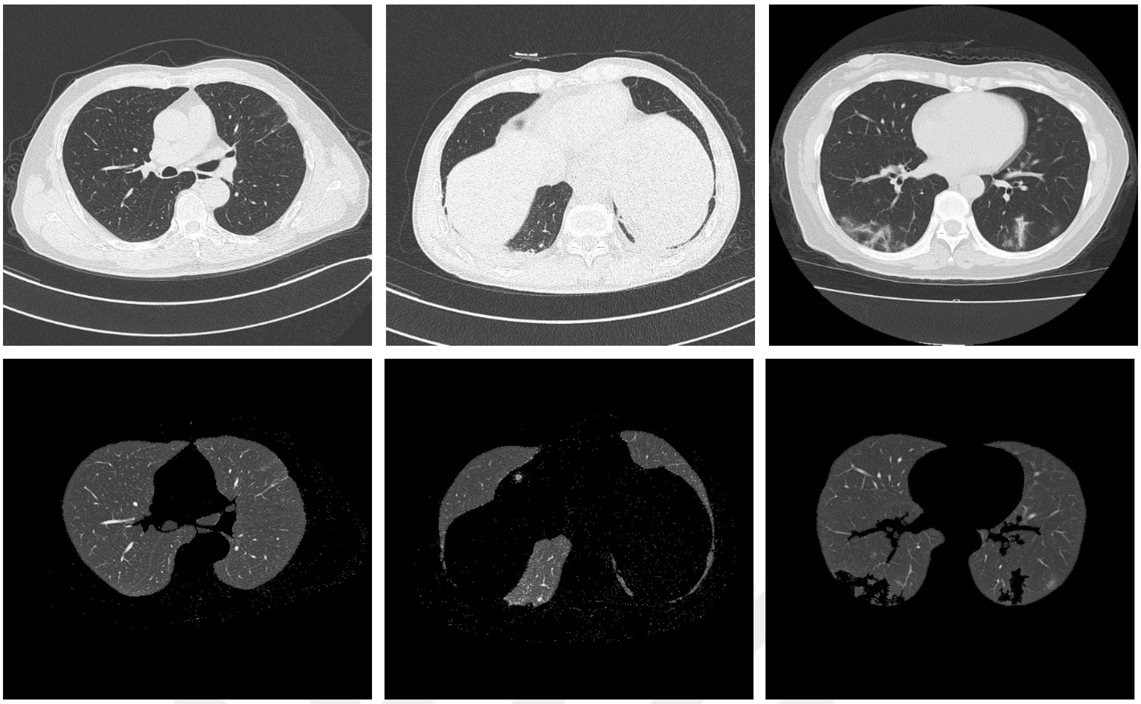
Figure 3.2. Image samples of Large COVID-19 CT scan slice dataset a) non-COVID-19 patient, b) COVID-19 infected patient, c) CAP infected patient

3.1.3. CT Scans For COVID-19 Classification

This dataset was randomly collected by (Ning et al., 2020) from 61 and 43 patients with and without COVID-19 pneumonia. It consists of 5705 non-informative CT images (NiCT), 4001 positive COVID-19 CT images (pCT), and 9979 negative COVID-19 CT images (nCT). This dataset contains two versions; One without pre-processing and the other with a segmentation pre-processing technique in which lung regions were segmented. Samples from the two versions of this dataset are illustrated in Figure 3.3 whereas more details of them are in Table 3.1.

Table 3.1. A comparison between the datasets used in this thesis

Dataset name	**Year of publication**	**Number of images**	**Number of classes**	**Size**
COVID-CT	2020	746	2 (COVID-19, non-COVID-19)	97.23 MB
Large COVID-19 CT scan slice dataset	2021	17,104	3 (non-COVID-19, COVID-19, CAP)	2.28 GB
CT Scans for COVID-19 Classification	2020	19,685	3 (nCT, NiCT, pCT)	3.05 GB



a) b) c)

Figure 3.3. Image samples of CT Scans for COVID-19 Classification dataset, top row from the original version and bottom row from the pre-processed version a) negative COVID-19, b) non-informative image, c) positive COVID-19

3.2. Experiment

The experiment was implemented in two stages on a Windows 10-based workstation PC with an intel core i7 3202 2.00 GHz CPU, Nvidia GTX 1080ti 8GB GPU, and 32GB RAM using Python programming language along with scikit-learn, TensorFlow, and Keras ML libraries(Chollet & others, 2015).

In the first stage, radiomics features were extracted from the "COVID-CT" dataset to train ML algorithms such as SVM, k-NN, RF, and XGBoost to classify COVID-19. As for the second stage, CNN-based pre-trained DL models such as VGG-19, ResNet-50, EfficientNet-B0, DensNet201, and Xception were used with "Large COVID-19 CT scan slice dataset" and "CT Scans for COVID-19 Classification" datasets to classify COVID-19.

3.2.1. Machine Learning Stage

In this stage, we first applied feature extraction method to "COVID-CT" dataset to extract the meaningful features that characterize the infected tissues of the lung in the CT images. Radiomics features such as shape-based, first-order-based, and texture-based features were extracted using the Pyradiomics library. GLCM and GLRLM were extracted for each angle and their mean was utilized for feature extraction. Moreover, the distance δ value for which GLCM, NGTDM, and GLDM were extracted has been kept as the default (1) since it didn't change the results significantly. Then, the features set was standardized and split into 80% training set and 20% test set. The test set was kept for later final evaluation. The 80% training set was utilized for hyperparameter tuning/optimization. The optimization method used to find the best hyperparameter values is called GridSearchCV in the scikit-learn library. What this method does is train several algorithms using predefined hyperparameters and return the performance result of each model with its corresponding hyperparameter values. GridSearchCV method performs a k-fold CV technique when training algorithms for better generalization on the training set. Table 3.2 lists the hyperparameters provided to the optimization method in addition to the best hyperparameters it returned for every algorithm. The best hyperparameters are bolded in the table.

Table 3.2. Hyperparameters of machine learning algorithms

Classifier	**Hyperparameter Grid**
SVM	Kernel = {linear, poly, **RBF**, sigmoid}, C = {10^{-2},10^{-1}, 10^{0},10^{1},$\mathbf{10^{2}}$,10^{3}}, Gamma = {10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3},$\mathbf{10^{-2}}$}
k-NN	k = {3, **5**, 7}, Distance metrics = {**Euclidean**, Manhattan, Minkowski}
RF	Min samples leaf = {**1**, 3, 5, 7}, Min samples split = {**2**, 8, 10, 12} Number of trees = {10,50,100, **500**,1000}
XGBoost	Learning rate = {0.01, 0.1, **0.3**, 1.0}, Max depth= {4, **6**, 8, 10} Number of trees = {10, 50, **100**, 500}

The left aside 20% test set was used to evaluate the final model by making it predict the labels of the test samples and then comparing them with the true labels. The process of finding the best values for hyperparameters and training and evaluating were implemented for the other ML algorithms as well. Figure 3.4 shows the workflow diagram of this stage.

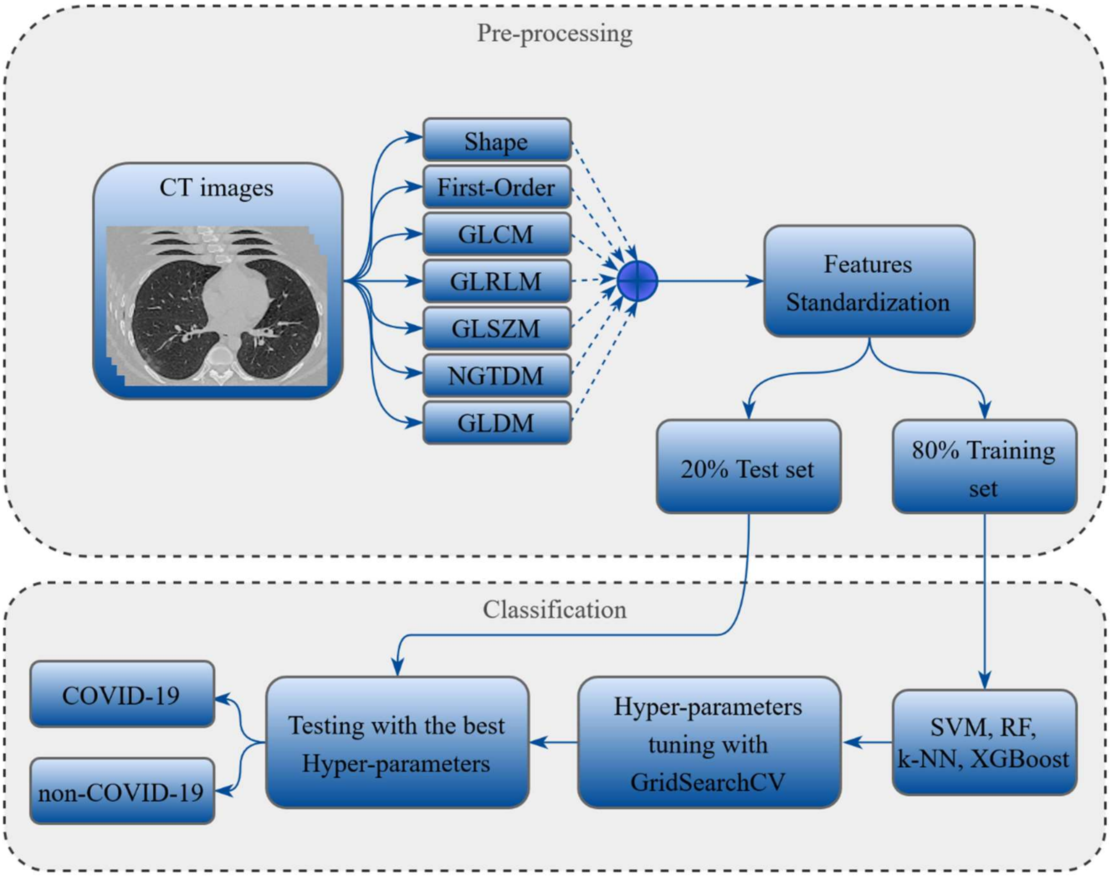


Figure 3.4. Machine learning stage workflow diagram

3.2.2. Deep Learning Stage

In this stage, "Large COVID-19 CT scan slice dataset" and "CT Scans for COVID-19 Classification" were used to train transfer learning models such as VGG-19, ResNet-50, EfficientNet-B0, DenseNet-201, and Xception. The experiment was carried out separately for each dataset. Since the datasets have a large number of images enough to train the models, there was no need to use augmentation techniques.

All the models were loaded with ImageNet weights and an input shape of 224,224 was used. Therefore, the CT images were resized to 224,224 to match the input of the models.

After that, the dataset was normalized and split into an 80% training set and a 20% test set. Fine-tuning the models is an important practice before the training process. Fine-tuning was implemented in three steps;

- freezing the top layers i.e., the convolutional layers in which the pre-trained weights are
- removing the fully connected layers since the models were trained on a 1000-class dataset
- adding two fully connected layers with initialized weights for classification

The added fully connected layers consist of one layer that has 256 neurons with the ReLU activation functions and an output layer of three neurons with SoftMax activation Function. The training set was fid into the model to extract the features using their convolutional layers and then train the fully connected layer on these features. A 20% of the training set was used for the validation process in each epoch All the models were fine-tuned and then trained for 50 epochs with Adam optimizer to update the weights during training and enhance the accuracy in every epoch. After training the models, we evaluated them on the 20% test set to determine their prediction's performance. The workflow of this stage is demonstrated in Figure 3.5.

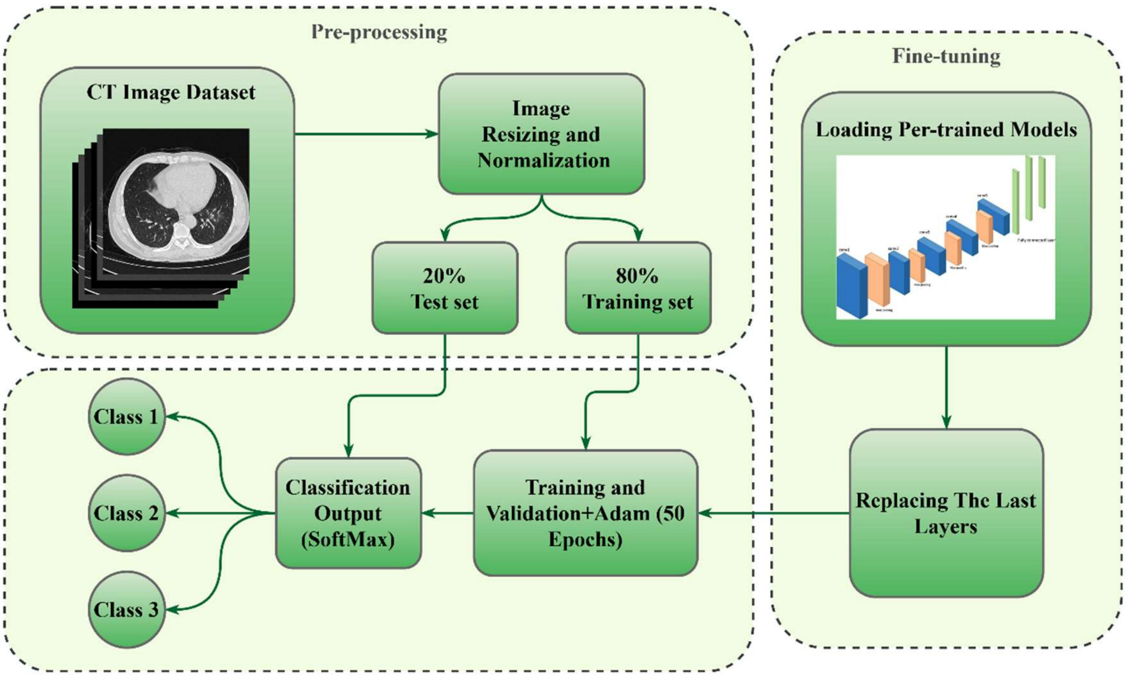


Figure 3.5. Deep learning stage workflow diagram

4. RESULT AND DISCUSSION

In this study, we have used ML and DL methods to classify COVID-19 using CT image datasets. Several ML algorithms and DL pre-trained models were trained on the datasets and evaluated. Evaluation metrics and confusion matrices were obtained for both stages. For ML, confusion matrices of the best models only were obtained. We also have included the time that classifiers took to predict the test set. For DL, we also included the training and validation accuracy vs epochs' graph. The results are presented and discussed in this section.

