REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

COVID-19 DIAGNOSIS WITH ARTIFICIAL INTELLIGENCE ALGORITHMS

Lujain Qasim Naser LAMI

Master's Thesis

Supervisor

Asst. Prof. Dr. Mesut ÇEVİK

Istanbul, 2023

COVID-19 DIAGNOSIS WITH ARTIFICIAL INTELLIGENCE ALGORITHMS

Lujain Qasim Naser LAMI

Electrical and Computer Engineering

Master's Thesis

ALTINBAŞ UNIVERSITY

2023

The thesis titled COVID-19 DIAGNOSIS WITH ARTIFICIAL INTELLIGENCE ALGORITHMS prepared by LUJAIN QASIM NASER LAMI and submitted on 12/04/2023 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering.

____

Asst. Prof. Dr. Mesut ÇEVİK

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Mesut ÇEVİK	Department of Electrical and Electronics Engineering, Altınbaş University	__________________
Asst. Prof. Dr. Timur INAN	Department of Software Engineering, Altınbaş University	__________________
Asst. Prof. Dr. Atilla ÖZMEN	Department of Electrical and Electronics Engineering, Kadir Has University	__________________

I hereby declare that this thesis meets all format and submission requirements of a master's thesis.

Submission date of the thesis to the Graduate Education Institute: ___/___/___

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Lujain LAMI

Signature

DEDICATION

To my darling husband Mustafa ….

My first Supporter has always supported what I wanted to do and has respected and encouraged me wholeheartedly along the road.

Special gratitude to my mother, father, brothers, and my darling little girl (Deniz) for being a part of my life and for their unwavering support while I was studying.

PREFACE

To begin with, I would like to thank my advisor, Assistant Professor Dr. Mesut ÇEVİK, for giving me guidance and support while I was writing this dissertation. As a consequence of the regular discussions I have with my supervisor, Assistant Professor Dr. Mesut ÇEVİK, in which we discuss my progress, difficulties, and ideas, my understanding of the logic underlying the research has significantly improved. It made me more conscious of the necessity for this research from a practical standpoint.

ABSTRACT

COVID-19 DIAGNOSIS WITH ARTIFICIAL INTELLIGENCE ALGORITHMS

LAMI, Lujain Qasim Naser

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Mesut ÇEVİK

Date: April / 2023

Pages: 73

When the covid-19 virus initially spread globally in 2019, As a result, numerous people passed away. Due to this sickness, medical staff members are busier than ever. For reducing this load, computer-aided diagnostic tools and machine learning will assist in some ways. Any scientific investigation into this disease can help it be eliminated rapidly. In our study, we utilized the Covid-19 dataset which is publicly accessible (14486 cases) from the Kaggle website. Training and test data categories were created from the dataset, using 80% of the data for training and 20% for testing. This research has two main stages: First, we extract Visual Geometry Group (VGG16) features from the dataset, then based on these extracted features, five machine learning algorithms were employed for classification (k-nearest neighbor (KNN), random forest (RF), extreme gradient boosting (XGBoost), decision trees (DT), and support vector machine (SVM)). Second, we extract gray-level co-occurrence matrix (GLCM) features from the dataset. Then, we applied the same five machine learning algorithms (DT, RF, KNN, XGBoost, and SVM) for classification based on these extracted features. Performance metrics were examined utilizing a confusion matrix, accuracy, recall, and precision together with the F1 score.

Regarding classification accuracy, the Support vector machine algorithm and VGG16 were the classification techniques that excel at 98.22% accuracy and achieve the impressively rapid and best result.

Keywords: COVID-19, Comparing Classification Algorithms, Machine Learning, Deep Learning, Chest CT Scan, GLCM, VGG16.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

ANN	:	Artificial Neural Network
AI	:	Artificial Intelligence
CNN	:	Convolutional Neural Network
VGG16	:	Visual Geometry Group
RT-PCR	:	Reverse Transcription-polymerase Chain Reaction
ML	:	Machine Learning
DL	:	Deep Learning
DNN	:	Deep Neural Network
SVM	:	Support Vector Machine
DT	:	Decision Tree
XGBoost	:	Extreme Gradient Boosting
CT	:	Computerized Tomography
GLCM	:	Gray-level Co-occurrence Matrix
RF	:	Random Forest
KNN	:	K-Nearest Neighbor
NN	:	Neural Network
CAD	:	Computer-Aided Diagnosis
RTLAMP	:	Reverse Transcription Loop-mediated Isothermal Amplification
ESR	:	Erythrocyte Sedimentation Rate
GGO	:	Ground Glass Opacities

ReLU : Rectifier Linear Unit

CRP : C-reactive Protein

1. INTRODUCTION AND PURPOSE

In late December 2019, a new version of the coronavirus was discovered in China. After that, it rapidly spread throughout the entire world [1]. The World Health Organization (WHO) defined COVID-19 as an epidemic and SARS-CoV-2 to be the associated coronavirus on March 11, 2020[2]. Even though COVID-19 is a condition that can be adequately treated, there is a chance that it will cause death. Because of the substantial alveolar damage and accumulative respiratory failure brought on by the manifestation of intensive disease, death may follow. The most crucial course of action is to stop the disease from spreading; this is made feasible by the availability of quick and precise diagnostic techniques. Reverse transcription-polymerase chain reaction (RT-PCR)-based laboratory assays are thought to be the gold standard for diagnosing. These tests, however, can produce false-negative results. Inadequate RT-PCR test tools may also affect the last therapeutic choice and course of the treatment plan due to the worldwide pandemic situation. Within those conditions, it was acknowledged that chest computerized tomography (CT) imaging was an essential diagnostic method necessary for both the prognosis and diagnosis of COVID-19 cases [2]. Because the CT scan provides speedy diagnostic results and has a low chance of misdiagnosis, early isolation of patients is achieved, which helps avoid the spread of disease [3]. Recent years have seen significant development in both machine learning (ML) and artificial intelligence (AI). Medical image processing, image fusion, Computer-Aided Diagnosis (CAD), picture interpretation, image retrieval, image analysis, image-guided therapy, image registration, and image segmentation are just a few of the tasks that have greatly benefited from the application of AI and ML techniques. In order to extract information from images and use it effectively, ML techniques handle the images. By using AI and ML, researchers are better able to evaluate the various symptoms that can point to a certain disorder, which speeds up and improves disease diagnosis [2].

Finally, it was discovered that computerized tomography (CT) scans are good clinical techniques for corona patient diagnosis. Following these findings, CT scans were increasingly carried out mainly in the Hubei area of China. During the height of the disease's spread, the radiologist had trouble distinguishing COVID-19 from CT scans due to the vast number of infected people. The pleural indentation sign, pulmonary ground-glass opacities that may be unilateral or bilateral, morphological rounded opacities and patchy consolidative pulmonary opacities that are most visible in the lower lung are the most important CT scan markers of COVID-19 [2]. Deep learning and machine learning algorithms can identify these signs as symptoms of COVID-19 infection. This was used as the main cue for processing several special ML algorithms for the analysis of different radiological images, which correctly and swiftly detected COVID-19 cases and distinguished them from other Non-COVID-19 examples [2]. In times of epidemic emergencies, it is meant to help medical workers quickly classify patients. After the Coronavirus era is over, this technique can be used for other medical images [2].

1.1 COVID-19

A major viral family called coronaviruses typically attacks the respiratory system. The phrase "virus" is a translation of the Latin term "corona," which means "crown" and refers to a spiny edge that envelops these viruses. Many animals, such as cats, birds, and bats, contract illnesses. Only seven, including Covid-19, SARS, and MERS, have been linked to human infection [2]. The creation of rapid and precise procedures for COVID-19 diagnosis is an essential part of avoiding new infections.

Technologies used in the detection of antibody and antigens include genome sequencing, nucleic acid molecular testing, clustered regularly interspaced short palindromic repeats editing, computed tomography imaging, and others have all been used in combination to detect and screen COVID-19 infections. existing SARS-CoV-2 detection techniques with the goal of assisting researchers in the development of efficient and timely technology to identify this newly emerging virus and its variations [3]. The coronavirus symptoms are depicted in (Figure 1.1).

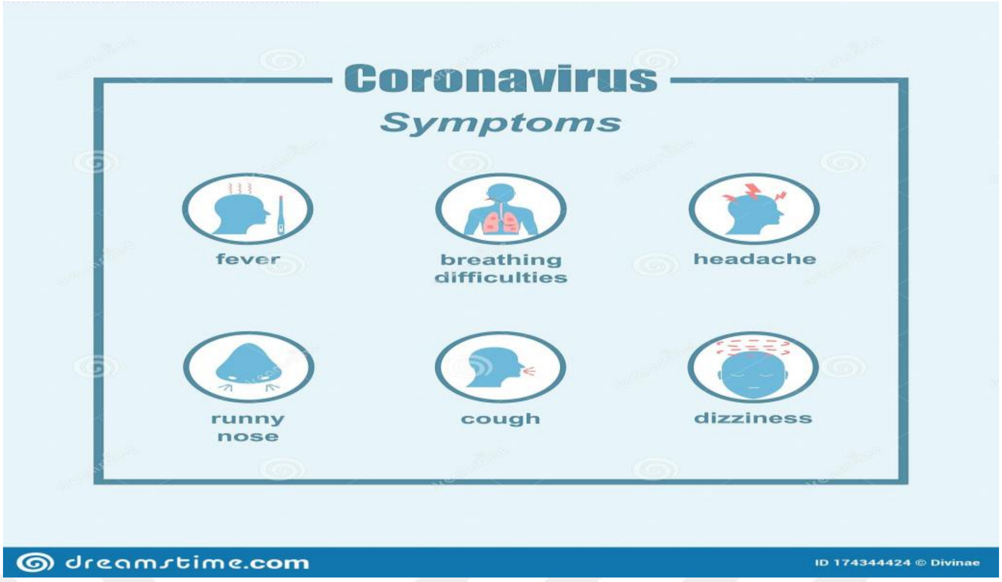


Figure 1.1: The Coronavirus's Symptoms.

1.2 THE MAIN DETECTION TECHNIQUES

1.2.1 Swab

Real-time RT-PCR (rRT-PCR), reverse transcription loop-mediated isothermal amplification, and reverse transcription polymerase chain reaction (RT-PCR) are the most frequently used molecular methods for the detection of COVID-19 (RT-LAMP). These techniques are used on swab samples (nasal, nasopharyngeal, and oropharyngeal). However, these tests had drawbacks, including the length of time it took to get answers, the prevalence of false-positive or false-negative results, and the scarcity of test kits [4].

1.2.2 Tests of Blood

C-reactive protein (CRP), differentiated white cell count, erythrocyte sedimentation rate (ESR), and D-dimer tests, coupled with a medical history, are thought to be quick techniques for diagnosing COVID-19 cases [4].

1.2.3 Medical Imaging

The clinical assessment of COVID-19 patients required the use of various imaging techniques, including magnetic resonance imaging (MRI), lung ultrasounds, chest CT scans, and X-rays. In this thesis Using Chest Computed Tomography (CT) to detect coronavirus, when PCR tests are not available, inconclusive chest radiograph findings, and extremely unwell patients, CT scans are frequently employed in decision-making [10].

1.3 RADIOLOGICAL SIGNS OF COVID-19 USING CHEST CT SCAN

The distribution of the infection, the density of the lesion, and the number of batches can all be used to pinpoint the COVID-19 chest CT result.

1.3.1 Distribution

The distribution of the batches often includes diffuse distribution, lobular distribution, sub-pleural distribution, and mixed types of distribution. The big airway is where the little virus particles travel during inhalation, and thanks to its many cilia, it has a reasonably effective immune system**.** After generating bronchiolitis and peripheral inflammation in the bronchioles, the virus then spreads throughout the entire lung tissue.

1.3.2 Number

The lesions are often many and bilaterally spread across the lung, while they can occasionally appear as a single lesion in milder cases or early stages of infection.

1.3.3 Density

There are three different ways to describe a lesion's density: consolidation, consolidation, and ground glass opacities (GGO), respectively. The onset of marginal fuzzy GGO marks the beginning of the infection. The fuzzy GGO becomes more noticeable as the illness develops, and then the consolidation starts to appear with a straight edge [4].

1.4 ARTIFICIAL INTELLIGENCE

In the 1950s, an attempt was made to automate intellectual functions that were previously handled by humans; this was the birth of artificial intelligence. This makes AI a broad field that includes both deep learning and machine learning [5], as seen in (Figure 1.2).

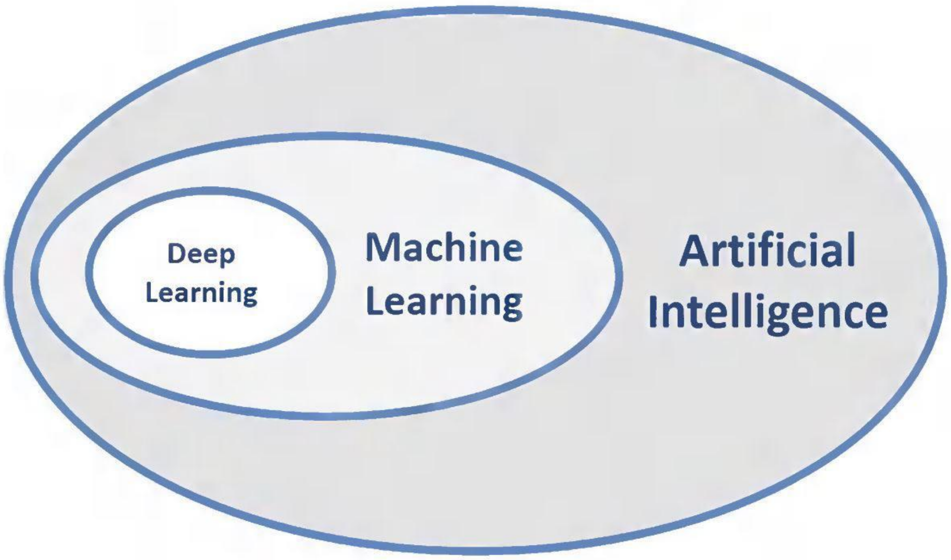


Figure 1.2: Artificial Intelligence.