4.1. Machine Learning Results

These results were obtained when ML models were trained using the best hyperparameters obtained by the GridSearchCV method on the 80% training set of "COVID-CT" dataset and evaluated on the 20% test set. One thing to mention is that we have excluded the results of NGTDM and GLDM for the mean accuracy being less than 70%.

The results obtained when classifying the features extracted by the Shape 2D method are shown in Table 4.1. We can see that RF scored the highest in terms of accuracy with a 93.82%, while SVM has the lowest accuracy of 88.61%. RF also has the best sensitivity of 93.37%. The confusion matrix of RF is shown in Figure 4.1. From the confusion matrix, we can see that the number of the test samples is 150, 70 of which are for the positive class and 80 for the negative class. The number of true positives is 65 meaning 5 positive samples were misclassified. For the negative class, the true negative is 77 with 3 misclassified samples.

Table 4.1. Results of Shape 2D features (%)

Metrics	Accuracy	Precision	Sensitivity	F1-Score	Time (s)
SVM	91.3	91.3	90	90.64	0.001
RF	93.82	92.34	93.37	92	0.082
k-NN	88.75	87.95	91	89.38	0.014
XGBoost	91.42	91.83	93.22	92.37	0.005

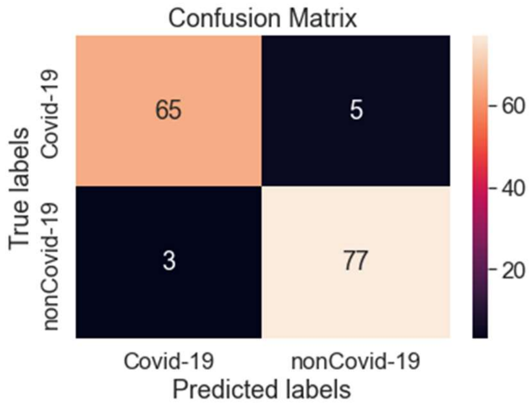


Figure 4.1. Confusion matrix of Shape 2D results with RF classifier

In Table 4.2, we can notice that FOS features results are higher than Shape 2D results. The highest accuracy was achieved by the XGBoost classifier with a classification accuracy score of 98.65%. It also scored the highest sensitivity result of 96.98%. This time, k-NN gave the lowest performance with an 83.9% accuracy rate. Figure 4.2 shows the confusion matrix of XGBoost in which all the positive samples were correctly predicted. While three of the negative samples were misclassified.

Table 4.2. Results of First-Order Statistics features (%)

Metrics	Accuracy	Precision	Sensitivity	F1-Score	Time (s)
SVM	94.36	100	88.9	94.39	0.001
RF	94.1	96.1	93.59	94.25	0.061
k-NN	83.9	90.86	81.17	84.98	0.014
XGBoost	98.65	99.73	96.98	98.35	0.011

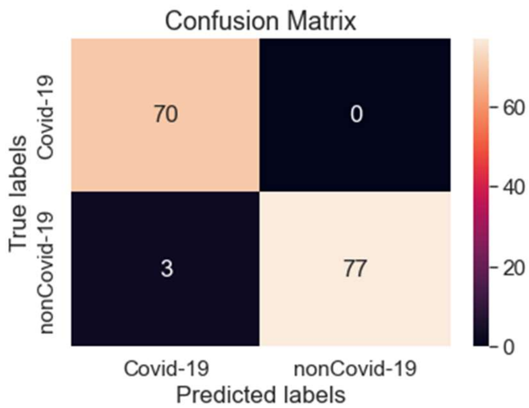


Figure 4.2. Confusion matrix of First-Order Statistics results with XGBoost

The results of GLCM features in Table 4.3 show significant degradation in the performance of all models with a mean accuracy of 75.24%. The highest accuracy is 77.9% and the lowest is 70.37% obtained by XGBoost and SVM respectively. The sensitivity along with the other metrics is also low with 79.67% being the highest by RF.

Table 4.3. Results of GLCM features (%)

Metrics	Accuracy	Precision	Sensitivity	F1-Score	Time (s)
SVM	72.66	73.21	75.98	75.25	0.001
RF	77	79.57	79.67	79.68	0.096
k-NN	73.4	74.5	76.4	74.1	0.008
XGBoost	77.9	79	77.9	78	0.003

In Table 4.4, the GLRLM features results as well show a low classification performance for all the models, especially for SVM of 68.78% accuracy. The mean accuracy is 76.16% and the highest is 78.42% for XGBoost. The sensitivity has also improved with the highest score of 82.46% by RF.

Table 4.4. Results of GLRLM features (%)

Metrics	Accuracy	Precision	Sensitivity	F1-Score	Time (s)
SVM	72.6	70.4	71.42	70.93	0.002
RF	79.35	79	82.46	80.9	0.064
k-NN	74.27	75.6	78.75	77.96	0.008
XGBoost	78.42	79.42	82.42	78.3	0.003

For GLSZM results in Table 4.5, we can notice a slight performance improvement compared to GLRLM except for SVM which has a lower score. The mean accuracy of GLRLM classification is about 78.45%. RF has achieved a slightly better results than XGBoost with an accuracy of 81.25% and a sensitivity of 84.14%.

Table 4.5. Results of GLSZM features (%)

Metrics	Accuracy	Precision	Sensitivity	F1-Score	Time (s)
SVM	73.99	74.35	80.13	78.21	0.001
RF	81.25	82	84.14	82.85	0.057
k-NN	78	79.3	81.64	80.4	0.008
XGBoost	80.54	82.11	82.8	82.18	0.003

The weak results given in Tables 4.3-4.5 made us decide to combine all the features of GLCM, GLRLM, and GLSZM into a single feature vector to check if that will affect the classification process and improve the results. The results of the combined features are listed in Table 4.6. The mean accuracy has increased slightly to become 83.4% as we can see. The highest accuracy is obtained by RF of 85.52%, while XGBoost obtained an insignificantly better sensitivity result than RF of 85.78%.

Table 4.6. Results of GLCM + GLRLM + GLSZM combination features (%)

Metrics	**Accuracy**	**Precision**	**Sensitivity**	**F1-Score**	**Time (s)**
SVM	82.85	84.04	83.79	83.41	0.003
RF	85.52	86.22	85.77	86	0.060
k-NN	80.9	84	81.1	80.2	0.013
XGBoost	84.32	85.14	85.78	85.45	0.003

Although we have eliminated the results of GLDM, we replaced the GLCM with it to investigate how the performance will be affected. Table 4.7 shows the results of the combination of GLDM, GLRLM, and GLSZM features. From the table, we can observe that the performance of the models decreased a little except for k-NN which increased. The mean accuracy decreased to 81.9%, while the highest accuracy is obtained by RF of 83.24%. The mean values of all the other metrics have decreased as well. We have obtained the confusion matrix for this combination with the RF classifier in Figure 4.3 to be compared with one in Figure 4.2. From this confusion matrix, we can tell that RF forest has misclassified 10 positive samples as negative, while 4 negative samples as positive.

Table 4.7. Results of GLDM+GLRLM+GLSZM combination features (%)

Metrics	**Accuracy**	**Precision**	**Sensitivity**	**F1-Score**	**Time (s)**
SVM	78.95	75.88	89.66	81.54	0.002
RF	83.24	82.82	86.57	85	0.055
k-NN	82.98	84.48	81.8	83.54	0.012
XGBoost	82.44	84.72	83.48	84.47	0.003

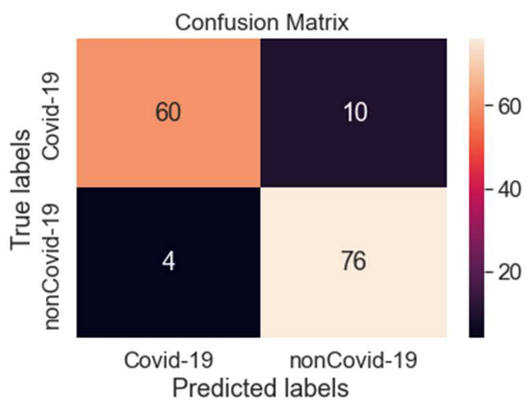


Figure 4.3. Confusion matrix of GLDM+GLRLM+GLSZM combination with RF

Another combination included GLSZM, NGTDM, and GLDM. This time we replaced GLRLM with NGTDM. The results in Table 4.8 show that the performance of SVM and k-NN has decreased, while for RF and XGBoost it increased. The mean value of all accuracies is 79.35%. RF has achieved the highest accuracy score of 83.51%.

Table 4.8. Results of GLSZM+NGTDM+GLDM combination features (%)

Metrics	Accuracy	Precision	Sensitivity	F1-Score	Time (s)
SVM	70.11	72.72	74.34	72.15	0.002
RF	83.51	84.25	85.29	84.64	0.060
k-NN	80.96	81.33	82.87	82.37	0.012
XGBoost	82.83	82.88	85.41	83.51	0.003

The last combination included GLCM, NGTDM, and GLDM. In this combination, we replaced GLSZM with GLCM. Table 4.9 shows the results of this combination. The overall mean values of metrics have decreased for all the models except SVM. The mean also decreased to 78.57%. The highest accuracy result is achieved by RF with an accuracy of 80.82%.

Table 4.9. Results of GLCM+NGTDM+GLDM combination features (%)

Metrics	Accuracy	Precision	Sensitivity	F1-Score	Time (s)
SVM	74.78	76.87	79.98	79.97	0.002
RF	80.82	81	83.6	83	0.055
k-NN	78.28	78.78	79.4	78.7	0.018
XGBoost	80.41	82.57	78.42	81.73	0.003

4.2. Deep Learning Results

Several transfer learning models such as VGG-19, ResNet-50, DenseNet210, Xception, and EfficientNet-B0 were utilized to classify COVID-19 from CT images. The datasets used to train and test the models were “Large COVID-19 CT scan slice dataset” and “CT Scans for COVID-19 Classification”. The results were obtained when models were evaluated on the 20% test set of each dataset. The AUC metrics were included as well as an evaluation metric for the DL models.

4.2.1. VGG-19 Results

The VGG-19 model was loaded with ImageNet weights and input shape of (224x224x3). Fine-tuning the model was implemented by removing the fully connected layers and adding new two layers; one with 256 neurons and ReLU activation function, one as output with three neurons, and SoftMax activation function. After resizing and normalizing the images of the datasets they were split into 80% training set and 20% test set. The training set was again split into 80% training and 20% validation. The model was trained and validated for 50 epochs with an Adam optimizer. After the model finished the training for 50 epochs, it was evaluated on the 20% test set. The process of training and evaluating the model was executed three times separately for each dataset. The performance metrics of the model are shown in Table 4.10.

Table 4.10. Results of the VGG-19 model on both datasets (%)

Dataset \ Metrics	Accuracy	Precision	Sensitivity	F1-score	AUC
Large COVID-19 CT scan slice dataset	98.04	98.06	98.04	98.04	99.79
CT Scans for COVID-19 Classification - original version	99.54	99.54	99.54	99.54	99.95
CT Scans for COVID-19 Classification – pre-processed version	99.21	99.21	99.21	99.21	99.96

From the results in the table, we can see that VGG-19 has scored an accuracy of 98.04% on the first dataset and better results of 99.54% and 99.21% on the second dataset for both versions respectively. The results show that pre-processing the second dataset didn't improve the model classification performance, but rather decreased it slightly. However, the highest AUC score was achieved on the pre-processed version dataset. The confusion matrices of the VGG-19 model's results are shown in Figure 4.4. The number of test samples for the first dataset is 3421 samples, 1519 of which belong to the COVID-19 class. 45 samples of the COVID-19 class were misclassified as a normal class, while 5 of them were predicted as CAP class. On the other hand, the matrices of the second dataset show that there are 3937 test samples for both versions, 800 of which are for positive class. The model classified 796 of them as a true class and 4 were misclassified for both versions as shown in Figure 4.4. The graphs in Figure 4.5-Figure 4.7 show the improvement of the training and the validation accuracy in every epoch along with the loss of training and validation in every epoch for the two datasets. The model reached the peak performance after 20 epochs and settled for the rest of the epochs.

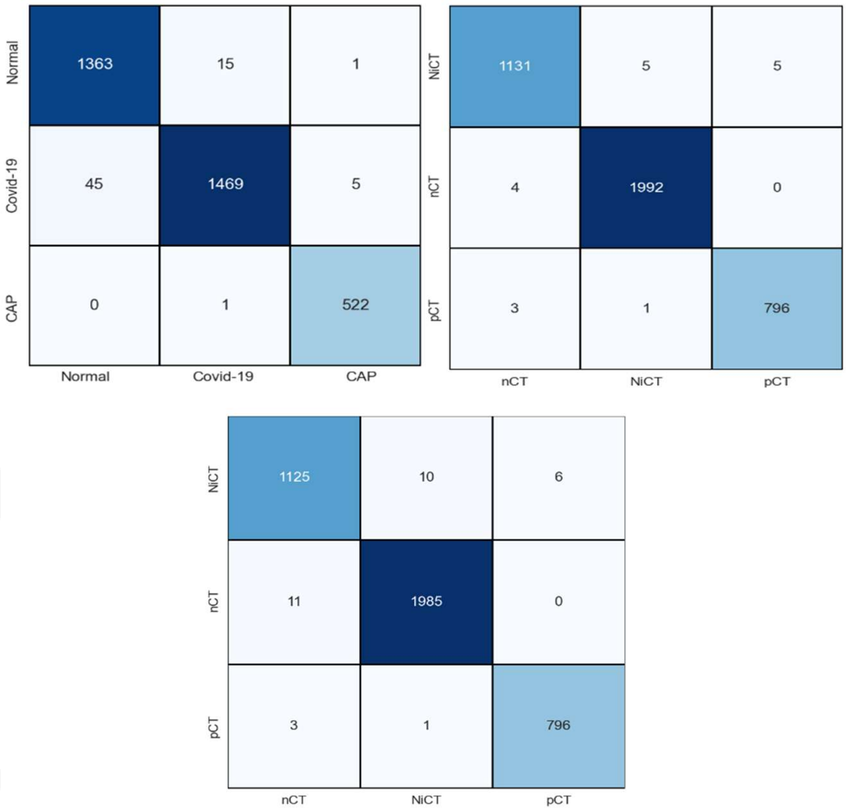


Figure 4.4. Confusion matrices of the VGG-19 model on both datasets

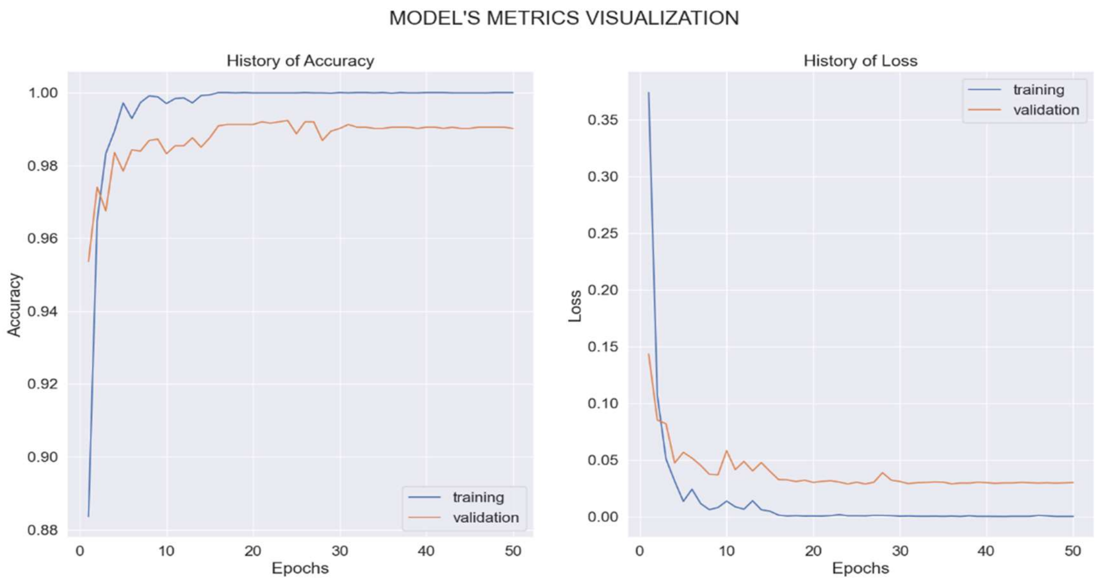


Figure 4.5. History graphs of accuracy and loss values of training and validation through 50 epochs for the VGG-19 model on Large COVID-19 CT scan slice dataset

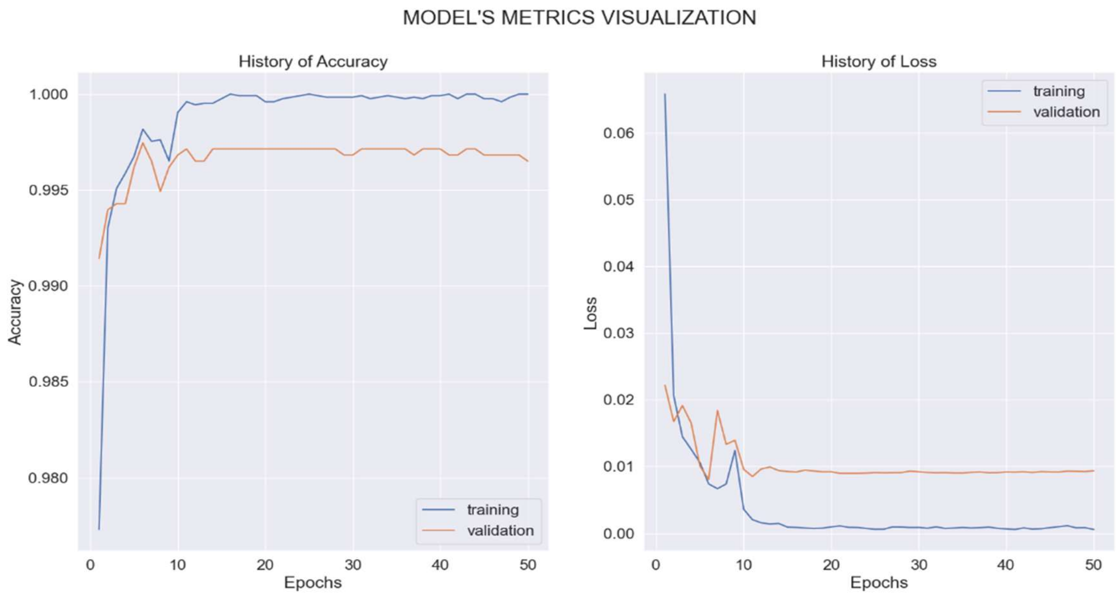


Figure 4.6. History graphs of accuracy and loss values of training and validation through 50 epochs for VGG-19 model on CT Scans for COVID-19 Classification - original version dataset

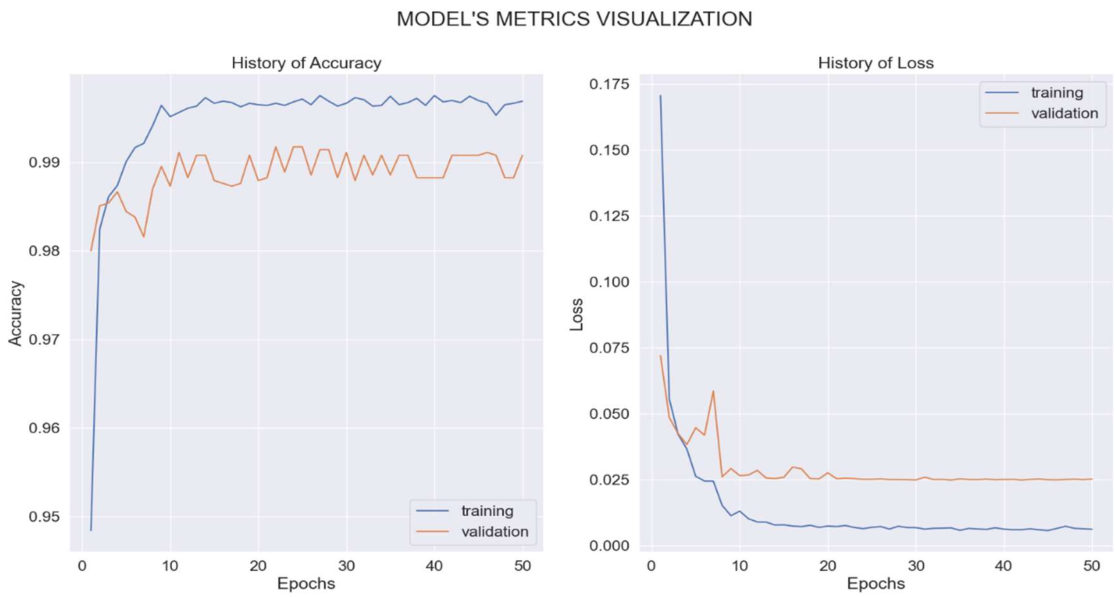


Figure 4.7. History graphs of accuracy and loss values of training and validation through 50 epochs for VGG-19 model on CT Scans for COVID-19 Classification - pre-processed version dataset

4.2.2. Resnet-50 Results

The ResNet-50 model was also loaded with ImageNet weights and input shape of (224x224x3). We Fine-tuned the model by removing the fully connected layers and adding new two layers; one with 256 neurons and ReLU activation function, and the other one is the output with three neurons and SoftMax activation function. Images in the datasets were resized and normalized before they were split into 80% training set and 20% test set. A 20% of the training set was held for model validation after each epoch. The model was trained and validated for 50 epochs with an Adam optimizer. After the model finished the training process, it was evaluated on the 20% test set. Training and evaluating the model were executed three times for each dataset separately. Table 4.11 shows the performance metrics of the model for each dataset.

Table 4.11. Results of the ResNet-50 model on both datasets (%)

Dataset \ Metrics	Accuracy	Precision	Sensitivity	F1-score	AUC
Large COVID-19 CT scan slice dataset	97.84	97.86	97.84	97.84	99.78
CT Scans for COVID-19 Classification - original version	99.62	99.62	99.62	99.62	99.99
CT Scans for COVID-19 Classification – pre-processed version	98.55	98.55	98.55	98.55	99.78

The results in the table, show that ResNet-50 has scored an accuracy of 97.84% on the first dataset and better results than VGG-19 of 99.62% and 99.55% on the second dataset for both versions respectively. The results also ensure that pre-processing didn't improve the model classification performance, but rather decreased it slightly. Additionally, the highest AUC score was achieved on the original dataset this time with a score of 99.99%. The confusion matrices of the ResNet-50 model's results are shown in Figure 4.8. The number of test samples for the first dataset is 3421 samples, 1519 of which belong to the COVID-19 class. This time, the number of the misclassified COVID-19 class was 76 samples, while only 1 was predicted as a CAP class. From the other confusion matrices, it's seen that there are 3937 test samples for versions, 800 of which are for positive class. The model classified 794 of them as a true class and 6 were misclassified for the original version, while classified 783 as true and misclassified 17 on the pre-processed version as shown in Figure 4.8. The graphs in Figure 4.9-Figure 4.11 show the improvement of the training and the validation accuracy in every epoch along with the loss of training and validation in every epoch for the two datasets. The model reached the peak performance after 30 epochs and settled for the rest of the epochs.

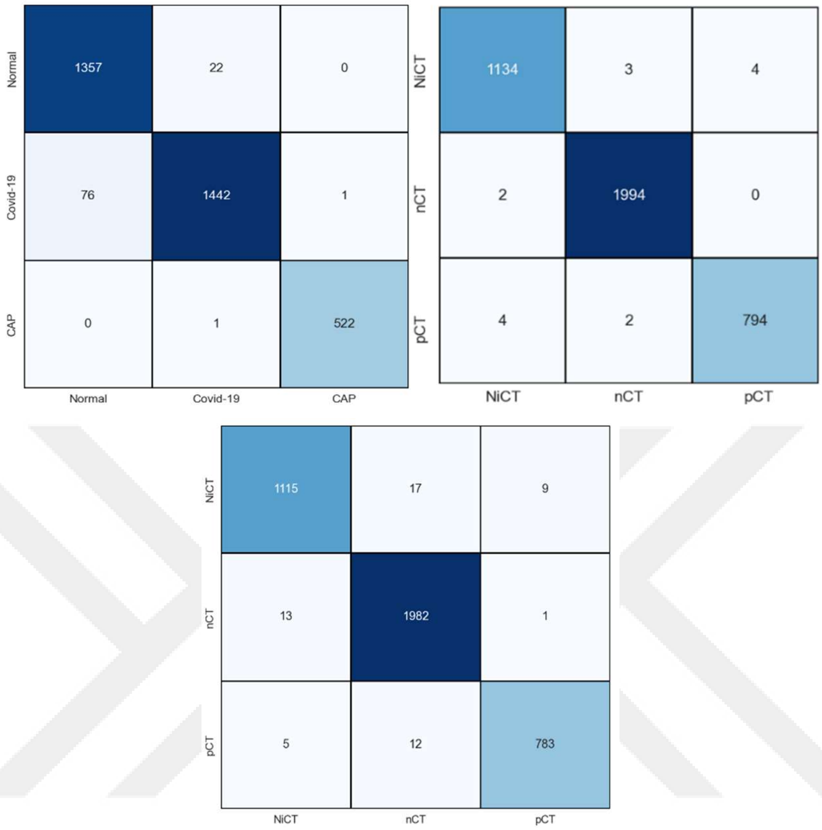


Figure 4.8. Confusion matrices of ResNet-50 on both datasets

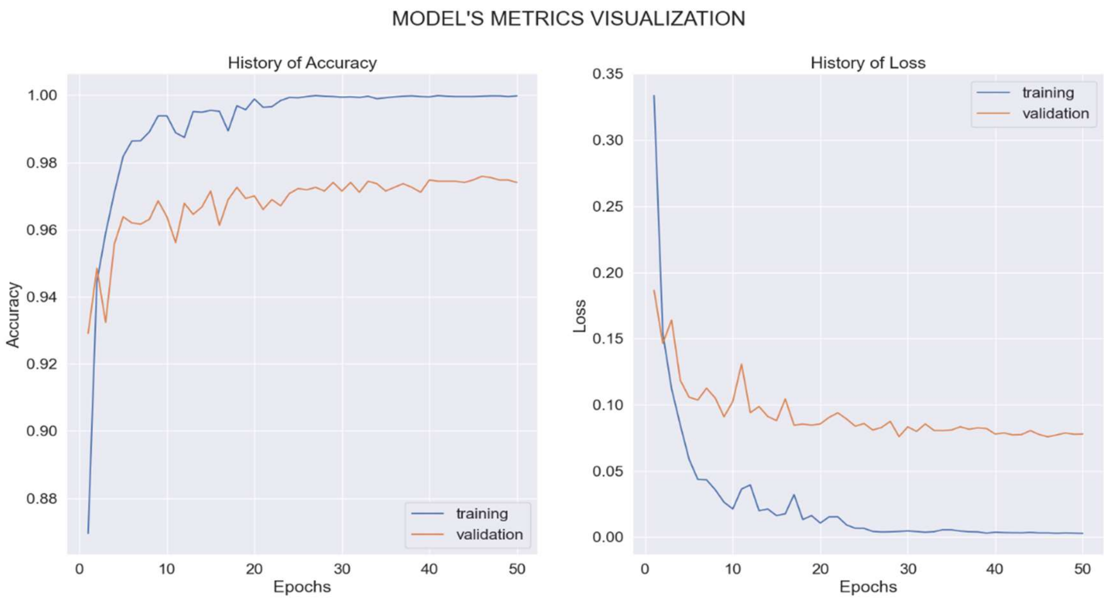


Figure 4.9. History graphs of accuracy and loss values of training and validation through 50 epochs for ResNet-50 model on Large COVID-19 CT scan slice dataset