1.4.1 Machine Learning

Machine learning is a system that is trained rather than formally designed. If you provide a machine learning system with a lot of instances of images that have previously been labeled by people, the algorithm will learn statistical principles for connecting certain images to specific tags [5]. Shallow learning is a term sometimes used to describe machine learning systems that only use one or two layers of representations of the data. Machine learning models are Utilized to solve two main tasks: classification and regression [6], in this thesis we use classification.

1.4.2 Deep Learning

Deep learning (DL) is a subfield in machine learning that focuses on creating algorithms that can learn and explain both high-level and low-level data abstractions, which are sometimes impossible for conventional machine learning algorithms to do. Deep learning models frequently take their cues from a variety of fields of study, including neuroscience and game theory, and many of them closely resemble the fundamental organization of the human nervous system [7]. Deep learning is the name for architectures that use deep networks (many hidden layers) to learn various features at various abstraction levels [8]. (Figure 1.3) shows how traditional programming and machine learning differ from one another.

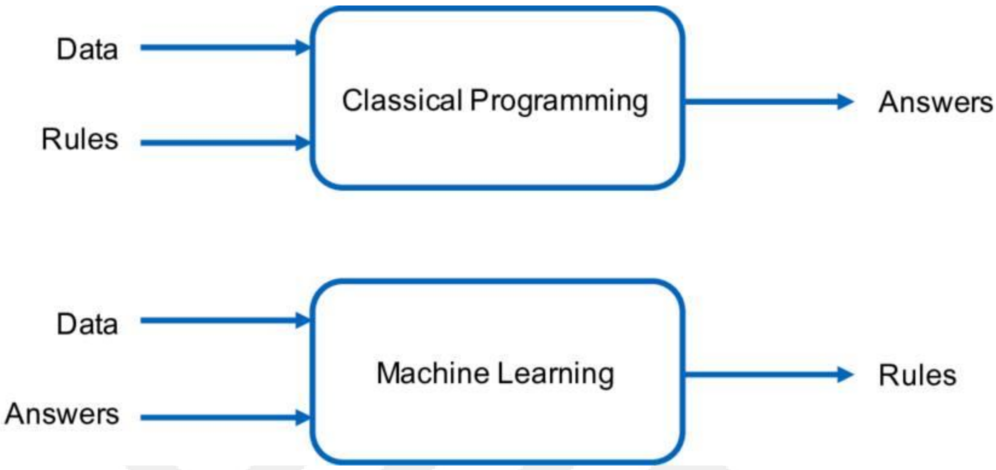


Figure1.3: Classical Programming Versus Machine Learning Approaches.

1.5 NEURAL NETWORKS

The learning process in biological organisms is simulated by artificial neural networks, a common machine learning technique. Neuronal cells, or neuronal tissue, are present in the human nervous system. Only one input layer and one output node make up the most basic neural network [9], as seen in (Figure 1.4).

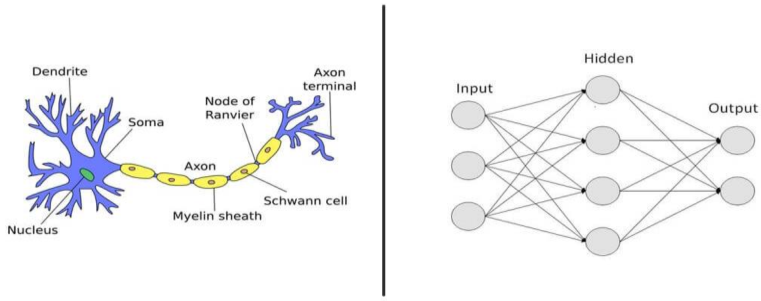


Figure 1.4: Artificial Neural Networks Versus Biological Neural Networks.

Deep neural networks are characterized by their abundance of hidden layers, which are so-called because their inputs and outputs aren't always clear to us beyond the fact that they are the result of the layer before them. One architecture differs from another based on the addition of layers and the neuronal functions found inside these layers, which also determine the various use cases of a given model [7], according to (Figure 1.5).

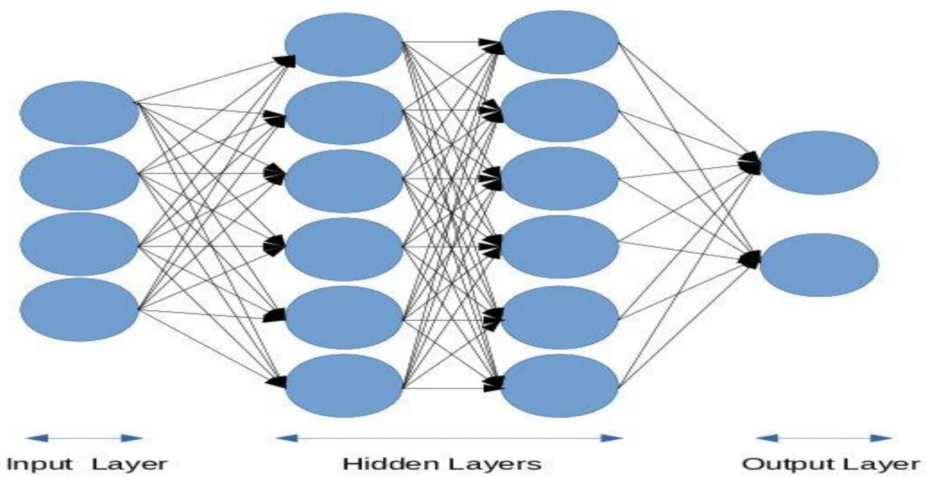


Figure 1.5: Deep Neural Network.

The number of connected nodes and hidden layers present design challenges. There isn't a standard formula for calculating the number of hidden layers in literature. Experience has shown that adding more hidden layers can minimize training errors but also make the algorithm more complex and lower the system's capacity to generalize [10]. Additionally, the system seems to be unable to modify the weights as the number of hidden layers rises, leading to the appearance of additional local minima. The model, on the other hand, does not function properly if the system's hidden layer count and the nodes in those layers are insufficient [11]. Calculations are made for the number of neurons needed in the output layers based on the output required. Single neuron represents the output in binary classification and multiple outputs in multiclass classification.

1.6 BACKPROPAGATION ALGORITHM

In the backpropagation technique, the weight vectors are initially given random values, causing the network to only carry out a sequence of arbitrary modifications. At first, the network's output may be extremely different from what it should be, which could result in a very high loss score. The weights are modified in a way that causes the loss score to go down with each example that is fed to the network. Iteratively repeating this approach yields the weight values needed to in order to reduce the loss function. When the output values of the network are as close as they can be to the target value, the network is considered to have learned [8].

Backpropagation training of MLP involves two basic processes. The backpropagation and feedforward steps.

The following succinct description of this training algorithm:

a. Make the weights initial.
b. Use the feedforward method for each sample, calculating the nodes' output value.
c. Update weights via backpropagation.
d. Repeat steps 2 and 3 as necessary until convergence is reached[11], as seen in (Figure 1.6).

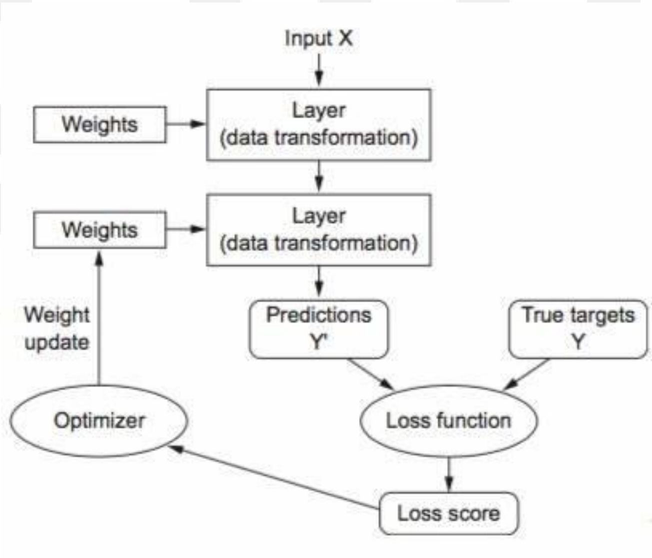


Figure 1.6: Simple Deep Neural Network Training Representation [5].

The parameters and hyperparameters used in the classification system in this thesis are further explained in the parts below, which are organized as follows.

1.6.1 Activation Functions

The selection of the activation function is crucial to the architecture of neural networks. Any given complex function could be approximated by a sufficiently large neural network employing a sigmoid, tanh, or rectifier linear unit (ReLU) function in nonlinearity components. However, the selection of the activation functions has a significant impact on the network performance in deep networks with a specified number of hidden layers [12].

Strong representational models of neural networks require activation functions [13]. When the goal is a real value, the linear activation function is frequently used at the output node. Even when a normalized replacement loss function needs to be developed for distinct outputs, it is utilized [9].

In neural networks, a variety of activation functions are employed.

1.6.1.1 The sigmoid function

Numerous activation functions are members of the logistic class of transforms, which, when graphed, look like a letter (S). This category of function is known as sigmoidal. There are various varieties in the sigmoid family of functions, and one of them is referred to as the Sigmoid function. Sigmoid can, like all logistic transforms, lower extreme values or outliers in data without really eradicating them. The decision boundary is shown as a vertical line in figure 1.7. A sigmoid function is a device that transforms independent variables with nearly limitless ranges into straightforward probabilities between 0 and 1, with the majority of its output being extremely close to either 0 or 1[14].

1.6.1.2 Tanh

Known as "Tanh", the hyperbolic trigonometric function is pronounced as "Tanch". The hyperbolic sine to the hyperbolic cosine is represented by Tanh, much as the tangent represents the ratio between the adjacent and opposing sides of a right triangle. As shown in equation (1.1).

$$tanh(x) = \frac{sinh(x)}{cosh(x)} \tag{1.1}$$

The normalized range of Tanh is from -1 to 1, unlike the Sigmoid function. Tanh's ability to handle negative numbers more easily is a benefit [14], as shown in Figure (1.7).

1.6.1.3 Rectified linear unit

The rectifier, also known as a rectified linear unit, is one of the most widely utilized activation functions in modern technology (ReLU). The rectifier was first released in the early 2000s, and it was then successfully revived in 2010. ReLU features a substantially steeper profile, facilitating quicker and more effective learning [15], according to (Figure 1.7).

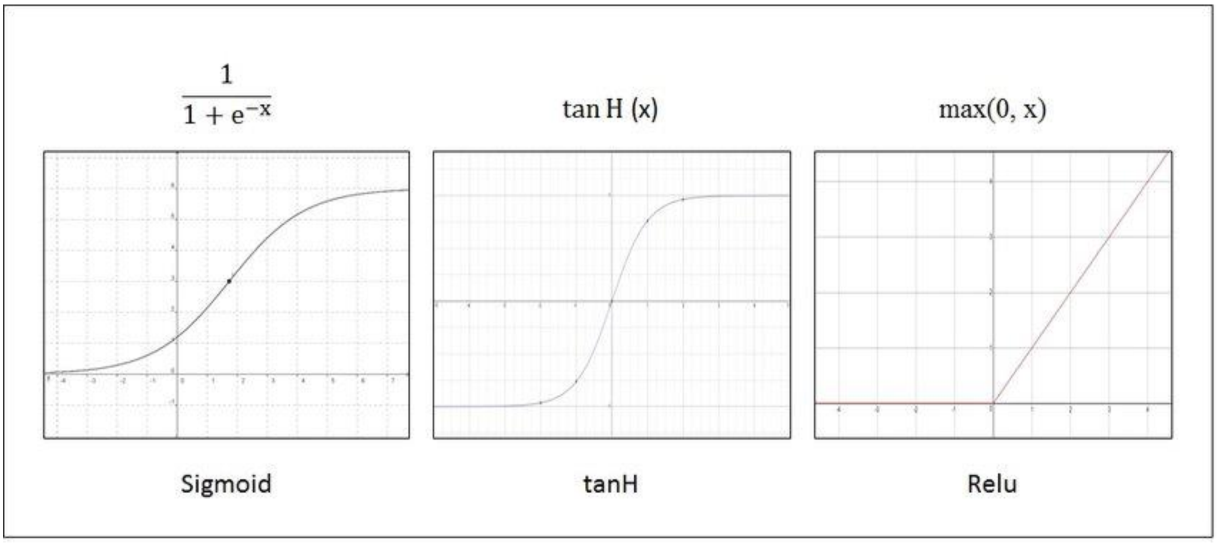


Figure 1.7: Three Neural Network Activation Functions.

1.6.2 Loss Function

The Loss function, or objective function, refers to the amount that will be reduced during training. It serves as a benchmark for the task at hand's level of success [5]**.** During a deep network's training phase, It utilizes a loss function to determine the mistake (difference between the prediction and the ground truth label) [8].