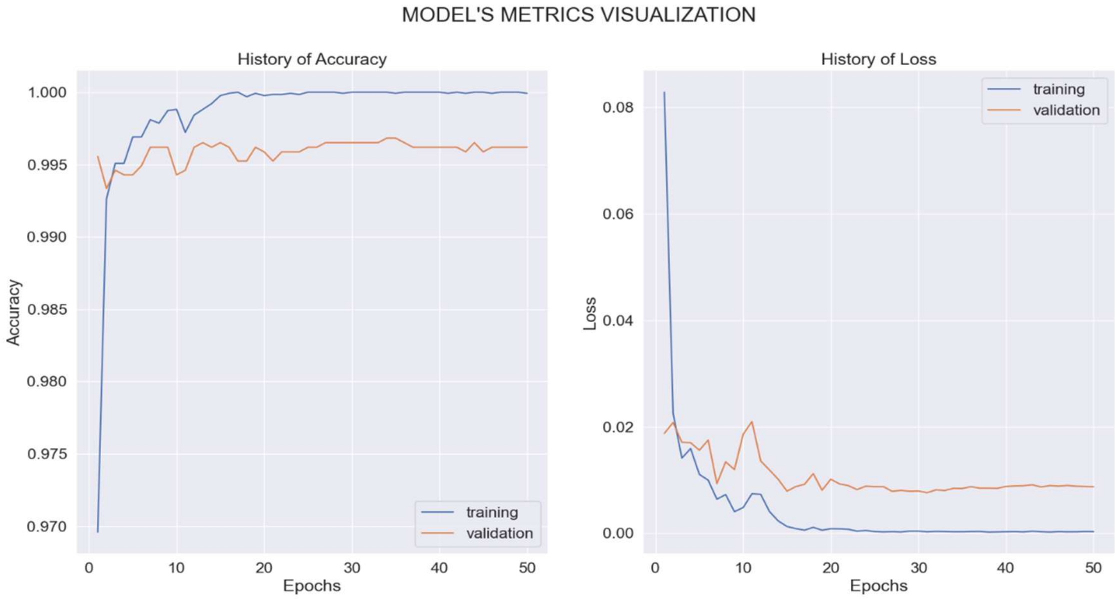


Figure 4.10. History graphs of accuracy and loss values of training and validation through 50 epochs for ResNet-50 model on CT Scans for COVID-19 Classification - pre-processed version dataset

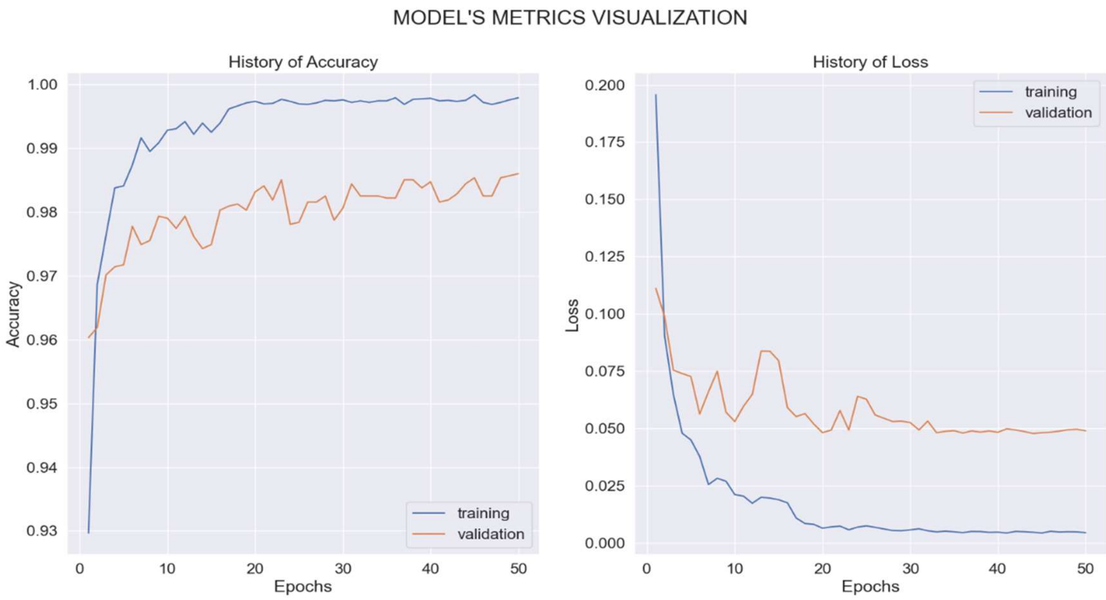


Figure 4.11. History graphs of accuracy and loss values of training and validation through 50 epochs for ResNet-50 model on CT Scans for COVID-19 Classification - original version dataset

4.2.3. Densenet-201 Results

DenseNet-201 model accepts an input shape of (224x224x3) and it was loaded with ImageNet weights. Fine-tuning the model by removing the fully connected layers and adding new two layers was carried out. The first added layer has 256 neurons and is equipped with a ReLU activation function, the other layer is the output layer with three neurons and a SoftMax activation function. We have resized the images in the datasets and normalized them before training the model. The datasets were split into 80% training set and 20% test set. A 20% of the training set was utilized for the validation. Adam optimizer was used with the model in the training process for 50 epochs with Adam. After the model finished the training process, it was evaluated on the 20% test set. Training and evaluating the model were executed three times for each dataset separately. The performance metrics of the model are shown in Table 4.12 for each dataset.

Table 4.12. Results of the DenseNet-201 model on both datasets (%)

Dataset \ Metrics	Accuracy	Precision	Sensitivity	F1-score	AUC
Large COVID-19 CT scan slice dataset	98.01	98.04	98.01	98.01	99.85
CT Scans for COVID-19 Classification - original version	99.42	99.42	99.42	99.42	99.99
CT Scans for COVID-19 Classification – pre-processed version	99.16	99.16	99.16	99.16	99.99

The scores in the table, show that DenseNet-201 has scored a better accuracy of 97.84% than ResNet-50 and slightly lower than VGG-19 on the first dataset. For the second dataset, it archived lower results than VGG-19 and ResNet-50 of 99.42% on the original version, while a better accuracy than ResNet-50 and slightly lower than VGG-19 of 99.16% on the pre-processed version. Additionally, it obtained the highest AUC score for now with a score of 99.99% on the second dataset and 99.85% higher than VGG-19 on the first dataset. The confusion matrices shown in Figure 4.12 are for the DenseNet-201 model's results. The model has misclassified 53 test samples of the COVID-19 class as normal and 1 as CAP. For the other confusion matrices, the model classified 795 test samples of the positive class correctly and 5 were misclassified for the original version, while classified 796 correctly and misclassified only 4 on the pre-processed version as shown in Figure 4.12.

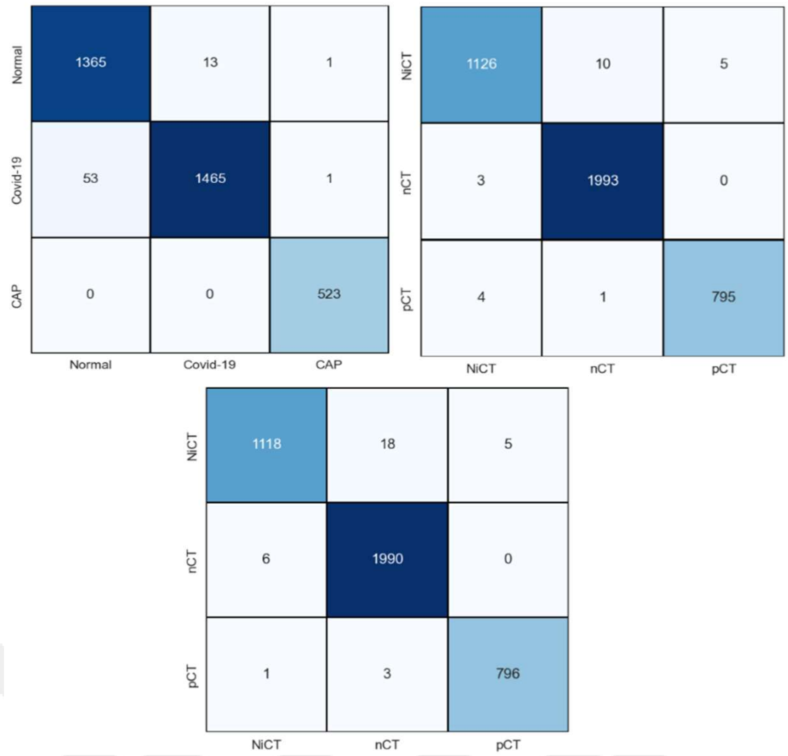


Figure 4.12. Confusion matrices of DenseNet-201 on both datasets

The model reached the peak performance after 30 epochs and settled for the rest of the epochs except for the pre-processed version in which the validation accuracy has remained fluctuating throughout the epochs.

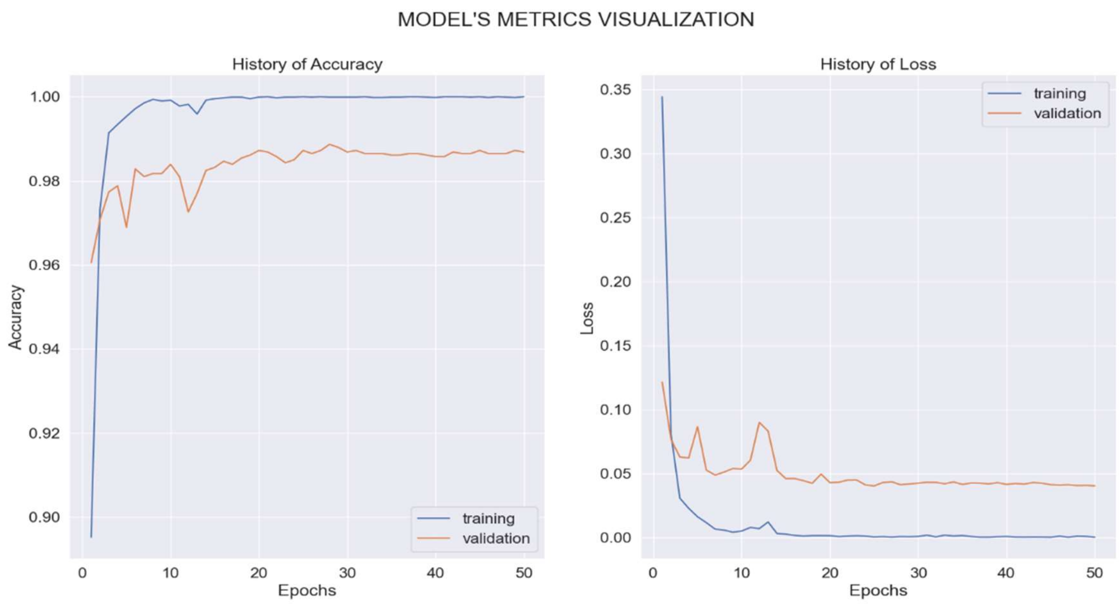


Figure 4.13. History graphs of accuracy and loss values of training and validation through 50 epochs for DenseNet-201 model on Large COVID-19 CT scan slice dataset

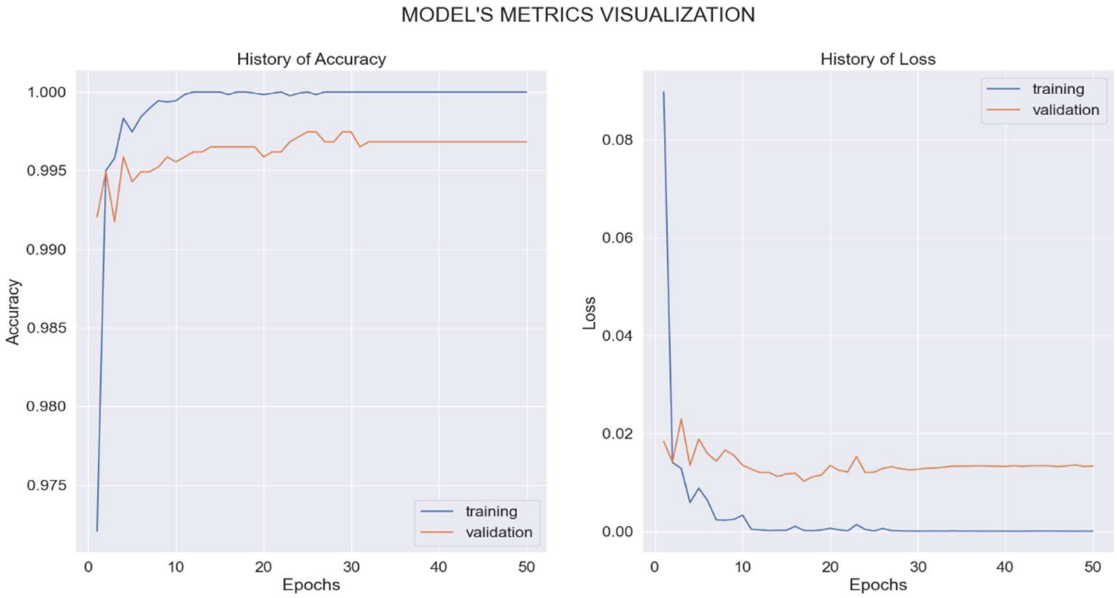


Figure 4.14. History graphs of accuracy and loss values of training and validation through 50 epochs for DenseNet-201 model on CT Scans for COVID-19 Classification - original version dataset

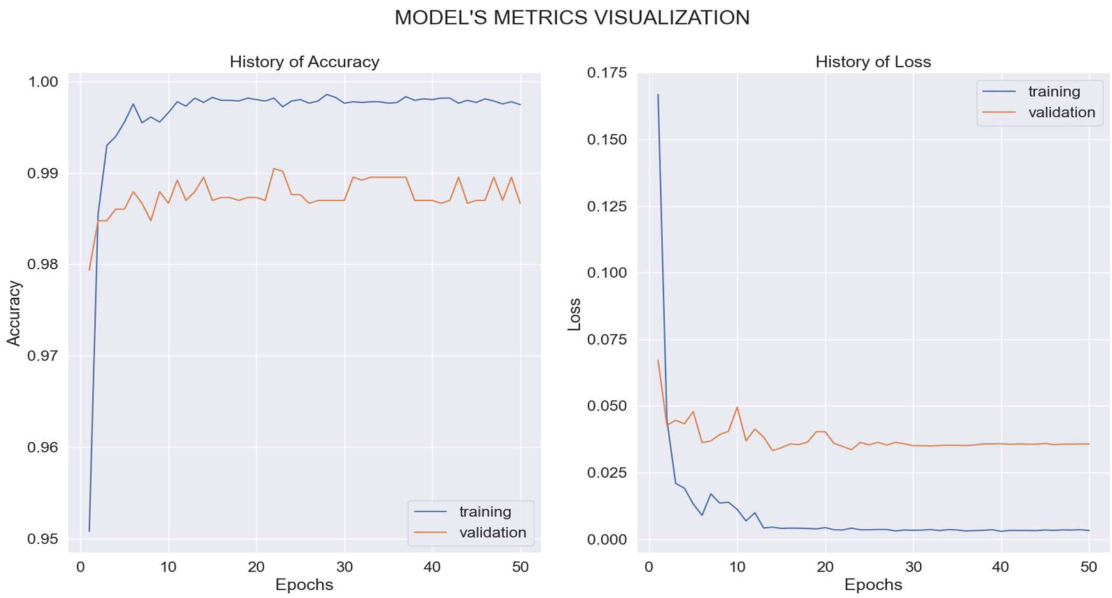


Figure 4.15. History graphs of accuracy and loss values of training and validation through 50 epochs for DenseNet-201 model on CT Scans for COVID-19 Classification - pre-processed version dataset

4.2.4. Xception Results

Xception model accepts an input shape of (299x299x3) as default because it was trained with that shape, however, it can be changed to (224x224x3), and that is what we did. The ImageNet weights were used with the model. The fine-tuning process was carried out as well. We added one layer with 256 neurons and equipped with ReLU activation function and another layer as an output layer with three neurons and SoftMax activation function. We have also resized the images and normalized them before training the model. The datasets were split into 80% training set and 20% test set. A 20% of the training set was held for model validation after each epoch. The model was trained and validated for 50 epochs with an Adam optimizer. After the training process, the model was evaluated on the 20% test set. Training and evaluating the model were applied three times for each dataset. The performance metrics of the model are shown in Table 4.13 for each dataset.

Table 4.13. Results of the Xception model on both datasets (%)

Metrics / Dataset	Accuracy	Precision	Sensitivity	F1-score	AUC
Large COVID-19 CT scan slice dataset	95.35	95.4	95.35	95.35	99.38
CT Scans for COVID-19 Classification - original version	99.39	99.39	99.39	99.39	99.96
CT Scans for COVID-19 Classification – pre-processed version	97.71	97.71	97.71	97.71	99.63

The results show that Xception has scored the lowest accuracies of 95.35%, 99.39%, and 97.71% for each dataset respectively. The difference between the performance of the model on the original and the pre-processed versions can be noticed as well. Additionally, the model obtained the lowest AUC scores compared to the previous models except for the original version of the second dataset where it achieved a little higher score than VGG-19 of 99.96% but still lower than the ResNet-50 and DenseNet-201 models. Figure 4.16 shows the confusion matrices for the Xception model's results. Out of 1519 samples of COVID-19 class, the model has misclassified 99 of them as normal and 3 as CAP. For the other matrices, the model classified 795 test samples of the positive class correctly and 5 were misclassified for the original version, while classified 771 correctly and misclassified 29 on the pre-processed version as shown in Figure 4.16. The graphs in Figure 4.17-Figure 4.19 show the improvement of the training and the validation accuracy in every epoch along with the loss of training and validation in every epoch for the two datasets. The model reached the peak performance after 30 epochs and settled for the rest of the epochs.

Figure 4.16. Confusion matrices of DenseNet-201 on both datasets

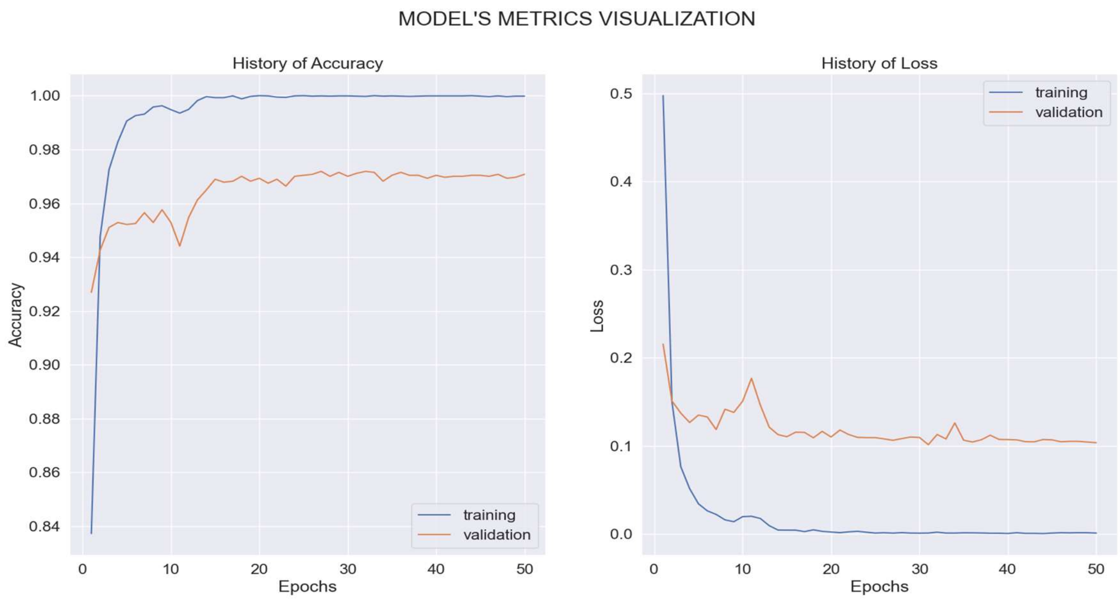


Figure 4.17. History graphs of accuracy and loss values of training and validation through 50 epochs for Xception model on Large COVID-19 CT scan slice dataset

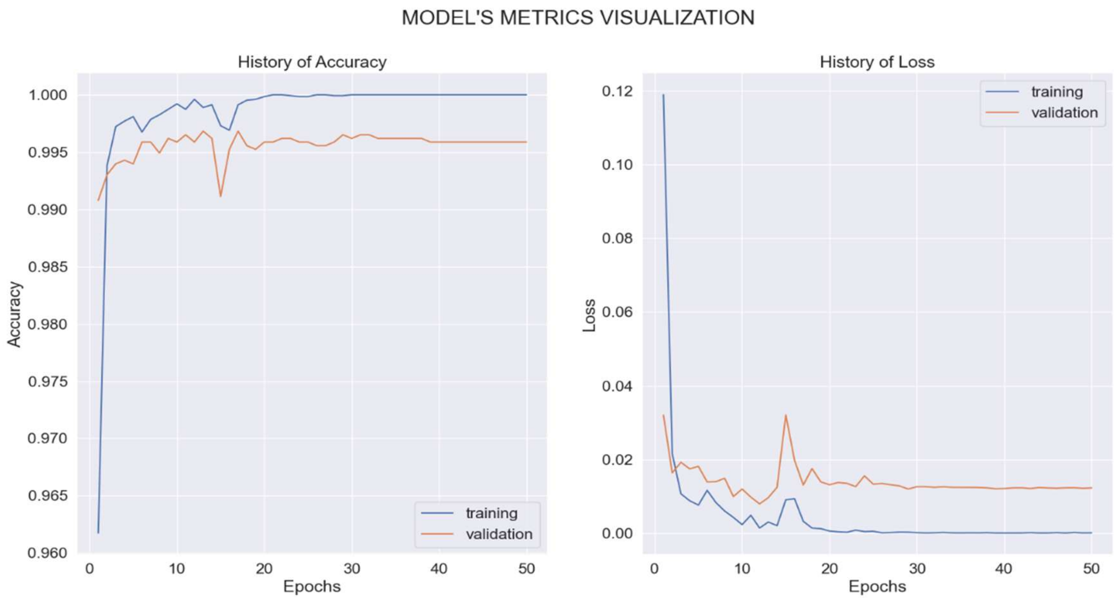


Figure 4.18. History graphs of accuracy and loss values of training and validation through 50 epochs for Xception model on CT Scans for COVID-19 Classification - original version dataset

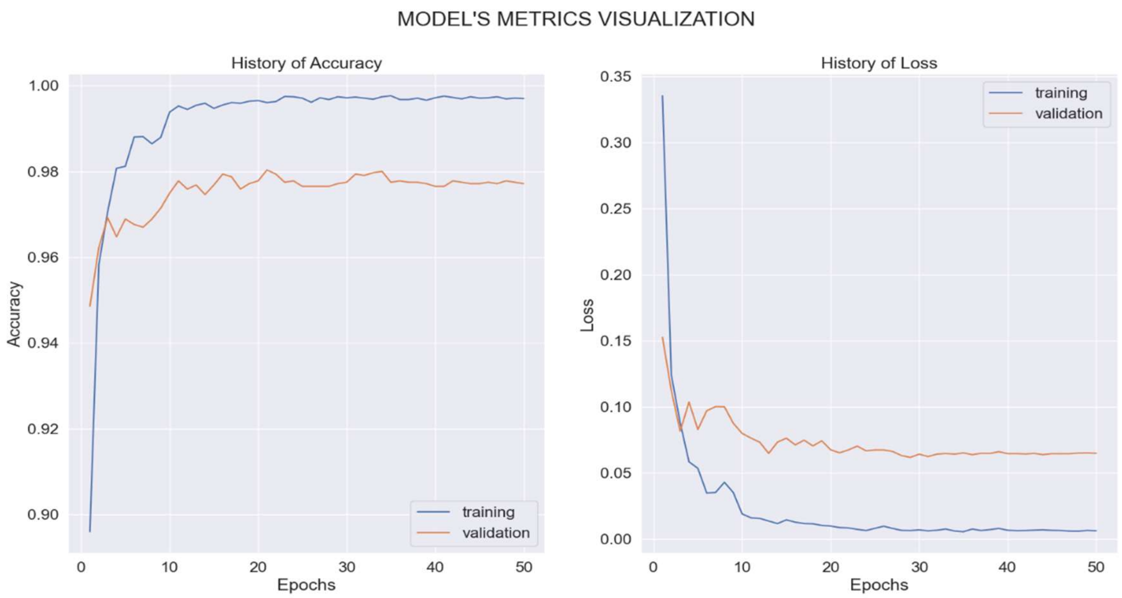


Figure 4.19. History graphs of accuracy and loss values of training and validation through 50 epochs for Xception model on CT Scans for COVID-19 Classification - pre-processed version dataset

4.2.5. Efficientnet-B0 Results

EfficientNet-B0 is the newest and the last model we have used in this experiment. The model has an input shape of (224x224x3) and was loaded with ImageNet weights. Fine-tuning process was carried to this model as well. One layer with 256 neurons and ReLU activation function were added along with an output layer with three neurons and SoftMax activation function. The images were resized and normalized before training the model. Then, the datasets were split into 80% training set and 20% test set. A 20% of the training set was assigned for model validation after each epoch. The model was trained and validated for 50 epochs with an Adam optimizer. After the training process, the model was evaluated on the 20% test set. Training and evaluating the model were executed three times for each dataset. The performance metrics of the model are shown in Table 4.14 for each dataset.

Table 4.14. Results of the EfficientNet-B0 model on both datasets (%)

Metrics / Dataset	Accuracy	Precision	Sensitivity	F1-score	AUC
Large COVID-19 CT scan slice dataset	97.84	97.86	97.84	97.84	99.82
CT Scans for COVID-19 Classification - original version	99.57	99.57	99.57	99.57	99.94
CT Scans for COVID-19 Classification – pre-processed version	98.93	98.94	98.93	98.93	99.91

The results show that EfficientNet-B0 has scored the average accuracies of 97.84%, 99.57%, and 98.93% for each dataset respectively. The model achieved better scores than Xception and similar to ResNet-50, but lower than VGG-19 and DenseNet-201 for the first dataset. In the original version of the second dataset, it obtained better results than VGG-19, DenseNet-201, and Xception but slightly lower than ResNet-50. Figure 4.20 shows the confusion matrices for the EfficientNet-B0 model's results. Out of 1519 samples of COVID-19 class the model has classified 1469 correctly as VGG-19 and misclassified 50; 49 as normal and only 1 as CAP. For the other matrices, the model classified 795 test samples of the positive class correctly and 5 were misclassified for the original version, while classified 791 correctly and misclassified 9 for the pre-processed version as shown in Figure 4.20. The graphs in Figure 4.21-Figure 4.23 show the improvement of the training and the validation accuracy in every epoch along with the loss of training and validation in every epoch for the two datasets. The model reached the peak performance after 30 epochs and settled for the rest of the epochs.