1.6.3 Optimization Techniques for Gradient Descent

The magnitude of a function's gradient, this represents the graph's slope in that direction, indicates where the function's pace of growth is greatest. How can we use the idea of a gradient to enhance an algorithm iteratively keeping this in mind? An iterative process called gradient descent updates a parameter by the gradient's opposite subject to a threshold you specify or a certain number of iterations [7]. Adam Optimizer, which stands for, was utilized in this work (adaptive moment estimation). The Adam approach exponentially smoothes the first-order gradient to include momentum in the update and applies a similar "signal-to-noise" normalization to AdaGrad and RMSProp. Moreover, it immediately corrects the bias introduced by exponential smoothing when a smoothed variable's running estimate is inadvertently started to 0.[9].

$$m(t)=\beta_1 m(t-1)+(1-\beta_1)\frac{\partial E}{\partial w_{ij}} \tag{1.2}$$

$$v(t) = \beta_2 v(t-1) + (1-\beta_2)\left(\frac{\partial E}{\partial w_{ij}}\right)^2 \tag{1.3}$$

$$w_{ij_{new}} = w_{ij} - \frac{\alpha}{\sqrt{v(t)}} m(t) \tag{1.4}$$

where the suggested values for the learning hyper-parameters, $\beta 1$ and $\beta 2$, are 0.9 and 0.999, respectively. The adaptive learning algorithm Currently, Adam is the most famous and commonly utilized [11].

1.6.4 Learning Modes

When neural networks are used, there are three different learning modes that can be used:

1.6.4.1 Stochastic learning

Backpropagation occurs when only one sample has been propagated. Thus, each operation includes a weight update.

1.6.4.2 Batch learning

The dataset is transmitted in its entirety, and weights are changed based on the cumulative sum of the datasets [11].

1.6.4.3 Mini-batch learning

The datasets utilized for this particular gradient descent application are separated into a predetermined number of batches, and the biases and weights are only changed after every batch has been entered into the model. The field of deep learning uses this technique the most frequently by a large margin [16].

1.6.5 Learning Rate

The hyperparameter causes the model weights to adjust in response to estimated errors [17]. which is one of the most crucial features of the gradient descent technique and is symbolized as (α). The learning rate controls how quickly the gradient descent method finds the best answer. The learning rate is normally initialized at a low value, usually.01 or less [7]. In training, selecting the right learning rate parameter is essential. High learning rates speed up the decay of the loss function, but they can also cause it to get trapped in local minima. By keeping an eye on the loss plots during the training process, an optimal value can be selected [4], as shown in (Figure 1.8).

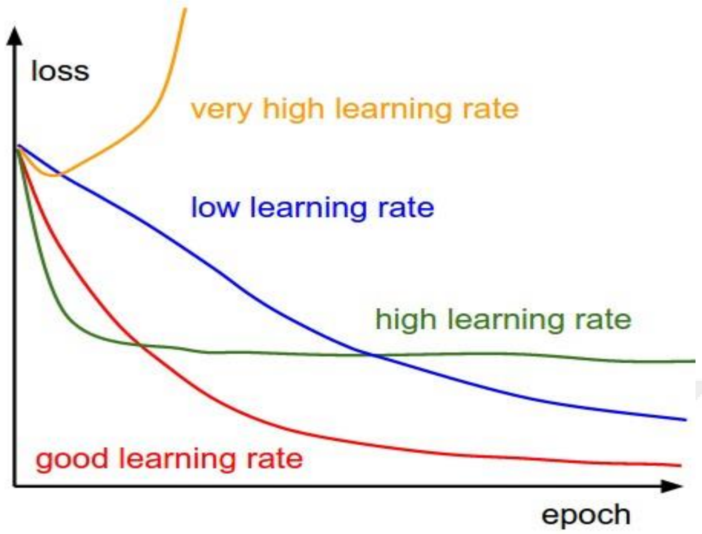


Figure 1.8: Learning Rate's Impact on Training Loss.

1.6.6 Epochs

The training epochs represent the quantity of data flows made through the network during training. For ALL training samples, 1 epoch is equal to 1 forward pass and 1 backward pass. Short epochs (don't have enough training) lead to underfitting (poor accuracy in training and validation), whereas a high number of epochs (overfit) lead to an increase in training accuracy but a decrease in validation accuracy [4].

1.6.7 Dropout

A dropout is a simple but useful method for training a deep network with a little amount of data. Dropout aims to erratically disable a portion of the units, it naturally aids in avoiding overfitting by eliminating complex co-adaptations with dropout, this enhances the trained model's generalization, as shown in (Figure 1.9). Another notable outcome of dropout is that it offers a method for effectively merging exponentially numerous distinct network designs. It can be argued that each iteration trains a separate network, but the connection weights are shared because the random and temporal removal of units during training results in diverse network designs. The network should be fully operational during testing, with no dropout, but the weights are half to maintain the same output range [18].

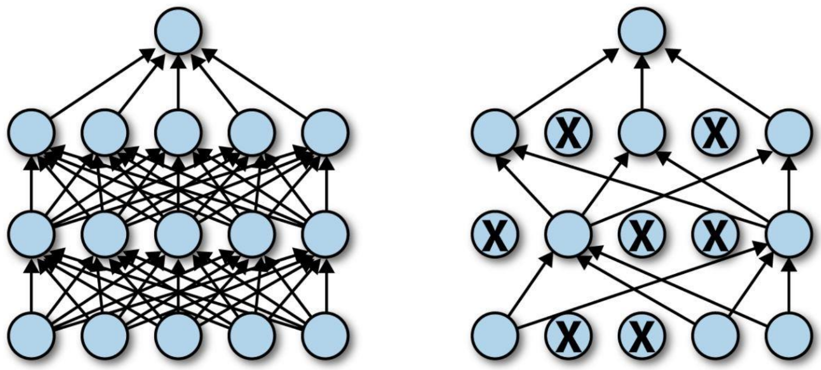


Figure 1.9: Comparison of The Same Network with A Dropout Application and A Regular Deep Neural Network. Units That Have Been Deactivated Are Represented as Circles with A Cross Inside.

1.7 PROBLEM STATEMENT

When epidemic diseases spread, whether Corona or another epidemic, it is difficult for the medical staff to control the situation due to the fact that the number of afflicted people has doubled that of the staff. On the other hand, avoiding mixing infected people with medical staff to prevent infection and increase the spread of the epidemic, so the need for technological intervention has become urgent, as artificial intelligence has contributed significantly. And we simply intend to provide assistance in urgent situations, not to replace medical personnel with technology. All medical institutions must employ artificial intelligence because it has helped to forecast the presence or absence of disease through images.

1.8 STUDY MOTIVATION

It is challenging for the medical personnel to maintain numerical control of the situation when epidemic diseases spread, as the number of sick people doubles that of the medical staff. On the other hand, keeping sick persons away from medical personnel is important to prevent infection and slow the development of the pandemic. Because artificial intelligence has made such a large contribution, technology intervention is now urgently needed. And we don't want to use technology to replace medical professionals, we only want to help in emergencies. Considering that artificial intelligence has made it possible to predict the occurrence or absence of disease through.

1.9 HYPOTHESIS STATEMENT

The effectiveness of deep learning and machine learning algorithms for categorizing the Corona epidemic from chest CT scans is evaluated in this study. We'll contrast algorithms' accuracy as well.

1.10 GOALS

Focus on the following in order to reach this goal:

a. Determine how well deep learning and machine learning systems can categorize the Corona epidemic from chest CT scans. Also, comparison of the accuracy of algorithms.

b. Provide experts with automatic detection tools, so they can act quickly and protect citizens.

c. Offer a perfect and precise deep learning or machine learning model.

1.11 RESEARCH QUESTIONS

a. Which machine learning algorithms are most effective for determining whether a person is healthy or has the Coronavirus?

b. What crucial features have a bearing on classification, GLCM features or VGG16 features?

1.12 ORGANIZATION OF THESIS

The five chapters of the research work are as follows:

Chapter 1 (Introduction and purpose): This section contains information about the project's topic, highlighting its importance, our goals, and the factors that led us to choose this particular topic.

Chapter 2 (General Information): This section's goal is to provide a recent literature review that will aid in the project's comprehension and intelligibility.

Chapter 3 (Methodologies): This chapter describes the step-by-step experimental assessment process with an example of the results.

Chapter 4 (Results): All the findings were illustrated in this chapter.

Chapter 5 (Discussion and Conclusions): In this chapter conclusion of the lookup carried alongside the prospect is discussed.

2. GENERAL INFORMATION

2.1 INTRODUCTION

At a conference in San Francisco in 1956, John McCarthy created the term artificial intelligence (AI), and the field of AI developed from it. According to Kaplan and Haenlein, AI is the "ability to use existing knowledge logically and learn from it to reach given goals and tasks." In order to solve complex problems that can only be handled by experts, artificial intelligence (AI) uses computers to simulate human cognitive processes and behaviors such as intelligence, learning, and reasoning [19]. Artificial intelligence in medicine (AIM) was first proposed in the early 1970s. It is expected that artificial intelligence (AI) technologies will increase the effectiveness of medical diagnosis and treatment. In 1959, Arthur Samuel used the term "machine learning" to refer to the classification of algorithms and the building of classifiers. To accurately predict new data inputs, the method uses a model built from input data. The back-propagation algorithm approach was one of several significant advancements in machine learning algorithms when it was first developed in the early 1960s. In NN, Paul applied the automatic differentiation method in 1982. The "decision tree" machine learning technique was developed by Ross Quinlan in 1986 and includes classifying data based on a set of criteria. Due to the decision tree and random subspace method, Tin Kam Ho developed the fundamental algorithm known as "random forest" (double spatial background subtraction algorithm) [7]. The SVM model was created in 1995 by Vladimir [20]. The fundamentals of the deep learning technique were first established in 2006 by Geoffrey Hinton, a pioneer in the field of convolutional neural networks. In deep learning, a field of machine learning, convolutional neural networks belong to the most widely utilized techniques.

2.2 ARTIFICIAL INTELLIGENCE APPLICATIONS IN MEDICINE (AIM)

A sizable data bank is being built in the process of diagnosing, detecting, and treating diseases. It can be hard for medical professionals to quickly sort through this data. Machine learning is gaining popularity as a tool for medical professionals to better predict sickness and treatment outcomes as a result of its expanding use in medicine. Studies employing AI and machine learning to categorize diseases based on images have grown recently, and in this study, we focused on the categorization of chest CT scan images for Covid-

19 classification. ML can also forecast how well radiation will work. Radiation therapy is frequently administered to patients who have lung cancer, particularly cell lung cancer (SCLC, for example). On the other hand, prolonged radiation exposure can result in serious issues such as radiation pneumonitis, which can potentially have a large negative impact on morbidity and mortality in the respiratory system. ANN was used by researchers to create a system for predicting radiation severe consequences. They also created a network with a lot of memory and training data that could be better at predicting problems [21].

2.3 LITERATURE REVIEW

Many models based on machine learning and deep learning have been created to identify COVID-19 using clinical images since the epidemic's start (CT scans, and X-rays of the chest). The review that follows provides an overview of the most recent works in this field that should be taken into account when doing the work for this study:

Ü. Yavuz et al in Jun 2020 [22], used the Covid-19 dataset, It was made accessible through the website Kaggle, has been exposed to the use of the machine learning classification methods (KNN, SVM, NB, DT). With 100% accuracy, the support vector machine method produces the best classification results.

A.shamsi et al in Jul 2020 [23], Applied (VGG16, DenseNet121, ResNet50, and InceptionResNetV2) to CT scan and X-ray images in order to extract features in the initial stage. For COVID-19 identification, a deep uncertainty-aware transfer learning paradigm has been proposed. Extracted features were examined using several statistical and machine-learning modeling techniques to identify COVID-19 cases. Additionally, to identify areas where the models trained lack confidence in their judgments, we compute and display the epistemic uncertainty of categorization findings (out of distribution problem). In-depth simulation findings show that linear SVM and neural network models produce the greatest outcomes in terms of sensitivity, accuracy, specificity, and AUC. Also, it is discovered that CT scan prediction uncertainty estimates are much higher than X-ray scan prediction uncertainty estimates.

S.prabira et al in March 2021[24], used chest CT scan X-rays images to objectively review and contrast a range of computer-aided COVID-19 testing methods. The computer-aided screening procedure includes deep extraction of features, transfer learning, and image classification techniques employing machine learning. The deep extraction of features and transfer learning approach considered 13 pre-trained CNN models. The machine learning approach includes three classification models and three sets of specially created features. Used AlexNet, Resnet101, VGG16, Densenet201, VGG19, Resnet50, Resnet18, MobileNetv2, Inceptionv3, GoogleNet, Inceptionresnetv2, Xception, and ShuffleNet accuracy for the hand-crafted features with SVM and CNN. The hand-crafted features GLCM, LBP, and HOG are combined with the machine learning-based algorithms KNN, NB, and SVM. Additionally, the various classifier paradigms are also examined. Overall, 65 classification models are subjected to a comparative study comprising 39 using machine learning methods, 13 using deep extraction of features, and 13 using transfer learning. Every categorization model works more effectively. Among of 65 classification models tested on chest X-ray images, the VGG19 with SVM achieved the highest accuracy (99.81%).

M. Ahmed et al in 2021 [25], offered a system that could mechanically reveal discriminative features on chest X-ray images and was based on CNN architecture. The CNN model's SqueezeNet was used to extract features. approaches for machine learning The obtained deep discriminative features were input into (NB, NN, DT, LR, RF, and KNN) models. The NN classifier produced the best outcomes, with a sensitivity of 0.9724, a specificity of 0.9858, and an F-score of 0.972. Its accuracy was 97.24 percent. The successful detection of COVID-19 instances in CXR pictures using this study's excellent detection performance highlights the value of deep CNN features and a NN classifier technique. This would greatly expedite disease diagnosis given the available resources.

Aa. shernas mol et al in October 2021[26], used chest X-rays, for distinguished between Covid-19 patients, patients with pneumonia, and healthy people. Machine learning employed statistical techniques to unearth the image's hidden information, which could then be used to make wise decisions. Grayscale images from X-rays have essentially identical textural characteristics. They used their model to extract the GLCM characteristics from the patients' chest X-ray images. The collected patient features were then subjected to a number of conventional machine learning algorithms for classification, including (SVM, NB, and KNN). The accuracy of these classifiers was compared, and it was discovered that they were lower accurate. Then, advanced machine learning ensemble techniques for categorization, including XGBoost and Random Forest, were evaluated. The model's comparison analysis shows that employing ensemble methods with GLCM to classify a dataset of X-ray pictures outperforms GLCM combined with traditional machine-learning approaches.