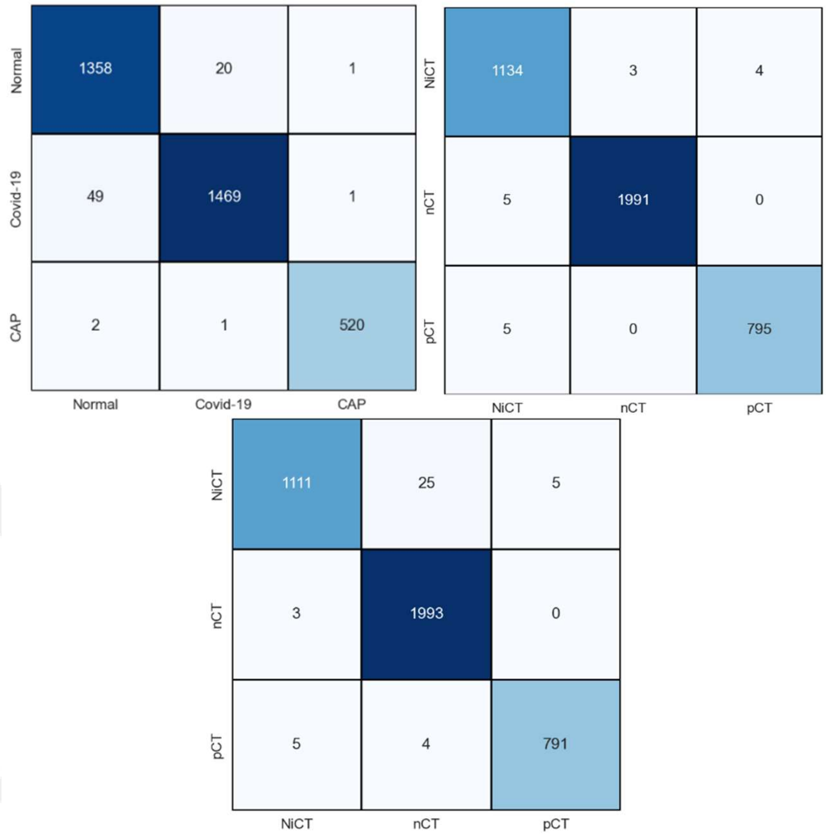


Figure 4.20. Confusion matrices of EfficientNet-B0 on both datasets

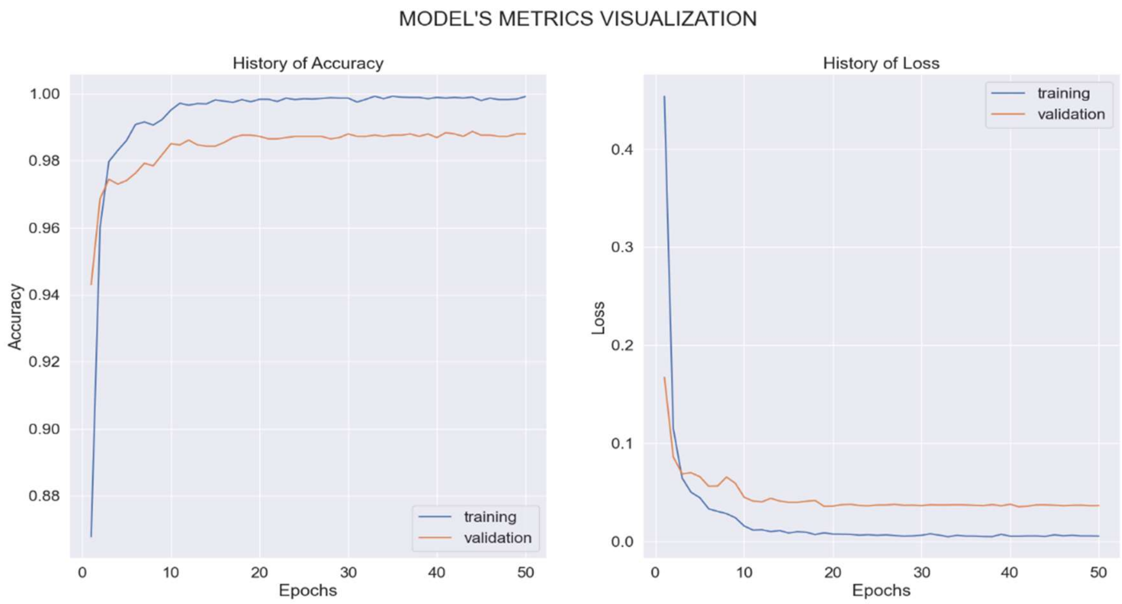


Figure 4.21. History graphs of accuracy and loss values of training and validation through 50 epochs for EfficientNet-B0 model on Large COVID-19 CT scan slice dataset

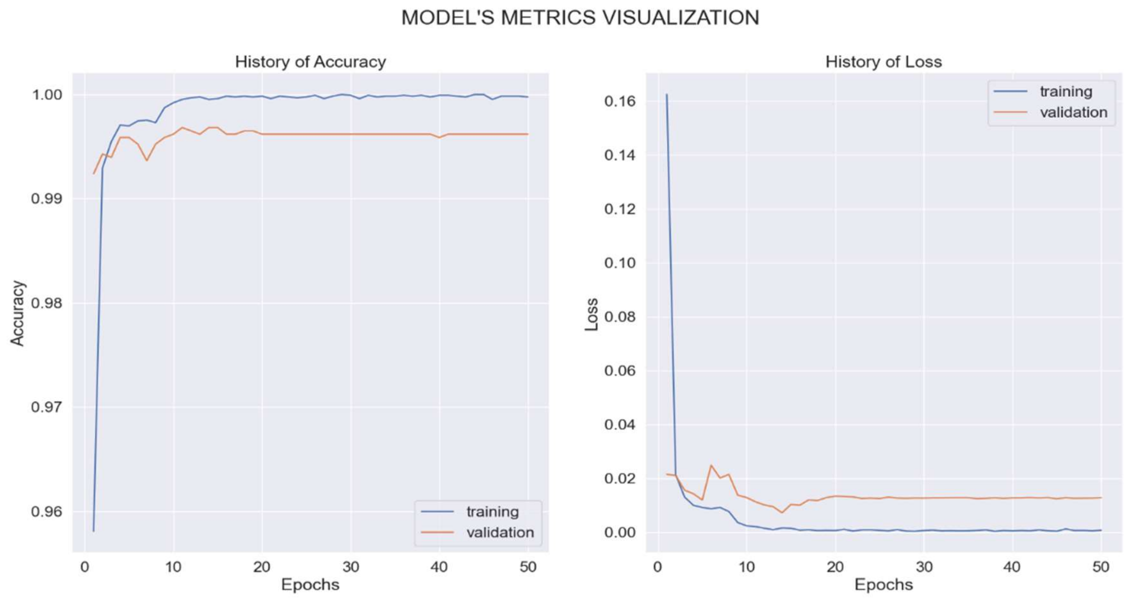


Figure 4.22. History graphs of accuracy and loss values of training and validation through 50 epochs for EfficientNet-B0 model on CT Scans for COVID-19 Classification - original version dataset

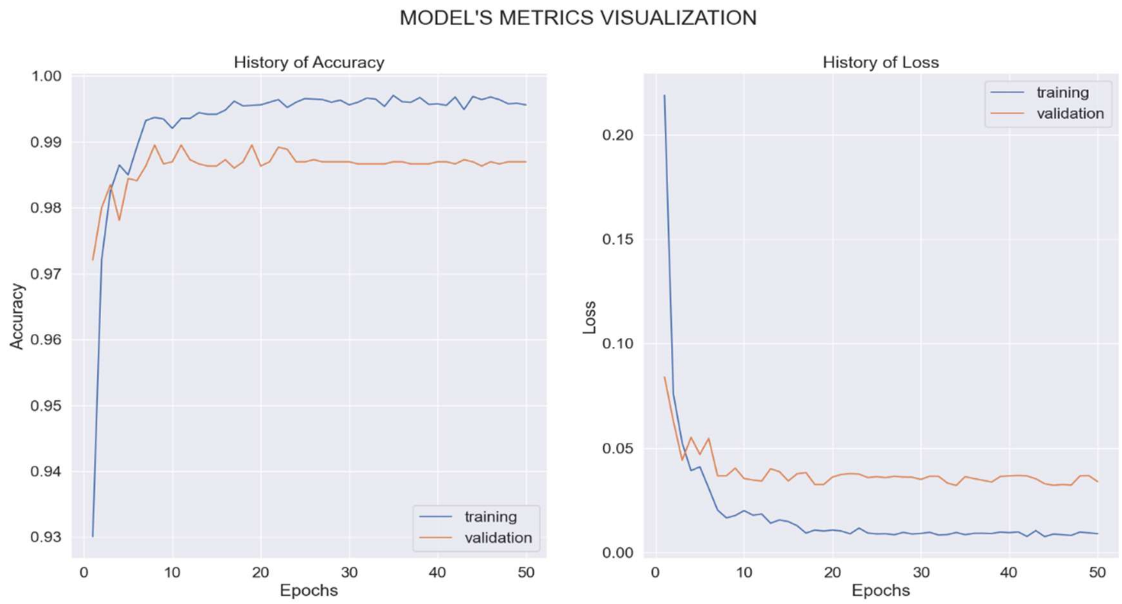


Figure 4.23. History graphs of accuracy and loss values of training and validation through 50 epochs for EfficientNet-B0 model on CT Scans for COVID-19 Classification – pre-processed version dataset

4.2.6. Comparing Deep Learning Models

All the results of the pre-trained models shown in Table 4.10 to Table 4.14 are rearranged according to the dataset and combined in one table for comparison. These results are listed in Table 4.15.

Table 4.15. Results of all the models for both datasets (%)

Dataset	Model	Accuracy	Precision	Sensitivity	F1-score	AUC
Large COVID-19 CT scan slice dataset	VGG-19	**98.04**	98.06	98.04	98.04	99.79
	ResNet-50	97.84	97.86	97.84	97.84	99.78
	DenseNet-201	98.01	98.04	98.01	98.01	99.85
	Xception	95.35	95.4	95.35	95.35	99.38
	EfficientNet-B0	97.84	97.86	97.84	97.84	99.82
CT Scans for COVID-19 Classification original version	VGG-19	99.54	99.54	99.54	99.54	99.95
	ResNet-50	**99.62**	99.62	99.62	99.62	99.99
	DenseNet-201	99.42	99.42	99.42	99.42	99.99
	Xception	99.39	99.39	99.39	99.39	99.96
	EfficientNet-B0	99.57	99.57	99.57	99.57	99.94
CT Scans for COVID-19 Classification pre-processed version	VGG-19	**99.21**	99.21	99.21	99.21	99.96
	ResNet-50	98.55	98.55	98.55	98.55	99.78
	DenseNet-201	99.16	99.16	99.16	99.16	99.99
	Xception	97.71	97.71	97.71	97.71	99.63
	EfficientNet-B0	98.93	98.94	98.93	98.93	99.91

For the “Large COVID-19 CT scan slice dataset” dataset the best accuracy and sensitivity were achieved by VGG-19 of 98.04% for both and the mean accuracy is 97.41%. DenseNet-201 has achieved the highest AUC score of 99.85%. For the original version of the second dataset, ResNet-50 has achieved the highest accuracy of 99.62% followed by EfficientNet201 with an accuracy of 99.57%. The mean accuracy is 99.5%, while the best AUC score is 99.99% obtained by both DenseNet-201 and ResNet-50. In contrast, VGG-19 has achieved the best accuracy of 99.21% on the pre-processed version of the second dataset. The mean accuracy on this version is 98.71% which is lower than the original's mean value. This means that the segmentation methods do not always improve the classification process. Despite the fact that the VGG-19 model is the oldest model and the simplicity of its architecture, it achieved the best results among the rest of the models, followed by ResNet-50.

4.3. Comparing The Results with The Literature

The best results we obtained in both ML and DL stages are listed in Table 4.16.

Table 4.16. The best results obtained in this study

Dataset	Model	Best accuracy
COVID-CT	XGBoost	98.65%
Large COVID-19 CT scan slice dataset	VGG-19	98.04%
CT Scans for COVID-19 Classification original version	Resnet-50	99.62%
CT Scans for COVID-19 Classification pre-processed version	VGG-19	99.21%

For ML, XGBoost has achieved the best accuracy in FOS features extracted from the images of the COVID-CT dataset. For SVM, RF, KNN, and XGBoost, the training phases took about 1.5s, 17ms, 6ms, and 0.25s, respectively, while the test time for each image in the test set is less than 1ms. The following are the time-related findings of the studies that specified the time: (Dey et al., 2020) takes an average of 173 ±11s, (Zheng et al., 2020) takes an average of 1.93s, and (Abd Elaziz et al., 2021) takes an average of 3.123s. For DL pre-trained models, the VGG-19 and ResNet-50 models achieved the best accuracy scores.

To compare our best results with the literature, we have created Table 4.17. The table shows the best results of studies in the literature with the dataset they used and the model that achieved those results. Some of the studies in the literature used privet datasets while some used the same public datasets we used and their results have been bolded in the table. The table shows that this thesis obtained successful and competitive results with the literature.

Table 4.17. Comparison of the results of studies in the literature

Authors	Dataset	Pre-processing	Method	Result (%)
(Abbasian Ardakani et al., 2021)	612 CT images (306 COVID-19 and 306 non-COVID-19)	20 Radiological features extraction	DT, k-NN, Naïve Bayes, SVM and Ensemble.	91.94 by Ensemble
(Al-Karawi et al., 2020)	470 CT images (275 positive and 195 negative)	Features extraction (FFT-Gabor)	SVM	95.37
(Barstugan et al., 2020)	150 CT images	Patch regions cropping and feature extraction	SVM	99.64
(Dey et al., 2020)	400 CT images (200 normal and 200 COVID-19)	Segmentation and feature extraction	RF, k-NN, SVM, and DT	87 by k-NN
(Liu et al., 2020)	88 CT images (61 COVID-19 and 27 general pneumonia)	Delineation of ROIs and feature extraction	DT, SVM, LR, k-NN, and Ensemble of bagged tree	94.16 by EBT
(Özkaya et al., 2020)	150 CT images	*Deep feature extraction	SVM	98.27
(Kassania et al., 2021)	274 images (117 X-ray and 20 CT positive and 117 X-ray and 20 CT negative)	*Deep feature extraction	DT, RF, XGBoost, AdaBoost, Bagging, LightGBM	99 by DenseNet121+ Bagging
(Shi et al., 2021)	2,685 images (1658 COVID-19 and 1027 bacterial pneumonia)	Segmentation and radiomics feature extraction	LightGBM-based proposed method	89.4
(Saeedi et al., 2020)	“COVID-CT”	*Deep feature extraction	DenseNet121, ResNet-50 V1&V2, InceptionV3 and MobileNet V1&V2+NuSVM	**90.61**

Table 4.17. (Continued) Comparison of the results of studies in the literature

Authors	**Dataset**	**Pre-processing**	**Method**	**Result (%)**
(Alsharman & Jawarneh, 2020)	“COVID-CT”	*Deep feature extraction	GoogleNet	**82.14**
(Pham, 2020)	“COVID-CT”	*Deep feature extraction	16 pretrained CNNs	**96.20** by DenseNet-201
(Polsinelli et al., 2020)	“COVID-CT”	*Deep feature extraction	SqueezeNet-based model	85.03
(Hernández Santa Cruz, 2021)	“COVID-CT”	*Deep feature extraction	VGG-16, ResNet-50, Wide ResNet-50-2, DenseNet161, DenseNet169, and Inception v3 - ensemble	86.7
(Abd Elaziz et al., 2021)	“COVID-CT”	*Deep feature extraction	MobileNetV3	78.3
(Sakagianni et al., 2020)	“COVID-CT”	*Deep feature extraction	NAS-based model	88.31
(Shaik & Cherukuri, 2022)	“COVID-CT”	*Deep feature extraction	VGG-16, VGG-19, InceptionV3, ResNet-50, ResNet-50V2, InceptionResNetV2, Xception, and MobileNet-ensemble	93.33
(JavadiMoghaddam & Gholamalinejad, 2021)	“CT Scans for COVID-19 Classification”	*Deep feature extraction	Wavelet CNN-4	99.03
(Balaha et al., 2020)	“CT Scans for COVID-19 Classification”	*Deep feature extraction	VGG-19	99.33