M.attique et al in October 2021 [27], recommended using a Wiener filter and top-hat combo as a contrast optimization method. Two trained models using deep learning, AlexNet, and VGG16, were employed and improved in line with the classes (healthy and COVID-19). The characteristics were retrieved and fused using a parallel positive correlation in the parallel fusion approach. With the help of the entropy-controlled firefly optimizer, the best characteristics were chosen. Using classifiers based on machine learning, such as the multiclass support vector machine (MC-SVM), the chosen attributes were divided into categories. The Radiopaedia database was used in experiments, which had a 98% accuracy rate.

O. Çinare et al in November 2021 [28], were intended to automatically evaluate CT scans and promptly direct suspected COVID-19 cases to PCR tests. Deep learning methods such as ResNet-50 and Alexnet were employed to extract deep features. (KNN, SVM, DT, LDA, RF, and NB) approaches were used to classify the data and assess each participant's performance. SVM and ResNet-50 were the classification techniques that produced the best results. A 95.18% success rate was discovered.

S. V. Kogilavani et al in February 2022 [29], proposed CNN architectures such as DeseNet121, MobileNet, Xception, NASNet, VGG16, and EfficientNet were employed. The dataset includes "COVID" and "Non-COVID" CT scan images. The accuracy rates for MobileNet are 96.38%, VGG16 are 97.68%, Xception is 92.47%, DenseNet121 is 97.53%, NASNet is 89.51%, and EfficientNet is 80.19%.

A.saleh et al in March 2022[30], employed (SVM) and (KNN) classifiers that automatically collected features from seven deep learning models and were trained using manually created local binary patterns (LBP). A concatenated LBP and deep learning features were suggested to train the SVM and KNN in order to improve the performance of the classifier. The models VGG19 + LBP earned the greatest accuracy of 99.4% based on the performance criteria. The hybrid feature-trained SVM and KNN classifiers outperform the most recent model.

S Ankalaki et al in 2022 [31], distinguished between COVID-19 and non-COVID-19 classes, they used the DesNet121, VGG19, and RESNET50 deep learning convolutional neural networks with transfer learning. This study has been expanded to examine the elements that set COVID-19 photos apart from non-COVID-19 images. We have used GLCM features to do this assignment and have found that variance is the best feature for it. The CNN architectural choice has been interpreted using the GRAD-CAM method. The classification accuracy for binary classes in this study was 98.80% for VGG19 and DenseNet121 and 97.65% for ResNet50.

E Ali Abbood et al in 2022[32], Used GLCM approach method to extract statistical texture information from the photos. The three quantized copies of the original image are used to extract the GLCMs matrices. many new inputs The deep neural network's 1D CNN architecture is utilized to directly extract the useful features from GLCM matrices after lowering their dimensions with the PCA method. The approach is assessed using three datasets: the DLAI3 Hackathon COVID-19 Chest X-Ray, COVID-CT, and SARS-CoV-2 CT scan datasets. In comparison to existing methods, the suggested system improved classification accuracy, F1 score, and AUC metrics, surpassing 98%, 89%, and 93% for three datasets, respectively.

B. Jehangir et al in 2022 [33], used GLCM for increased, pre-processing, extraction of features from data, selection of features using PCA, and classification using a photo gradient enhancement machine were all steps in this investigation. The COVID-19 X-ray dataset was utilized to compare the proposed method of validation. The accuracy obtained was 92.40%.

L. Kong et al in May 2022 [34], suggested a strategy for classifying chest X-ray images that combine features from VGG16 and DenseNet. In order to extract deep features from the model, this paper incorporates an attention mechanism (global attention machine block and category attention block). To quickly obtain correct categorization, effective picture information is segmented using a residual network (ResNet). Their model can detect binary classification with an average accuracy of 98.0%. The average accuracy might be as high as 97.3% for three category categorizations.

O Jung et al in July 2022[35], performed data curation based on lung segmentation and selected critical slices for the prediction using the entropy of the GLCM value of each segment in CT data. The model, which is composed of a transformer architecture and a CNN layer, was then trained. The findings from experiments indicate that the recommended strategy considerably enhances detection performance without necessitating several difficult model tweaks.

G Ade Riyanto in 2022[36], divided the lungs into two classes lungs and Covid-19 lungs—and then created an automatic Covid-19 diagnosis system using backpropagation neural network classification and GLCM to features extracted. This study's lung classification method combined pre-processing, segmentation, feature extraction, and classification. The test results accuracy was estimated to be 85% accurate. The Covid-19 class's sensitivity value was 77.5%, compared to the normal class's sensitivity value of 92.5%. The normal class's specification value was 22.5%, while the covid-19 class's specification value was 7.5%. The created method can classify lungs using X-Ray lung images, as shown by the accuracy, sensitivity, and specification percentages.

D. Kuzinkovas et al in November 2022[37], displayed a model that had been trained using information from a COVID-QU-Ex chest x-ray together with an equal number of images from Non-COVID pneumonia cases, COVID-19 pneumonia cases, and normal cases. The real model combines pre-trained CNNs with GLCM textural properties (VGG19, ResNet50, and VGG16). On a controlled validation set of chest x-rays, it achieved a binary classification accuracy of 98.34% (COVID-19, non-COVID-19) and a discrimination accuracy of 94.68% (Non-COVID pneumonia, COVID-19, and normal chest x-rays). Additionally, they address the impact of dataset size and show that, even with a training dataset of only a thousands of images, the model can obtain a 3-class accuracy of 98.82%, but generalizability is compromised.

Table 2.1 below is a summary of previous studies.

Table 2.1: Summary of Literature Review.

NO.	Authors	Year	Methodology	Results
1	Ü. Yavuz et al	2020	NB, KNN, SVM, and DT	SVM=100%
2	A.shamsi et al	2020	VGG16, ResNet50, DenseNet121, and InceptionResNetV2 for features extraction, extracted features are subsequently analyzed using various machine learning and statistical modeling techniques.	NN and linear SVM produce the best outcomes
3	S.prabira et al	2021	VGG16, Resnet10, VGG19, Densenet201, Inceptionresnetv2, Resnet18, ShuffleNet, Resnet50, AlexNet, GoogleNet, Inceptionv3, Xception, MobileNetv2, GLCM, LBP, and HOG for feature extraction. + KNN, SVM, and NB for classification	With a 99.81% accuracy rate, the VGG19 with SVM came in first.
4	M.ahmed et al	2021	SqueezeNet for features extraction. +DT, RF, NN, NB, LR, and KNN for classification.	The NN classifier produced the best outcomes, Accuracy = 97.24 % Sensitivity = 0.9724% Specificity =0.9858% F-Score =0.972%
5	Aa. shernas mol et al	2021	GLCM for features extraction + KNN, NB, SVM, RF, and XGBoost for classification	Ensemble approaches with GLCM perform better than GLCM combined with traditional machine learning methods

Table 2.1: Summary of Literature Review "Table Continued".

6	M.attique et al	2021	features were extracted and fused using AlexNet and VGG16, Parallel Positive Correlation. The best features were picked using the entropy-controlled firefly optimization method + MC-SVM for classification	98% accuracy rate
7	O. Çinare et al	2021	ResNet-50, Alexnet for extracting deep features+ SVM, KNN, LDA, DT, RF, and NB for classification.	SVM and ResNet-50 were the classification techniques that produced the best results. = 95.18%
8	SV.Kogilavani et al	2022	VGG16, DeseNet121, MobileNet, NASNet, Xceptand ion, for classification	The accuracy rates for VGG16=97.68% DenseNet121=97.53% MobileNet=96.38% NASNet=92.47% Xception=80.19%,
9	A.saleh et al	2022	SVM and KNN for classification + LBP for features extraction	The models with the highest accuracy, 99.4%, were VGG19 + LBP
10	S Ankalaki et al	2022	RESNET50, VGG19, and DesNet121 for classification + GLCM for features extraction	The classification accuracy for binary classes in this study was 98.80% for VGG19 and DenseNet121 and 97.65% for ResNet50.
11	E Ali Abbood et al	2022	GLCM for features extraction+ CNN for classification three datasets: the COVID-CT, DLAI3 Hackathon COVID-19 Chest X-Ray, and SARS-CoV-2 CTscan datasets	Accuracy, F1 score, and AUC metrics, surpassing 98%, 89%, and 93% for three datasets, respectively.

Table 2.1: Summary of Literature Review " Table Continued".

.12	B. Jehangir et al	2022	GLCM for increased, pre-processing, feature extraction, feature selection using PCA, and classification utilizing a machine that enhances photo gradients	The accuracy was 92.40%.
13	L.kong et al	2022	combine features DenseNet and VGG16 + ResNet for classification	For three category categorizations, the average accuracy might reach 97.3%
14	O Jung et al	2022	GLCM for features extraction+ CNN for classification	The findings from experiments indicate that the recommended strategy considerably enhances detection performance without necessitating several difficult model tweaks.
15	G Ade Riyanto	2022	GLCM for features extraction+ the backpropagation neural network classification technique.	The accuracy of 85%
16	D. Kuzinkovas et al	2022	GLCM for features extraction + ResNet50, VGG19, and VGG16 for classification	The model can reach a 3-class accuracy of 98.82% and achieved a binary classification accuracy of 98.34% (COVID-19, non-COVID-19), and a discrimination accuracy of 94.68% (Non-COVID pneumonia, COVID-19, and normal chest x-rays)

3. METHODOLOGY

In this chapter, we covered the practical aspects while also outlining the techniques and datasets that were employed. We utilized the freely accessible Covid-19 dataset (14486 cases) from the Kaggle website. We separated the dataset into categories for training and test data, using 80% of the data for training and 20% for testing. This study involves two essential steps: First, we extract Visual Geometry Group (VGG16) features from the dataset, then based on these extracted features, five algorithms of machine learning were used to perform the classification (extreme gradient boosting (XGBoost), k-nearest neighbor (KNN), decision trees (DT), random forest (RF), and support vector machine (SVM)). Second, we extract gray-level co-occurrence matrix (GLCM) features from the dataset. Then, we applied the same five machine learning algorithms (DT, RF, KNN, XGBoost, and SVM) for classification based on these extracted features. Accuracy, precision, and recall were also evaluated, along with the F1 score, to investigate performance metrics. The procedure is explained in (Figure 3.1).

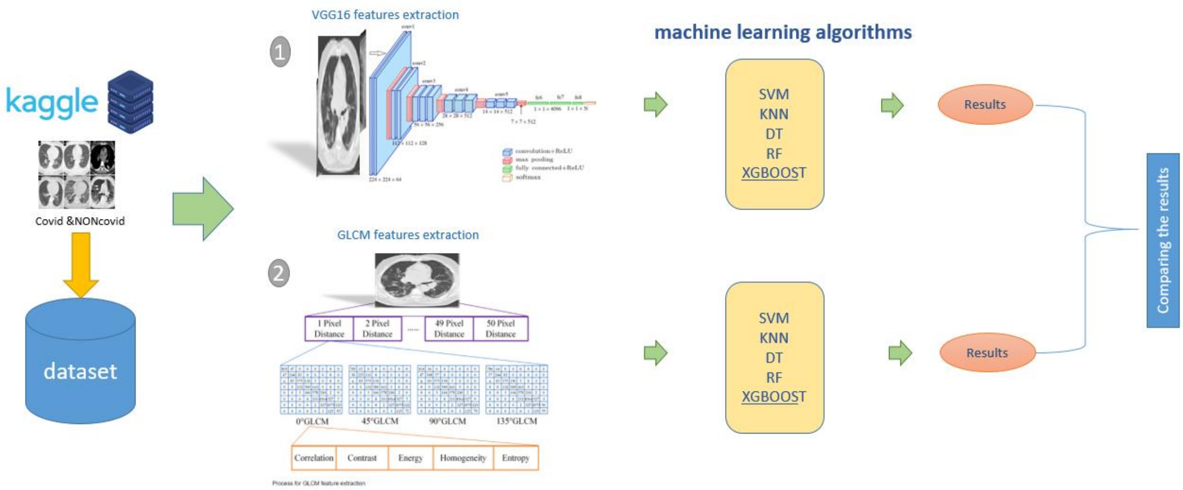


Figure 3.1: Block Diagram of The Experimental Work Steps.

3.1 DATASET COLLECTION

There were (14486) cases in all of the chest CT axial scans that were collected. However, the algorithm only took (14188) images and rejected some of the images since they did not meet the required image standards (type, shape, and size). All of the CT scan images were downloaded from the Kaggle website [42]. These included (COVID-19 7593 cases, Non-COVID-19 6893 cases), as shown in Table 3.1. The random split of the train-test set was (80%-20%), where 80% of the images (11350 images) were utilized for training and 20% of the images (2838 images) for testing. Sample data are shown in (Figure 3.2) and (Figure 3.3).

Table 3.1: Dataset Collection.

Collected cases	Total cases
COVID-19	7593
Non-COVID-19	6893
Total cases	14188

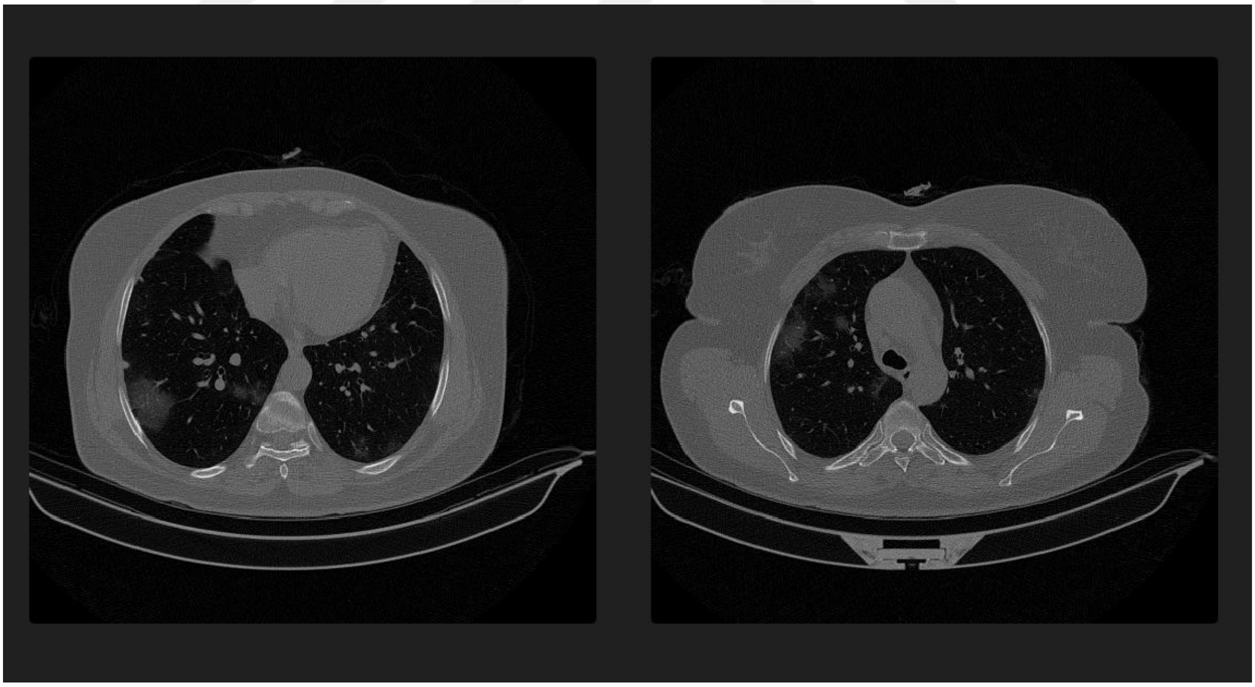


Figure 3.2: CT Scan Images for The Covid-19 Case.

Figure 3.3: CT Scan Images for The Non-Covid-19 Case.

3.2 MODELS DEVELOPMENT

A set of images is classified into preset classes through the process of image classification. Either supervised or unsupervised classification is possible. The model was trained on these target classes to categorize fresh samples using the supervised learning method, in which the training set was chosen for each target class. in this study, we divided data into two main groups based on COVID-19 symptoms (COVID-19 class and Non-COVID-19 class). Using machine learning algorithms as a classification method.

The following are the main steps in model composition:

a. Define the input data.

b. Describe the architecture of the network.

c. Set up the educational procedure.

d. Choose a training and validation methodology.

3.3 DEVELOPMENT OF A MODEL ARCHITECTURE

Regarding the functional layers of architecture, there are no official regulations. Since the first option made during practice is not the best, the model's architecture was later enhanced by repeatedly retraining it. To achieve a cutting-edge model that can perform the required task and to optimize the model architecture, it is also possible to combine multiple models with diverse but positive results.

3.4 CLASSIFICATION METHODS

The figure below (Figure 3.4) illustrates the five machine learning techniques we employed in this thesis.

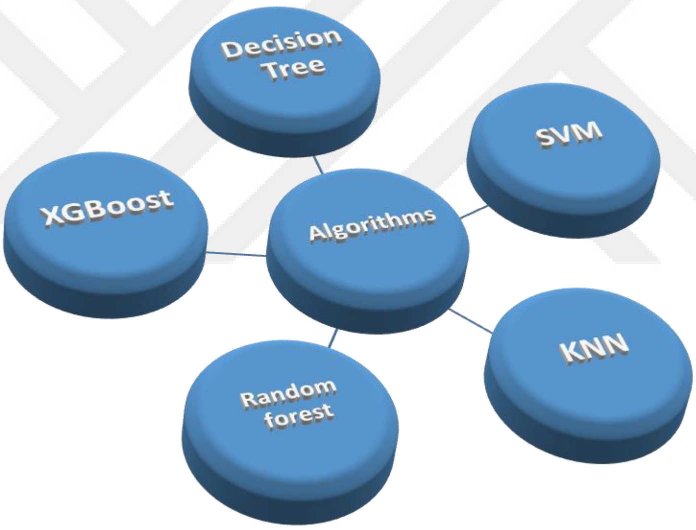


Figure 3.4: Classification Algorithms.

3.4.1 Decision Tree Algorithm

The decision tree algorithm (DT) is among the most exact methods for categorization in data mining. It is frequently applied to clustering, classification, prediction models, and the formation of subgroups inside the study domains related to an issue. The decision tree's problem-solving options are divided into a number of groups. In situations where there are numerous options, decision trees can be used to reach the best choice. Researchers can choose which aspects to consider during the decision-making process and how each component relates to the choice's potential outcomes and the past using the decision tree model. A very simple and uncomplicated model has been created using decision tree modeling [22, 43].

3.4.2 Support Vector Machine Algorithm

Classification and regression issues are successfully solved using support vector machines (SVM), which are constructed using statistical learning theory. Vladimir Vapnik and Alexey Chervonenkis' hypothesis is the foundation of SVM. The support vector machine is a classification method that creates hyperplanes for each class label in the multidimensional space by using margin values. The SVM aims to maximize margins between multiple classes by optimally separating hyperplanes. The hyperplane is one of the data instances from the provided dataset that the support vectors utilised. The margin, as depicted in (Figure 3.5), is the hyperplane's greatest separation from the support vector. After successful implementations in the 1990s, it attracted the attention of scientists working in the field of artificial intelligence.

SVMs do transformation by nonlinearly moving the collected vectors from one high-dimensional space into the low-dimensional input space. For the systems (machine or network) that executes it, a kernel that decides this conversion is supplied. In high-dimensional space during classification, the vectors moved and then became linear in nature. The best linear separator inside the separating planes is determined to be the vector with the greatest distance to classes [22, 43-45]. We employed linear SVM in our thesis out of the two forms of SVM: linear and non-linear. when employing linear SVM, a dataset is regarded to be linearly separable if it can be split into two categories by a single straight line. These data points are known as linearly separable data, and the classification algorithm used is a linear SVM classifier.

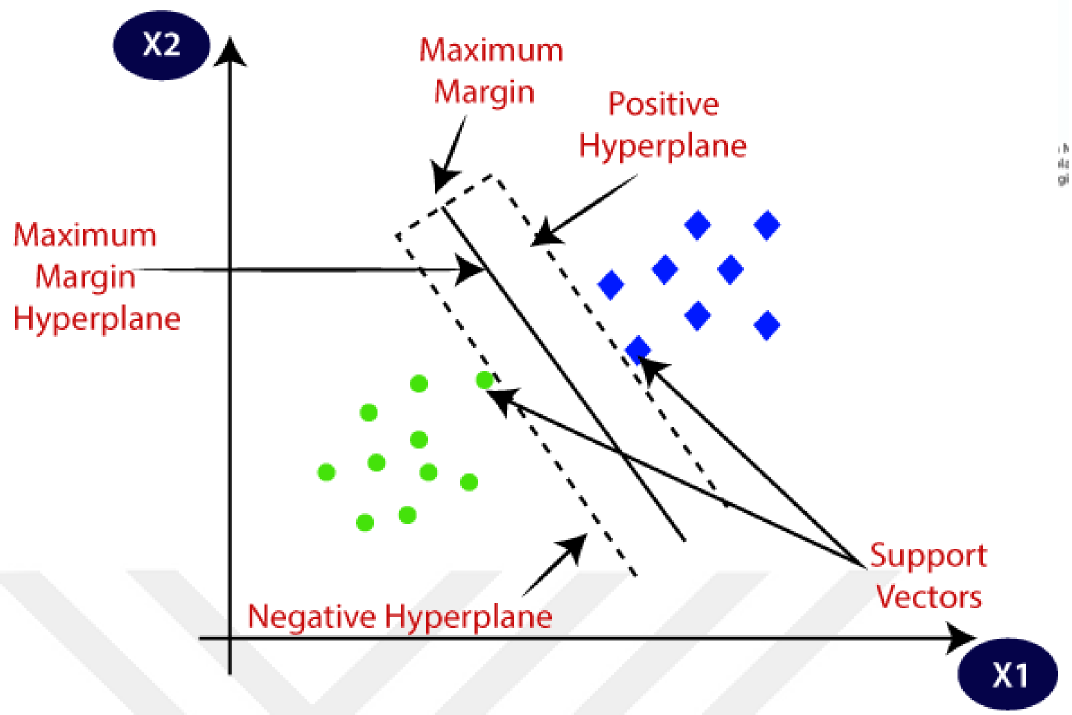


Figure 3.5: Support Vector Machine Algorithm.

3.4.3 K-Nearest Neighbor Algorithm

A machine learning strategy may benefit from using the well-known classification under the supervision algorithm KNN. The essential component of this method is the comparison of fresh data with k-training samples, commonly known as k-nearest neighbors. Depending on how far away it is from other training instances geographically, the one that is physically closest to each instance is given control over it. The nearest K value checks the data's neighbor to establish the category, and the distance between them is calculated. Euclidean distance was employed in the distance computation [22]. The Euclidean distance is defined as the total of the squared distances between the new sample (xi) and the current instance (yj), as shown in equation (3.1) [43, 44].

$$Euclidean i,j = \sqrt{\sum_{k=1}^{n} (xij - yjk)^2} \tag{3.1}$$

3.4.4 Random Forest Algorithm

The decision-making process used by the random forest (RF) classifier creates a large number of decision trees, with the final decision being made by the random forest depending on most of the trees. It is a tree-shaped graphic that is used to select a decision. Every branch of the tree symbolizes a potential choice, occurrence, or response, as shown in (Figure 3.6). One of the key benefits of using the random forest Algorithm is that it requires less training time and decreases the danger of overfitting. It additionally provides a high degree of precision. By estimating missing data, it operates effectively in huge databases and generates predictions that are practically exact [46].

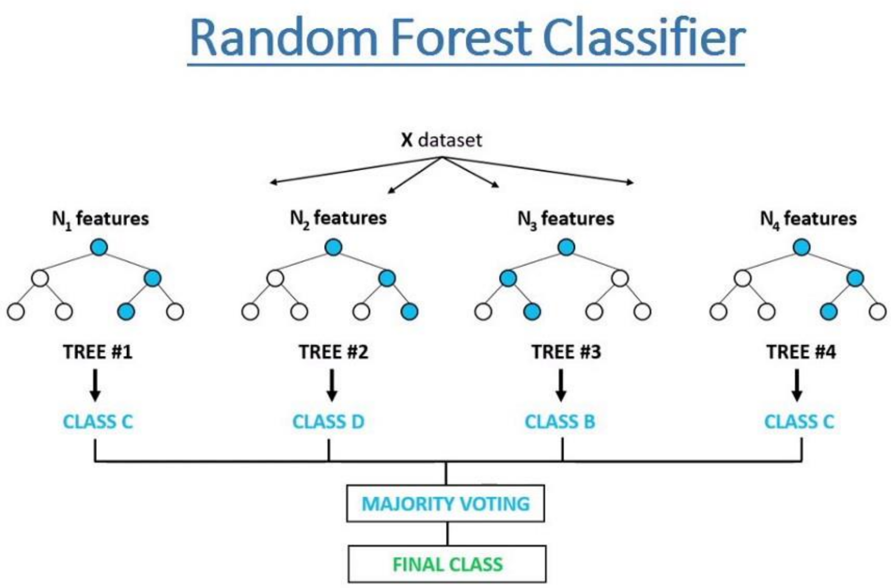


Figure 3.6: Random Forest Classifier.

3.4.5 Extreme Gradient Boosting Algorithm

The library of gradient boosting algorithms known as extreme gradient boosting, or XGBoost, is tailored for use with contemporary data science tools and issues. The fact that XGBoost is extremely scalable and parallelizable, rapid to execute, and often outperforms other algorithms are some of its main advantages.

It is an upgraded version of the decision tree as seen in (Figure 3.7). The XGBoost flow chart is depicted in (Figure 3.8). It also uses an even more normalized model formalization to avoid over-fitting, which enhances performance [46].

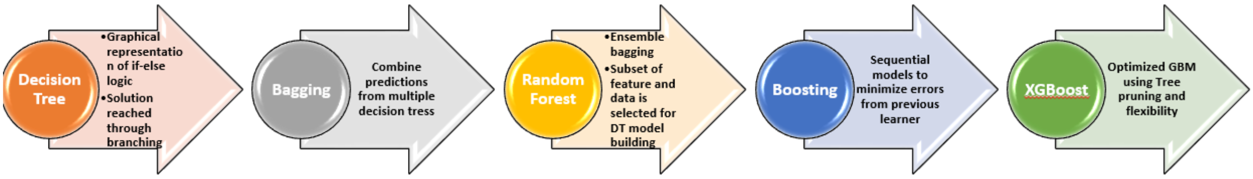


Figure 3.7: Evolution of Extreme Gradient Boosting.