Table 4.17. (Continued) Comparison of the results of studies in the literature

Authors	**Dataset**	**Pre-processing**	**Method**	**Result (%)**
(Neha et al., 2021)	"CT Scans for COVID-19 Classification"	*Deep feature extraction	CNN	99.1
(Ahamed et al., 2021)	"Large COVID19 CT slice dataset"	Augmentation and *deep feature extraction	ResNet-50V2	99.01
(Baghdadi et al., 2022)	"Large COVID19 CT slice dataset"	Augmentation and *deep feature extraction	SeresNext50, SeresNext101, SeNet154, MobileNet, MobileNetV2, MobileNetV3Small, and MobileNetV3Large	**98** by SeNet154
(Zheng et al., 2020)	540 CT images (313 COVID-19 & 229 non-COVID-19)	*Deep feature extraction	2D UNet, DeCoVNet	90.8
(Xu et al., 2020)	618 CT images (219 COVID-19, 175 healthy, and 224 (IAVP))	Segmentation and *deep feature extraction	ResNet-18-based model	86.7
(Song et al., 2021)	1990 CT images (777 COVID-19, 708 healthy, and 505 bacterial pneumonia)	*Deep feature extraction	ResNet-50-based model	93
(H. Wang et al., 2020)	775 CT images (520 TVP and 239 COVID-19)	*Deep feature extraction	GoogleNet inception model	89.5
Proposed (ML)	"COVID-CT"	Radiomics features	SVM, RF, k-NN, and XGBoost	**98.65, XGBoost**

*Deep learning-based feature extraction

Table 4.17. (Continued) Comparison of the results of studies in the literature

Authors	**Dataset**	**Pre-processing**	**Method**	**Result (%)**
Proposed (DL)	"Large COVID19 CT slice dataset"	*Deep feature extraction	VGG-19, ResNet-50, DenseNet-201, Xception, and EfficientNet-B0	**98.04 by VGG-19**
	"CT Scans for COVID-19 Classification" original version			**99.62 by ResNet-50**
	"CT Scans for COVID-19 Classification" original version			**99.16 by VGG-19**

*Deep learning-based feature extraction

5. CONCLUSIONS AND RECOMMENDATIONS

Late in 2019, the world was hit with a major health crisis when COVID-19 was discovered. Given the highly contagious nature of COVID-19, the most important step toward combating COVID-19 is; effective screening, early detection, isolation, and prompt treatment of infected persons to limit the spread of the virus. For this purpose, COVID-19 screening via medical images such as x-rays or CT scans is an alternative solution for rapid diagnosis of the disease.

COVID-19 virus classification from CT images is the first goal of this thesis. Another important goal includes testing the impact of different feature extraction methods on the performance of the classification process. Comparing ML and DL methods for COVID-19 classification is the last goal of this thesis. For those reasons, the experiment is carried out in two stages; ML stage and DL stage.

In the ML stage, methods such as SVM, RF, k-NN, and XGBoost are proposed for the classification of COVID-19 lesions from CT images. The dataset used in this stage is a small publicly available dataset namely “COVID-CT” that is also used in the literature. Different radiomics feature extraction methods such as Shape-based, FOS, and Texture features are used to extract the features that describe the infected lesions from the CT images and thus facilitate the classification process of the models. Accuracy, Precision, Sensitivity, and F1-score were used to evaluate the performance of models. The best results were obtained with FOS features, and XGBoost achieved the highest accuracy of 98.35%.

From the obtained results we found that because medical images consist of gray level intensities, intensity-based features such as FOS features provide a useful tool for investigating various aspects of medical images. Along with FOS features, Shape-based features have been shown to give more valuable information than texture-based features. Since the images in the dataset used in this stage were gathered from various papers and sources and were acquired by different CT scan systems, they have dissimilar resolutions, aspect ratios, and brightness. Despite all that, the proposed method achieved a competitive result of 98.65% accuracy which the studies in the literature that used the same dataset could not achieve. Furthermore, other studies that achieved higher results have used public or private data sets that contain consistent and clear images obtained from the same CT system.

In the deep learning stage, COVID-19 classification from CT images was performed with five different transfer learning models; ResNet-50, DenseNet-201, Xception, and EfficientNet-B0. These models were pre-trained on a huge dataset and acquired the knowledge to extract general features. The models are fine-tuned to be used for our task. In this stage, we used two large publicly available datasets which are also used in the literature. The images are resized to the size of the models' input which is (224,224,3) and normalized and then fed into the models for feature extraction and classification. Feature extraction is performed automatically on models' convolutional layers, while classification is performed in the dense layers that we added in the fine-tuning process. The models were trained for 50 epochs and evaluated on 20% of the datasets using Accuracy, Precision, Sensitivity, F1-score, and AUC score. We have also provided the confusion matrices along with performance graphs. The best results are achieved by VGG-19 and ResNet-50 with an accuracy of 98.04% and 99.62% on the first and second datasets respectively.

The obtained results show that transfer learning models can confidently be employed to distinguish between COVID-19 patients and non-Covied-19 patients. The classification results in this thesis are competitive and comparable with the literature, although dense pre-processing techniques like segmentation and augmentation were not utilized. Moreover, the results obtained from the segmented version of the second dataset were lower than the results of the original version and that confirms that segmentation is not always necessary for improving classification performance. Even though the results of the first dataset are lower than the results of the second dataset, they are far more realistic since the first dataset was constructed of images from 7 different datasets, whereas the second dataset was obtained from only two hospitals.

We can conclude from the results that in some cases with few images ML methods can achieve high results enough to compete with DL models and overcome them in some cases (Shwartz-Ziv & Armon, 2022). In addition, DL methods are known to be used with big datasets to obtain excellent results. However, transfer learning has changed this by providing pre-trained models that don’t need to be trained and have the potential to learn from small datasets.

In future studies, a more comprehensive study can be performed by adding more datasets to the study. At the same time, the study can be enriched by adding more images from a local hospital or adding more than three classes for more realistic results. The models proposed in this thesis provide promising results as they outperformed most of the advanced models in the literature. Thus, the proposed models have the potential to assist expert and non-expert radiologists in the diagnosis of COVID-19 by developing an open-source program with a user-friendly interface and can be used as well in areas with limited access to healthcare facilities.

REFERENCES

Abbasian Ardakani, A., Acharya, U. R., Habibollahi, S., & Mohammadi, A. (2021). COVIDiag: a clinical CAD system to diagnose COVID-19 pneumonia based on CT findings. *European Radiology*, *31*(1), 121–130. https://doi.org/10.1007/s00330-020-07087-y

Abd Elaziz, M., Dahou, A., Alsaleh, N. A., Elsheikh, A. H., Saba, A. I., & Ahmadein, M. (2021). Boosting covid-19 image classification using mobilenetv3 and aquila optimizer algorithm. *Entropy*, *23*(11), 1–17. https://doi.org/10.3390/e23111383

Agarwal, V. (2020). *Complete Architectural Details of all EfficientNet Models*. https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142

Ahamed, K. U., Islam, M., Uddin, A., Akhter, A., Paul, B. K., Yousuf, M. A., Uddin, S., Quinn, J. M. W., & Moni, M. A. (2021). A deep learning approach using effective preprocessing techniques to detect COVID-19 from chest CT-scan and X-ray images. *Computers in Biology and Medicine*, *139*(October), 105014. https://doi.org/10.1016/j.compbiomed.2021.105014

Al-Karawi, D., Al-Zaidi, S., Polus, N., & Jassim, S. (2020). Machine Learning Analysis of Chest CT Scan Images as a Complementary Digital Test of Coronavirus (COVID-19) Patients. *MedRxiv*. https://doi.org/10.1101/2020.04.13.20063479

Alsharman, N., & Jawarneh, I. (2020). GoogleNet CNN neural network towards chest CT-coronavirus medical image classification. *Journal of Computer Science*, *16*(5), 620–625. https://doi.org/10.3844/JCSSP.2020.620.625

Baghdadi, N. A., Malki, A., Abdelaliem, S. F., Magdy Balaha, H., Badawy, M., & Elhosseini, M. (2022). An automated diagnosis and classification of COVID-19 from chest CT images using a transfer learning-based convolutional neural network. *Computers in Biology and Medicine*, *144*(March), 105383. https://doi.org/10.1016/j.compbiomed.2022.105383

Balaha, H. M., El-gendy, E. M., & Saafan, M. M. (2020). CovH2SD: A COVID-19 detection approach based on Harris Hawks Optimization and stacked deep learning. *Expert Systems With Applications*, *January*.

Barstugan, M., Ozkaya, U., & Ozturk, S. (2020). Coronavirus (COVID-19) Classification using CT Images by Machine Learning Methods. *International Journal of Imaging Systems and Technology, 10.1002/Ima.22469. Advance Online Publication. Https://Doi.*, *5*, 1–10.

Bronzino, J. D. (2006). *The Biomedical Engineering Handbook, Third Edition - 3 Volume Set*. Taylor & Francis. https://books.google.com.tr/books?id=EnkxswEACAAJ

Bronzino, J. D., & Peterson, D. R. (2018). *The Biomedical Engineering Handbook: Four Volume Set*. CRC Press. https://books.google.com.tr/books?id=RRK2DwAAQBAJ

Burkov, A. (2019). *The Hundred-Page Machine Learning Book*. Andriy Burkov. https://books.google.com.tr/books?id=0jbxwQEACAAJ

Carotti, M., Salaffi, F., Sarzi-Puttini, P., Agostini, A., Borgheresi, A., Minorati, D., Galli, M., Marotto, D., & Giovagnoni, A. (2020). Chest CT features of coronavirus disease 2019 (COVID-19) pneumonia: key points for radiologists. *La Radiologia Medica*, *125*(7), 636. https://doi.org/10.1007/S11547-020-01237-4

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 1800–1807. https://doi.org/10.1109/CVPR.2017.195

Chollet, F., & others. (2015). *Keras*.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Dekhtiar, J., Durupt, A., Bricogne, M., Eynard, B., Rowson, H., & Kiritsis, D. (2018). Deep learning for big data applications in CAD and PLM–Research review, opportunities and case study. *Computers in Industry*, *100*, 227–243.

Dey, N., Rajinikanth, V., Fong, S. J., Kaiser, M. S., & Mahmud, M. (2020). Social Group Optimization–Assisted Kapur's Entropy and Morphological Segmentation for Automated Detection of COVID-19 Infection from Computed Tomography Images. *Cognitive Computation*, *12*(5), 1011–1023. https://doi.org/10.1007/s12559-020-09751-3

ElBagoury, M., Tolba, M. M., Nasser, H. A., Jabbar, A., Elagouz, A. M., Aktham, Y., & Hutchinson, A. (2021). The find of COVID-19 vaccine: Challenges and opportunities. *Journal of Infection and Public Health*, *14*(3), 389–416. https://doi.org/10.1016/J.JIPH.2020.12.025

Eli Bendersky's Website. (2018). *Depthwise separable convolutions for machine learning*. https://eli.thegreenplace.net/2018/depthwise-separable-convolutions-for-machine-learning/#id2

Expert.ai. (n.d.). *What Is the Definition of Machine Learning?* Retrieved May 29, 2022, from https://www.expert.ai/blog/machine-learning-definition/

Flach, P. (2012). Machine learning: the art and science of algorithms that make sense of data. In *Cambridge university press* (Vol. 10, Issue 3). https://pubs.acs.org/doi/10.1021/acscatal.9b04746

General Electric Healthcare. (2014). *Revolution CT User manual*. 702. www.gehealthcare.com

Géron, A. (2019). Hands-on Machine Learning whith Scikit-Learing, Keras and Tensorfow. In *O'Reilly Media, Inc*.

Gillies, R. J., Kinahan, P. E., & Hricak, H. (2016). Radiomics: Images are more than pictures, they are data. *Radiology*, *278*(2), 563–577. https://doi.org/10.1148/radiol.2015151169

Glen, S. (2019). *ROC Curve Explained in One Picture*. https://www.datasciencecentral.com/roc-curve-explained-in-one-picture/

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning* (Vol. 26, Issue 7553). MIT press.

Gwee, S. X. W., Chua, P. E. Y., Wang, M. X., & Pang, J. (2021). Impact of travel ban implementation on COVID-19 spread in Singapore, Taiwan, Hong Kong and South Korea during the early phase of the pandemic: a comparative study. *BMC Infectious Diseases*, *21*(1), 799. https://doi.org/10.1186/s12879-021-06449-1

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hernández Santa Cruz, J. F. (2021). An ensemble approach for multi-stage transfer learning models for COVID-19 detection from chest CT scans. *Intelligence-Based Medicine*, *5*(June 2020), 100027. https://doi.org/10.1016/j.ibmed.2021.100027

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Hui, D. S., Azhar, E. I., Madani, T. A., Ntoumi, F., Kock, R., Dar, O., Ippolito, G., Mchugh, T. D., Memish, Z. A., & Drosten, C. (2020). The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health—The latest 2019 novel coronavirus outbreak in Wuhan, China. *International Journal of Infectious Diseases*, *91*, 264–266.