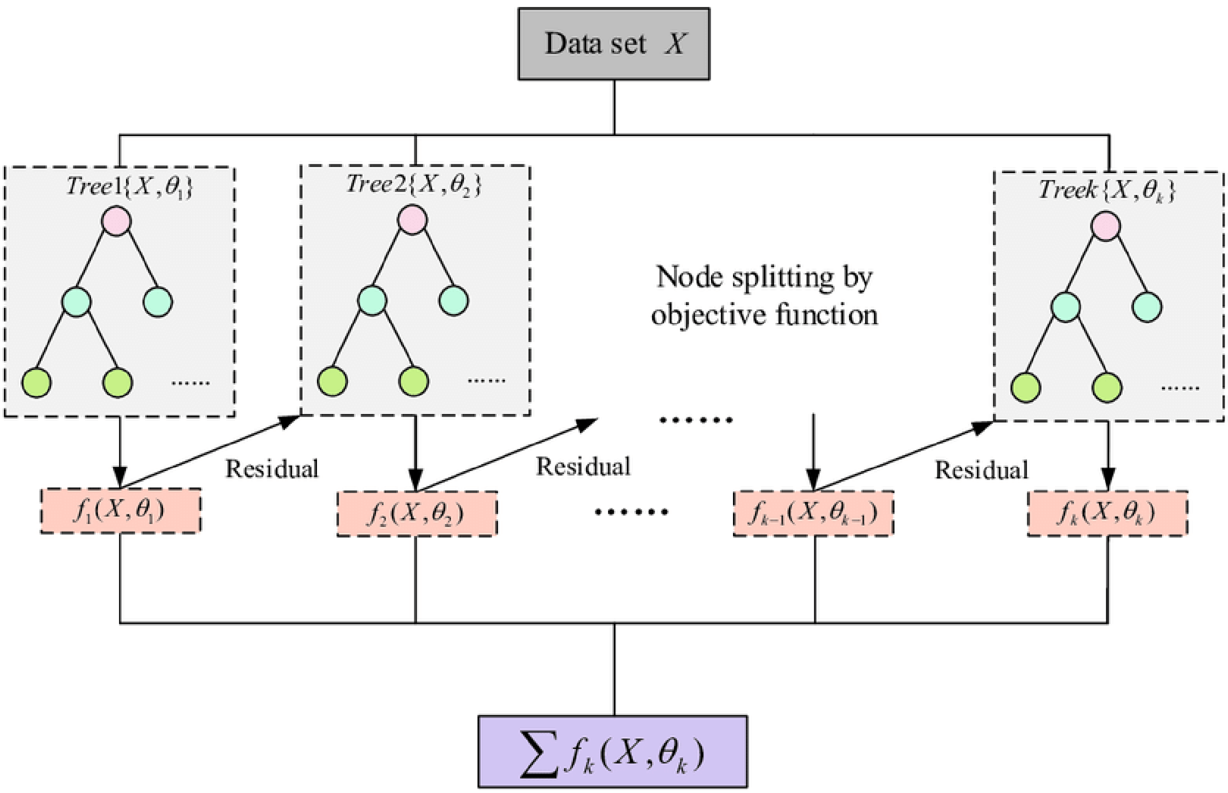


Figure 3.8: Flow Chart of Extreme Gradient Boosting.

3.5 FEATURE EXTRACTION

3.5.1 Feature Engineering

Feature engineering is the process of performing hardcoded (non-learned) transformations to the data before it enters the model utilizing your knowledge of the data and the machine-learning algorithm at hand to improve the algorithm's performance. It is frequently unrealistic to anticipate that a machine-learning model will be able to learn from fully unstructured input. It is necessary to give the model the data in a way that will simplify its task [5].

3.5.2 The Gray-Level Co-Occurrence Matrix

One of the most crucial elements in recognizing objects or areas of interest in an image is texture. Significant information on the structural organization of surfaces can be found in texture. Images are often categorized using the textural qualities provided by gray-tone spatial correlations. Spectral, textural, and contextual aspects are the three basic pattern elements that humans employ to analyze images. The fourteen textural elements that Haralick proposes provide details regarding the homogeneity, linear dependencies of gray tones, contrast, quantity and type of borders present, and complexity of the image. Information collected from blocks of graphical data blocks surrounding the area being examined makes up contextual characteristics. Co-occurrence probabilities were first used by Haralick et al. to extract various texture features from GLCM. The term "GLCM" stands for "Gray Level Dependency Matrix". It is described as a two-dimensional histogram of the gray values for a pair of pixels split by a given spatial link [47]. The relative occurrence of two grey-level pixels at a particular angle (which ranges in four directions, i.e., 0°, 45°, 90°, and 135) and a particular distance apart is shown by GLCM. The first pixel is known as the reference pixel and the second pixel is known as the neighbor pixel (ranges from 1 to the total size of the image), as shown in (Figure 3.9). The size of the GLCM square matrix, which is determined by the image's grey level scale, would be 256*256 for an 8-bit, single-channel gray-level image [4].

The count of instances of each pair of intensities appearing in the image with the specified spatial connection (d and θ) is contained in each element of the GLCM [i,j].

The definition of GLCM matrix G is:

$$x(i,j) = \sum_{m=1}^{M} \sum_{k=1}^{K} \begin{cases} 1, & if\ I(m,k) = i\ and\ I(m+dx, k+dy) = j \\ 0, & otherwise \end{cases} \tag{3.2}$$

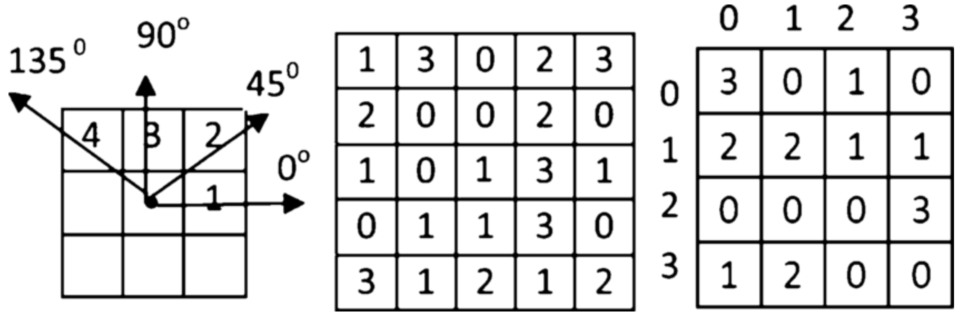


Figure 3.9: Calculating GLCM.

3.5.3 Feature of Haralick

After calculating the GLCM matrix from the image, 14 statistical texture descriptors are created for the image to provide important details on the distinctive variety of images [4]. Haralick Features are useful depictions of an image's textural qualities. attributes listed as follows:

a. Energy
b. Entropy
c. Sum Entropy
d. Difference Variance
e. Contrast
f. Variance
g. Homogeneity
h. Correlation
i. Sum Average
j. Sum Variance
k. Difference Entropy
l. Maximum Correlation Coefficient
m. The information measure of correlation 1
n. The information measure of correlation 2 [47].

3.5.4 Convolutional Neural Networks

The most common models for computer vision and image processing are convolutional neural networks (CNNs), CNN, or ConvNet, another name for a convolutional neural network. They are made to resemble the animal visual cortex in terms of structure. In particular, the width, height, and depth of CNN neurons are organized into three dimensions. The neurons in a layer are only hazily linked to a piece of the layer above them. The majority of image processing and computer vision applications use CNN models [7]**.**

CNNs are the first successful deep learning method that relies on this local neuronal connectivity as well as on hierarchically arranged picture transformations, where many layers of the hierarchy were successfully and effectively implemented. By including an architecture that takes advantage of spatial relationships and, as a result, progress feed-forward and back propagation training, the number of parameters was decreased. CNNs use features from traditional neural networks such as gradient descent, regularization, and backpropagation [48]. Most machine learning systems employ CNN for feature development and classification because it excels at both feature production and discrimination [49].

3.5.4.1 Input/output volumes

CNNs are frequently used with picture data. A matrix of pixel values makes up every image. Each pixel's bit size determines the range of values that can be encoded in that pixel. Typically, pixels with an 8-bit or 1-byte resolution. The range of values that a single pixel may represent is therefore [0, 255]. However, with colored images, especially those based on the RGB (Red, Green, Blue) color model [48]. Imagining a 3D image matrix as the input (RGB) [34]. In this thesis, we used 2D for the grayscale.

3.5.4.2 Layers of CNN

CNN's major structural elements consist of:

Convolutional layer:

Each convolution layer is made up of a collection of layers that each contains a group of neurons. The weights and biases shared by each neuron in a layer are the same.

Apply the filter over the input and create a dot product using the weights and volume of the input, as shown in (Figure 3.10) [48].

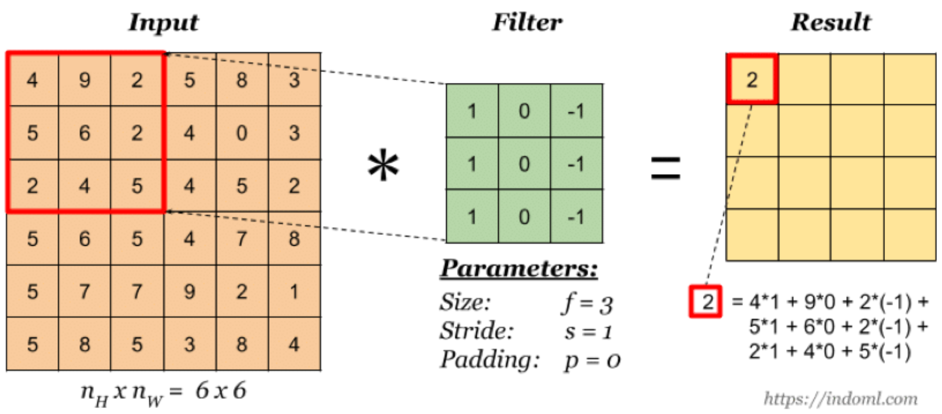


Figure 3.10: First Step of The Convolutional Layer.

Then, to obtain the following result, drag the overlay to the right place (or in accordance with the stride setting). Further, as seen in (Figure 3.11).

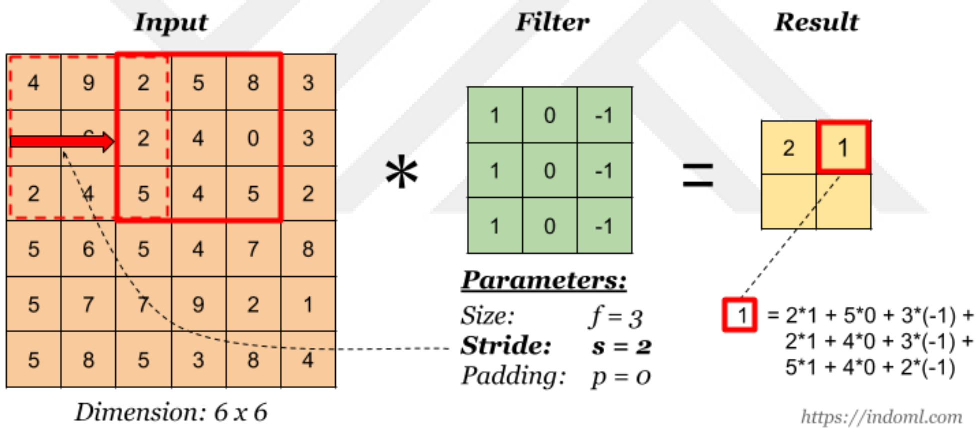


Figure 3.11: Second Step of The Convolutional Layer.

Only a small, focused area of the input image is connected by convolutional layers. The trained filters will respond to local input with the greatest strength thanks to this type of localized architecture. As a result, basic features of a limited number of input components can be created at lower layers. While the more abstract representation of vast areas is produced by the top layers [48].

Pooling layer:

A subsampling layer that lowers the dimensionality and provides some resistance to spatial shifts comes after the convolution step. The pooling layer's function is to combine semantically related features into a single entity. Since the precise location of a feature will no longer matter once it has been discovered, only its general placement in relation to other features will. Thus, just the most active unit was preserved in this manner. There are various methods for pooling the output from the preceding layer by using a tiny filter to sample the maximum or average values in a rectangular area. Every depth slice in the input is sampled by the common pooling layer by 2 along both width and height when it is applied with a stride of 2 down and has a kernel of size 2x2. (Figure 3.12) illustrates the operation of pooling layers. The box to the left has been down-sampled by obtaining the maximum value of each 2 2 sub-region [48].

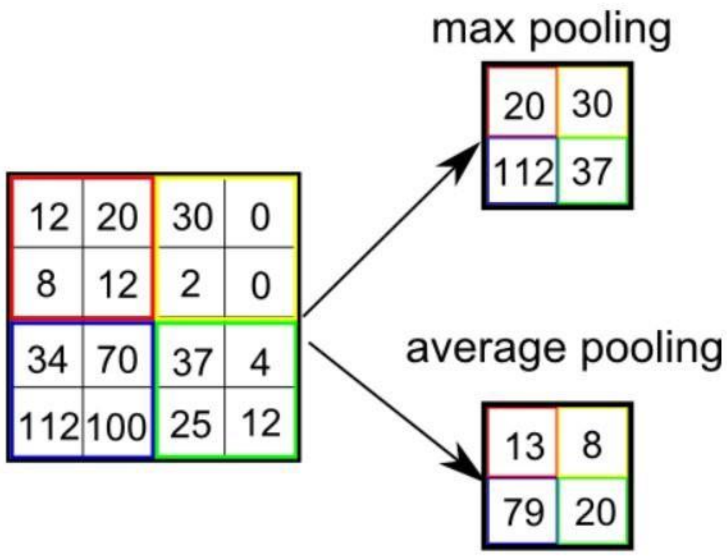


Figure 3.12: Pooling Layer.

Padding:
It refers to the image's encircling margins, which have 0 counts, as shown in (Figure 3.13). This is utilized when some pixels in the filtered image are repeatedly covered in the same frame as other pixels. Compared to typically covered pixels, it offers higher detail [17].