IBM Cloud Education. (2020). *What are Neural Networks?* https://www.ibm.com/cloud/learn/neural-networks

JavadiMoghaddam, S. M., & Gholamalinejad, H. (2021). A novel deep learning based method for COVID-19 detection from CT image. *Biomedical Signal Processing and Control*, *70*(July), 102987. https://doi.org/10.1016/j.bspc.2021.102987

Jin, C., Chen, W., Cao, Y., Xu, Z., Tan, Z., Zhang, X., Deng, L., Zheng, C., Zhou, J., Shi, H., & Feng, J. (2020). Development and evaluation of an artificial intelligence system for COVID-19 diagnosis. *Nature Communications 2020 11:1*, *11*(1), 1–14. https://doi.org/10.1038/s41467-020-18685-1

Kassania, S. H., Kassanib, P. H., Wesolowskic, M. J., Schneidera, K. A., & Detersa, R. (2021). Automatic Detection of Coronavirus Disease (COVID-19) in X-ray and CT Images: A Machine Learning Based Approach. *Biocybernetics and Biomedical Engineering*, *41*(3), 867–879. https://doi.org/10.1016/j.bbe.2021.05.013

Ketkar, N., & Moolayil, J. (2021). Deep Learning with Python. In *Deep Learning with Python*. https://doi.org/10.1007/978-1-4842-5364-9

Koçak, B., Durmaz, E. Ş., Ateş, E., & Kılıçkesmez, Ö. (2019). Radiomics with artificial intelligence: A practical guide for beginners. *Diagnostic and Interventional Radiology*, *25*(6), 485–495. https://doi.org/10.5152/dir.2019.19321

La Marca, A., Capuzzo, M., Paglia, T., Roli, L., Trenti, T., & Nelson, S. M. (2020). Testing for SARS-CoV-2 (COVID-19): a systematic review and clinical guide to molecular and serological in-vitro diagnostic assays. *Reproductive BioMedicine Online*, *41*(3), 483–499. https://doi.org/10.1016/j.rbmo.2020.06.001

Leonardo, M. M., Carvalho, T. J., Rezende, E., Zucchi, R., & Faria, F. A. (2019). Deep Feature-Based Classifiers for Fruit Fly Identification (Diptera: Tephritidae). *Proceedings - 31st Conference on Graphics, Patterns and Images, SIBGRAPI 2018*, *May 2019*, 41–47. https://doi.org/10.1109/SIBGRAPI.2018.00012

Lih, O. S., Jahmunah, V., San, T. R., Ciaccio, E. J., Yamakawa, T., Tanabe, M., Kobayashi, M., Faust, O., & Acharya, U. R. (2020). Comprehensive electrocardiographic diagnosis based on deep learning. *Artificial Intelligence in Medicine*, *103*, 101789.

Lin, S., Kantor, R., & Clark, E. (2021). Coronavirus Disease 2019. In *Clinics in Geriatric Medicine* (Vol. 37, Issue 4). https://doi.org/10.1016/j.cger.2021.05.001

Liu, C., Wang, X., Liu, C., Sun, Q., & Peng, W. (2020). Differentiating novel coronavirus pneumonia from general pneumonia based on machine learning. *BioMedical Engineering Online*, *19*(1), 1–14. https://doi.org/10.1186/s12938-020-00809-9

Maftouni, M., Law, A. C. C., Shen, B., Zhou, Y., Ayoobi Yazdi, N., & Kong, Z. (2021). *A Robust Ensemble-Deep Learning Model for COVID-19 Diagnosis based on an Integrated CT Scan Images Database*.

Mathworks. (n.d.). *Image Filtering* . Retrieved May 29, 2022, from https://www.mathworks.com/help/images/linear-filtering.html

Mayerhoefer, M. E., Materka, A., Langs, G., Häggström, I., Szczypiński, P., Gibbs, P., & Cook, G. (2020). Introduction to radiomics. *Journal of Nuclear Medicine*, *61*(4), 488–495. https://doi.org/10.2967/JNUMED.118.222893

Neha, K., Joshi, K. P., Jyothi, N. A., & Kumar, J. V. (2021). Preliminary detection of covid-19 using deep learning and machine learning techniques on radiological data. *Indian Journal of Computer Science and Engineering*, *12*(1), 79–88. https://doi.org/10.21817/indjcse/2021/v12i1/211201064

Ning, W., Lei, S., Yang, J., Cao, Y., Jiang, P., Yang, Q., Zhang, J., Wang, X., Chen, F., Geng, Z., Xiong, L., Zhou, H., Guo, Y., Zeng, Y., Shi, H., Wang, L., Xue, Y., & Wang, Z. (2020). *iCTCF: an integrative resource of chest computed tomography images and clinical features of patients with COVID-19 pneumonia*. Research Square. https://doi.org/10.21203/rs.3.rs-21834/v1

Oracle. (n.d.). *What is Artificial Intelligence (AI)?* Retrieved May 29, 2022, from https://www.oracle.com/artificial-intelligence/what-is-ai/

Özkaya, U., Öztürk, Ş., & Barstugan, M. (2020). *Coronavirus (COVID-19) Classification Using Deep Features Fusion and Ranking Technique* (pp. 281–295). https://doi.org/10.1007/978-3-030-55258-9_17

Patel, K. (2019). *Convolutional Neural Networks — A Beginner's Guide*. https://towardsdatascience.com/convolution-neural-networks-a-beginners-guide-implementing-a-mnist-hand-written-digit-8aa60330d022

Peper, J. S., & Dahl, R. E. (2016). Radiomics: a new application from established techniques. *Expert Review of Precision Medicine and Drug Development*, *1*(2), 207–226. https://doi.org/10.1080/23808993.2016.1164013.Radiomics

Pham, T. D. (2020). A comprehensive study on classification of COVID-19 on computed tomography with pretrained convolutional neural networks. *Scientific Reports*, *10*(1), 1–8. https://doi.org/10.1038/s41598-020-74164-z

Polsinelli, M., Cinque, L., & Placidi, G. (2020). A light CNN for detecting COVID-19 from CT scans of the chest. *Pattern Recognition Letters*, *140*, 95–100. https://doi.org/10.1016/j.patrec.2020.10.001

Rad Tech On Duty. (n.d.). *CT Scan Components*. Retrieved May 29, 2022, from http://www.radtechonduty.com/2017/03/ct-scan-components.html

Radiology Key. (n.d.). *Computed Tomography - Chapter 16*. Retrieved May 29, 2022, from https://radiologykey.com/computed-tomography-3/

Rahimzadeh, M., & Attar, A. (2020). Introduction of a new Dataset and Method for Detecting and Counting the Pistachios based on Deep Learning. *ArXiv Preprint ArXiv:2005.03990*.

Reji, T. (2018). *Convolutional Neural Networks. Fundamentals of Convolutional Neural Networks*. https://medium.com/geekculture/convolutional-neural-networks-ab4b24d1f916

Rizzo, S., Botta, F., Raimondi, S., Origgi, D., Fanciullo, C., Morganti, A. G., & Bellomi, M. (2018). Radiomics: the facts and the challenges of image analysis. *European Radiology Experimental*, *2*(1). https://doi.org/10.1186/s41747-018-0068-z

Saeedi, A., Saeedi, M., & Maghsoudi, A. (2020). A Novel and Reliable Deep Learning Web-Based Tool to Detect COVID-19 Infection from Chest CT-Scan. *ArXiv Preprint ArXiv:2006.14419*. http://arxiv.org/abs/2006.14419

Sakagianni, A., Feretzakis, G., Kalles, Di., Koufopoulou, C., & Kaldis, V. (2020). Setting up an easy-to-use machine learning pipeline for medical decision support: A case study for COVID-19 diagnosis based on deep learning with CT scans. *Studies in Health Technology and Informatics*, *272*(December 2019), 13–16. https://doi.org/10.3233/SHTI200481

Saleh, N. A. A., Ertunc, H. M., Saleh, R. A. A., & Rassam, M. A. (2021). A Simple Mask Detection Model Based on A Multi-Layer Perception Neural Network. *2021 International Conference of Technology, Science and Administration, ICTSA 2021*, 0–4. https://doi.org/10.1109/ICTSA52017.2021.9406523

Santa Cruz, J. F. H. (2021). An ensemble approach for multi-stage transfer learning models for COVID-19 detection from chest CT scans. *Intelligence-Based Medicine*, *5*, 100027.

Santos, I. de A., Grosche, V. R., Bergamini, F. R. G., Sabino-Silva, R., & Jardim, A. C. G. (2020). Antivirals Against Coronaviruses: Candidate Drugs for SARS-CoV-2 Treatment? *Frontiers in Microbiology*, *11*, 1818. https://doi.org/10.3389/FMICB.2020.01818/BIBTEX

Scikit-learn. (n.d.). *Cross-validation: evaluating estimator performance — scikit-learn 1.1.1 documentation*. Retrieved May 30, 2022, from https://scikit-learn.org/stable/modules/cross_validation.html

Shaik, N. S., & Cherukuri, T. K. (2022). Transfer learning based novel ensemble classifier for COVID-19 detection from chest CT-scans. *Computers in Biology and Medicine*, *141*(June 2021), 105127. https://doi.org/10.1016/j.compbiomed.2021.105127

Sharma, S. (2020). Drawing insights from COVID-19-infected patients using CT scan images and machine learning techniques: a study on 200 patients. *Environmental Science and Pollution Research*, *27*(29), 37155–37163. https://doi.org/10.1007/s11356-020-10133-3

Shi, F., Xia, L., Shan, F., Song, B., Wu, D., Wei, Y., Yuan, H., Jiang, H., He, Y., Gao, Y., Sui, H., & Shen, D. (2021). Large-scale screening to distinguish between COVID-19 and community-acquired pneumonia using infection size-aware classification. *Physics in Medicine and Biology*, *66*(6). https://doi.org/10.1088/1361-6560/abe838

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84–90. https://doi.org/10.1016/J.INFFUS.2021.11.011

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Song, Y., Zheng, S., Li, L., Zhang, X., Zhang, X., Huang, Z., Chen, J., Wang, R., Zhao, H., Chong, Y., Shen, J., Zha, Y., & Yang, Y. (2021). Deep Learning Enables Accurate Diagnosis of Novel Coronavirus (COVID-19) with CT Images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *18*(6), 2775–2780. https://doi.org/10.1109/TCBB.2021.3065361

Tahamtan, A., & Ardebili, A. (2020). Real-time RT-PCR in COVID-19 detection: issues affecting the results. *Expert Review of Molecular Diagnostics*, *20*(5), 453–454. https://doi.org/10.1080/14737159.2020.1757437

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 10691–10700.

Tang, Y., Schmitz, J. E., Persing, D. H., & Stratton, C. W. (2020). Laboratory Diagnosis of COVID-19: Current Issues and Challenges. *Journal of Clinical Microbiology*, *58*(6), 1–9.

UNICEF. (2021). *Everything you need to know about washing your hands to protect against coronavirus (COVID-19) | UNICEF*. https://www.unicef.org/coronavirus/everything-you-need-know-about-washing-your-hands-protect-against-coronavirus-covid-19

Veterian Key. (n.d.). *Principles of Computed Tomography and Magnetic Resonance Imaging*. Retrieved May 29, 2022, from https://veteriankey.com/principles-of-computed-tomography-and-magnetic-resonance-imaging/

Wang, H., Wei, R., Rao, G., Zhu, J., & Song, B. (2020). Characteristic CT findings distinguishing 2019 novel coronavirus disease (COVID-19) from influenza pneumonia. *European Radiology*, *30*(9), 4910–4917. https://doi.org/10.1007/S00330-020-06880-Z/TABLES/2

Wang, M. Y., Zhao, R., Gao, L. J., Gao, X. F., Wang, D. P., & Cao, J. M. (2020). SARS-CoV-2: Structure, Biology, and Structure-Based Therapeutics Development. *Frontiers in Cellular and Infection Microbiology*, *10*, 724. https://doi.org/10.3389/FCIMB.2020.587269/BIBTEX

Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., Cai, M., Yang, J., Li, Y., Meng, X., & Xu, B. (2021). A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19). *European Radiology*, *31*(8), 6096–6104. https://doi.org/10.1007/s00330-021-07715-1

Weidman, S. (2019). *Deep Learning from Scratch: Building with Python from First Principles*. O'Reilly Media. https://books.google.com.tr/books?id=MrWuDwAAQBAJ

WHO. (2020). *WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020*. https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020

Xu, M., Wang, D., Wang, H., Zhang, X., Liang, T., Dai, J., Li, M., Zhang, J., Zhang, K., Xu, D., & Yu, X. (2020). COVID-19 diagnostic testing: Technology perspective. *Clinical and Translational Medicine*, *10*(4), 1–15. https://doi.org/10.1002/ctm2.158

Zhao, J., Zhang, Y., He, X., & Xie, P. (2020). COVID-CT-Dataset: a CT scan dataset about COVID-19. *ArXiv Preprint ArXiv:2003.13865*.

Zheng, C., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., Liu, W., & Wang, X. (2020). Deep Learning-based Detection for COVID-19 from Chest CT using Weak Label.

IEEE Transactions on Medical Imaging, 1–13.

Zhou, Z., Zhao, G., Kijowski, R., & Liu, F. (2018). Deep convolutional neural network for segmentation of knee joint anatomy. *Magnetic Resonance in Medicine*, *80*(6), 2759–2770. https://doi.org/https://doi.org/10.1002/mrm.27229

PUBLICATIONS AND WORKS

Al-Areqi, F., & Konyar, M. Z. (2022). Effectiveness evaluation of different feature extraction methods for classification of covid-19 from computed tomography images: A high accuracy classification study. *Biomedical Signal Processing and Control, 76*, 103662.

Al-Areqi, F., & Konyar, M. Z. (2022). Comparative Analysis Of Machine Learning And Deep Learning Based Covid-19 Classification Methods. *In International Marmara Sciences Congress (IMASCON 2022 SPRING), Kocaeli, Turkey.*

Al-Areqi, F., & Konyar, M. Z. (2022). Transfer Öğrenme Mimarileri Kullanılarak Bilgisayarlı Tomografi Görüntülerinden Covid-19'un Yüksek Doğrulukla Sınıflandırılması. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, (accepted for publication).

BIOGRAPHY

He completed his high school in Al-Thawra high school-Sana'a-Yemen in 2013. He enrolled at the University of Science and Technology-Sana'a-Yemen in Biomedical Engineering in 2014 and graduated in 2018. He worked as a trainee Biomedical Engineer at the University of Science and Technology Hospital and Al-Thawra Hospital from 2018 until 2019. He is currently a master's student in the Department of Biomedical Engineering, University of Kocaeli. He loves reading about technology and learns new skills in his free time.