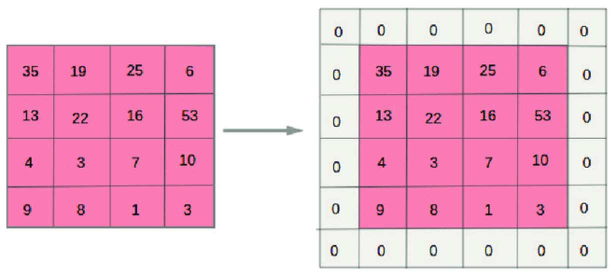


Figure 3.13: Padding Operation.

Non-linear layers:

CNNs frequently use non-linear signals, which show a different identity for each hidden layer's likely features. ReLUs and continuous (non-linear) triggering features are just two examples of the specific features that CNNs might use to execute this non-linear trigger [50].

Flatten:

The feature map (2D representation of the image) is obtained from several convolutional stacks and is passed via the flattened layer and fully connected layer before being converted to a vector, as shown in (Figure 3.14) [4].

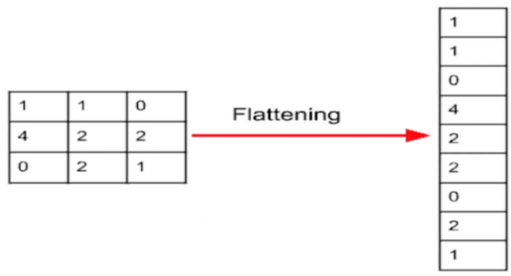


Figure 3.14: Flatten.

Layer fully connected:

The last layers of a CNN are typically used by fully connected layers. These mathematical levels rigorously weigh the preceding layer of data and determine the precise arrangement of "components" that will lead to a particular goal output consequence. The components of all the features from the preceding layer are used to calculate each component of each output feature for a connected layer [50].

3.5.4.3 Visual geometry group

The VGG model, commonly known as VGGNet, is referred to as VGG16. It is a 16-layer convolution neural network (CNN) model, as shown in (Figure 3.15). Simonyan and Zisserman were the ones who first proposed the VGG network architecture. Depending on the VGG models with 16 layers (VGG16) and 19 layers (VGG19), the Visual Geometry Group (VGG) team's proposal to the 2014 ImageNet Challenge took first and second place in the localization and classification categories, respectively. Five blocks of convolutional layers make up the VGG16 design, which is then followed by three entirely connected layers. A 33 kernel with a stride of 1 and padding of 1 is used in each activation map in a convolutional layer to preserve the same spatial dimensions as the layer before it. After each convolution, the spatial dimension is minimized using the max-pooling method, and then a rectified linear unit (ReLU) is activated. Max-pooling layers make sure that each spatial dimension of the activation map from the layer before is sliced in half by using 22 kernels with a stride of 2 and no padding. The third 1,000-connected softmax layer is used after using two fully linked layers with 4,096 ReLU-activated units. The VGG16 model's drawbacks include a high evaluation cost and extensive memory and parameter requirements. VGG16 contains more than 138 million parameters. These parameters total about 123 million in the fully connected layers [51].

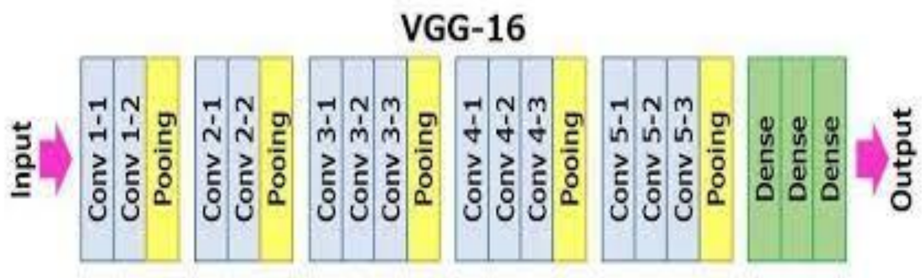


Figure 3.15: Visual Geometry Group (VGG16).

3.6 CLASSIFICATION MODELS

3.6.1 Machine Learning Algorithms Based on VGG16 Features (Model 1)

The model is composed of two key phases:

3.6.1.1 Extract VGG16 features

Based on our dataset, we first extracted features from the VGG16 model and obtained 14,714,688 parameters, according to (Figure 3.16).

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 226, 226, 3)]     0

 block1_conv1 (Conv2D)       (None, 226, 226, 64)      1792

 block1_conv2 (Conv2D)       (None, 226, 226, 64)      36928

 block1_pool (MaxPooling2D)  (None, 113, 113, 64)      0

 block2_conv1 (Conv2D)       (None, 113, 113, 128)     73856

 block2_conv2 (Conv2D)       (None, 113, 113, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

=================================================================
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688
_________________________________________________________________
```

Figure 3.16: Model of VGG16.

3.6.1.2 Classification

Based on the VGG16 features, we classified applying machine learning methods. In this model, we classified the covid19 based on the VGG16 features using five machine-learning methods (KNN, SVM, DT, XGBoost, and RF).

3.6.2 Machine Learning Algorithms Based on GLCM Features (Model 2)

This model consists of two primary phases, where Gray Level Co-occurrence was used to create second-order statistical texture features to represent each image in the dense classifier.

The model is composed of two key phases:

3.6.2.1 Extract GLCM and Haralick features

Here, the images were changed to grayscale before GLCM calculations were performed in four directions, as shown in (Figure 3.17).

a. 0° from left to right.
b. 135° from top left to bottom right.
c. 90° From top to bottom.
d. 45° from Top Right-to-Bottom Left.

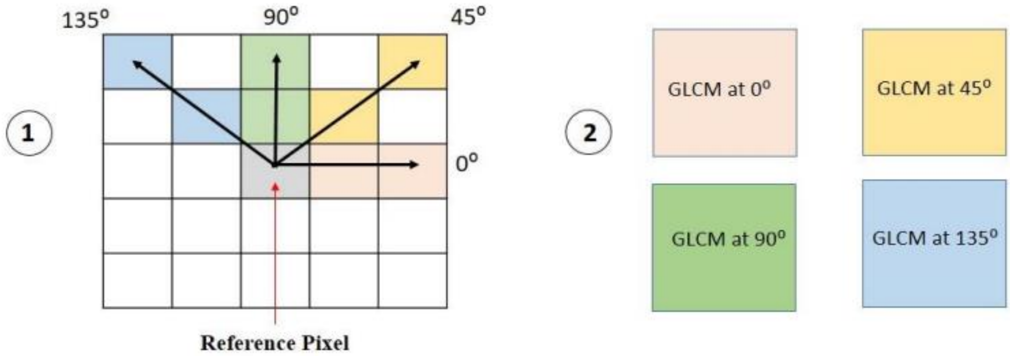


Figure 3.17: Extract GLCM Features.

Afterward, 6 of the 14 Haralick features were calculated (dissimilarity, correlation, homogeneity, contrast, energy, ASM), the test features shape is (2838 x 25) and the train features shape is (11350 x 25).

3.6.2.2 Classification

In this stage, we used the same machine learning algorithms (SVM, KNN, DT, XGBoost, and RF) to classify covid19 based on GLCM features.

3.7 LANGUAGES, LIBRARIES, PLATFORMS, AND FRAMEWORKS

The Image analysis models were implemented using an HP laptop with the following specification:

a. Central Processor unit (CPU): Intel ® core (TM) i7-12700H@ 1.80GHz2.30GHz, and RAM: 16.0 GB.

b. System type: 64-bit operating system x64/based processor.

c. Microsoft Windows 11 pro.

The software application setup included:

a. Anaconda Navigator.

b. Python3.9

c. spyder

d. Install Python Libraries for machine learning and image processing.

The Anaconda Navigator is a free and user-friendly environment for the scientific Python language. It is depicted in (Figure 3.18). It offers the majority of the libraries necessary for creating deep learning models and gives the user access to Python coding tools. The codes were run and the results were visualized using spyder. An open-source web program called Spyder was utilized in this study to create models by fusing coding, results displays, and descriptive writings.

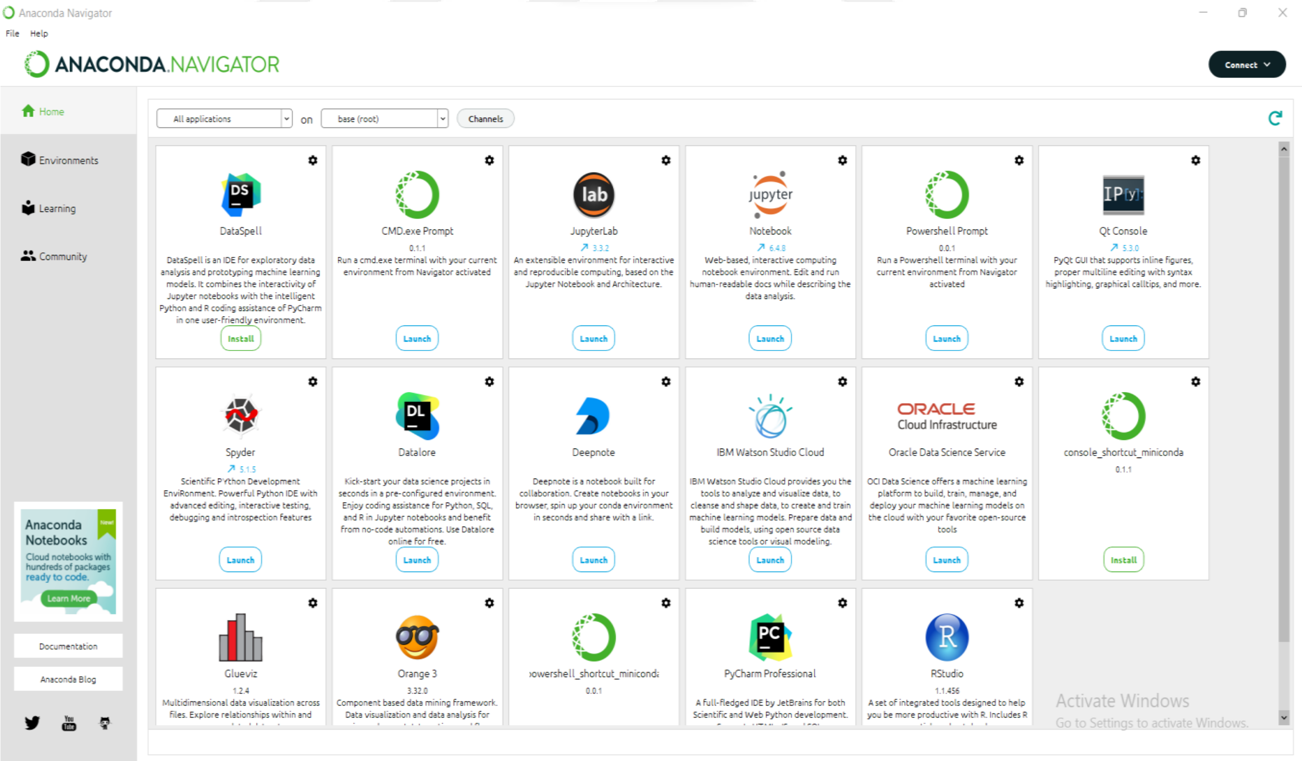


Figure 3.18: Anaconda Navigator Interface.

Python 3.9 was installed. Python is a programming language that is commonly used in machine learning applications because of its relative simplicity and the availability of many libraries to perform a variety of tasks. The large user base of this language is also helpful for debugging the program.

In order to do activities like machine learning, image processing, charting results, etc. without the need for specially written code, the Python libraries are installed, and the key libraries are displayed in (Figure 3.19). These libraries include:

a. Tensor Flow: is a free and open-source machine learning library. It can do a range of tasks. Nonetheless, it emphasizes the performance of deep neural models heavily.

b. Keras: is an open-source package that creates an interface for artificial neural networks and Python (serves as a TensorFlow library interface).

c. Scikit-learn (Sklearn): is a library for artificial intelligence. It incorporates a number of essential statistical modeling and machine learning techniques, including dimensionality reduction, regression, and classification.

d. OS module: The appropriate operating system interface features are enabled. The system files can be accessed through methods provided by the os and os.path modules.

e. OpenCV: A Python library for computer vision.

f. Matplotlib: is a sizable library that provides both static and interactive data charting.

g. Pandas: is a crucial library for information analysis.

h. NumPy: A library that manages matrices and multidimensional arrays and executes a variety of mathematical operations on them.

i. Mahotas: is a Haralick feature evaluation library for computer vision and image processing.

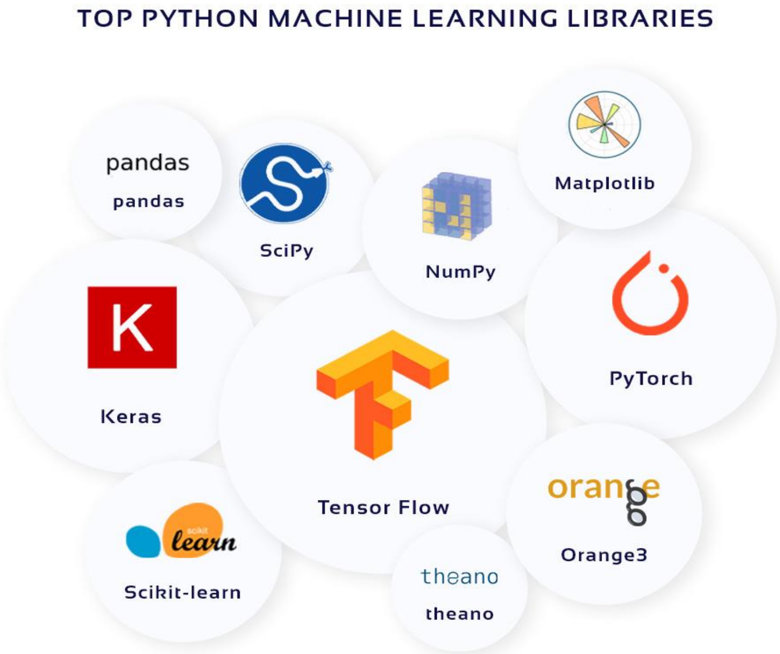


Figure 3.19: Main Python Libraries for Deep Learning and Image Processing.

3.8 METRICS FOR EVALUATING PERFORMANCE

3.8.1 Confusion Matrix

A prediction model's performance can be thoroughly understood using the confusion matrixIt gives a predictive performance in matrix format, including data on the proportion of accurately predicted cases, erroneously forecasted cases, and errors of both inaccurate and accurate prediction instances [22]. as shown in (Figure 3.20).

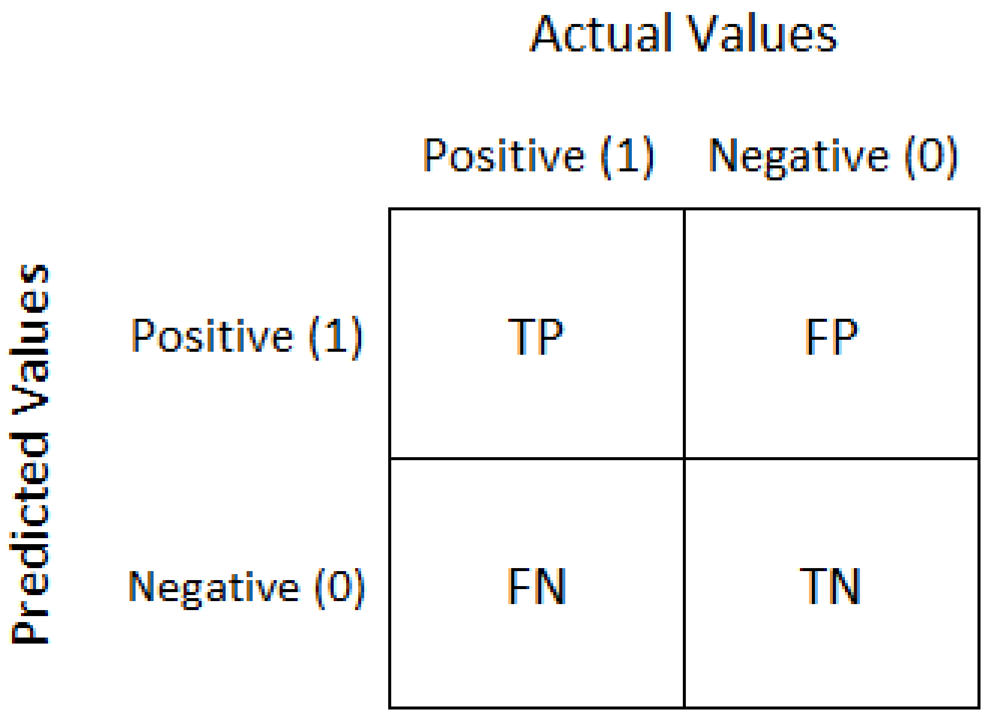


Figure 3.20: Confusion Matrix.

TP (True positive): Both the observation and the prediction are positive.

FN (False Negative): Positive observation with an anticipated negative outcome.

FP (False Positive): Both the observation and the prediction are negative.

TN (True Negative): A negative observation with a positive prediction.

3.8.1.1 Accuracy

The percentage of all successfully predicted classes to the actual classes in the dataset represents the accuracy of the prediction algorithm. Equation (3.3) is used to calculate the model's accuracy. Typically, any prediction model produces four separate results: true positive (TP), true negative (TN), false positive (FP), and false-negative (FN) [41].

$$Accuracy = \frac{TP + TN}{TP+TN+FN+FP} \tag{3.3}$$

3.8.1.2 Precision

It serves as a metric to show how accurate a positive forecast is. [52]. Equation (3.4) is utilized to calculate it.

$$Precision= \frac{TP}{TP+FP} \tag{3.4}$$

3.8.1.3 Recall

It shows the percentage of actual positive cases that our algorithm was able to predict accurately. [49]. The formula for calculating it is given below:

$$Recall =\frac{TP}{TP+FN} \tag{3.5}$$

3.8.1.4 F1-score

Comparing two models with contrasting characteristics, such as low accuracy and high recall, is difficult. We utilize the F-score to contrast them. The F1-score facilitates the simultaneous measurement of precision and recall. The following is the formula (3.6) used to determine it:

$$F1-Score=\frac{2*Recall*Precision}{Recall+Precision} \tag{3.6}$$

3.9 SUMMARY

This chapter covered the methodology, theoretical background, and approach that the study took. We thoroughly discussed each step. studying the best method for splitting the data into an 80/20 training to test ratio. After that, classified the features that were extracted, based on the models, and the outcomes are covered in the chapter that follows.

4. RESULTS

This chapter explains the findings of the tests and experiments conducted on the finished models. These models were tested with Python 3.9 and Anaconda3 (Spyder). The models were trained and tested using the two classes (COVID-19 and Non-COVID-19), with an 80% – 20% train–test split. 80% of the images were used for training (11350 images), and 20% of the images (2838 images) were utilized for testing. To evaluate the effectiveness of the models, the true positive (TP), true negative (TN), false positive (FP), and false-negative (FN) values were computed using the confusion matrix and the accuracy precision.

4.1 MODELS PERFORMANCE

In this work, two models using VGG16 features and GLCM features were classified using five machine learning techniques.

4.1.1 Machine Learning Algorithms Based On GLCM Features (Model1)

After Calculating the GLCM and Haralick features, we used machine learning algorithms to classify based on extracted features, then the result is shown below.

a. Decision Tree

The system predicted 180 Covid-19 scans and 219 non-Covid-19 scans incorrectly, while successfully predicting 1195 Covid-19 scans and 1244 non-Covid-19 scans based on the matrix's results, as seen in (Figure 4.1).

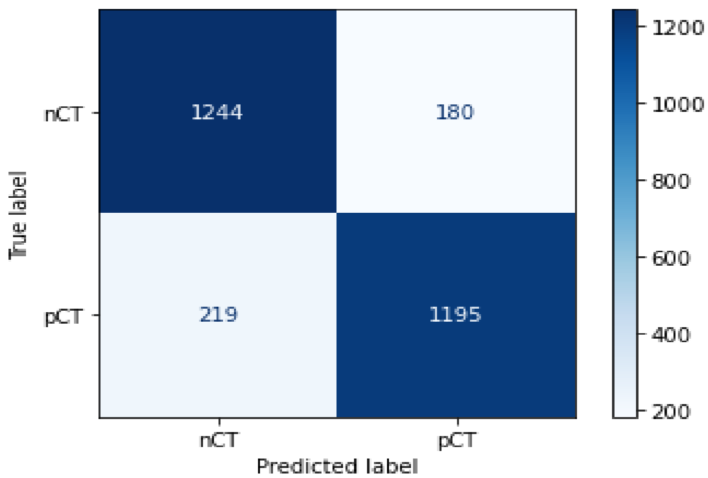


Figure 4.1: Confusion Matrix of Decision Tree.

b. Support vector machine

In accordance with the matrix's findings in (Figure 4.2), the algorithm accurately predicted 1101 Covid-19 images, and 1245 non-Covid-19 images, and forecasted 179 Covid-19 images and 313 non-Covid-19 images incorrectly.

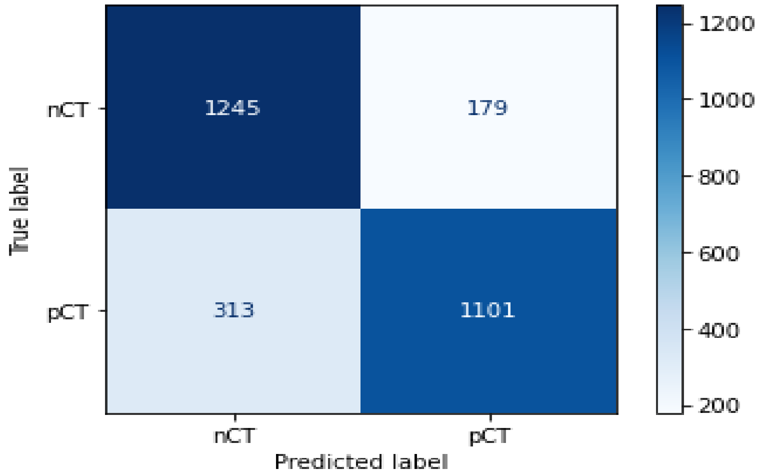


Figure 4.2: Confusion Matrix of Support Vector Machine.

c. K-Nearest neighbor

The algorithm accurately predicted 1153 Covid-19 photos, and 1109 non-Covid-19 photos, while making incorrect predictions for 315 Covid19 photos and 261 non-Covid-19 photos. These results are based on the matrix in (Figure 4.3).

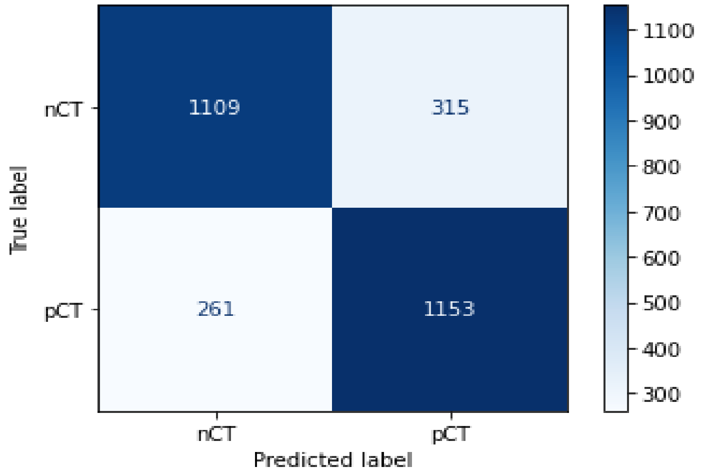


Figure 4.3: Confusion Matrix of K-Nearest Neighbor.

d. Random Forest

Based on the matrix's results, the system accurately predicted 1251 Covid-19 pictures and 1278 non-Covid-19 pictures, but forecasted wrongly 146 Covid-19 images and 163 non-Covid-19 images, according to (Figure 4.4).

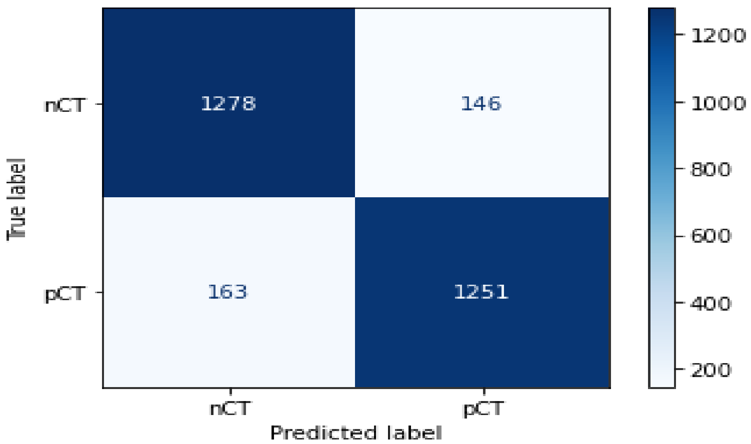


Figure 4.4: Confusion Matrix of Random Forest.

e. XGBoost

In accordance with the matrix's findings in (Figure 4.5), the algorithm accurately predicted 1311 Covid-19 images, and 1261 non-Covid-19 images, and forecasted 163 Covid-19 images and 103 non-Covid-19 images incorrectly.

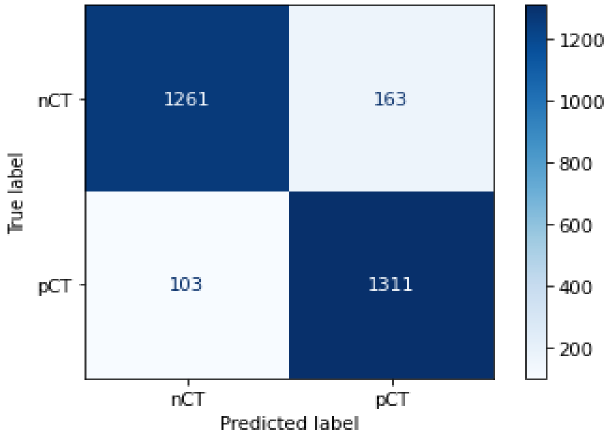


Figure 4.5: Confusion Matrix of XGBoost.

4.1.2 Machine Learning Algorithms Based On VGG16 Features (Model2)

a. Decision Tree

The system predicted 150 Covid-19 images and 172 non-Covid-19 images incorrectly, while successfully predicting 1242 Covid-19 images and 1274 non-Covid-19 images based on the matrix's results, as seen in (Figure 4.6).

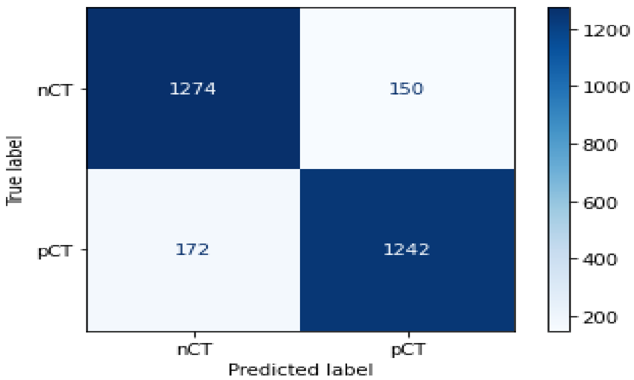


Figure 4.6: Confusion Matrix of Decision Tree.

b. Support vector machine

The algorithm accurately predicted 1382 Covid-19 images, and 1391 non-Covid-19 images, while making incorrect predictions for 33 Covid19 images and 32 non-Covid-19 images. These results are based on the matrix in (Figure 4.7).

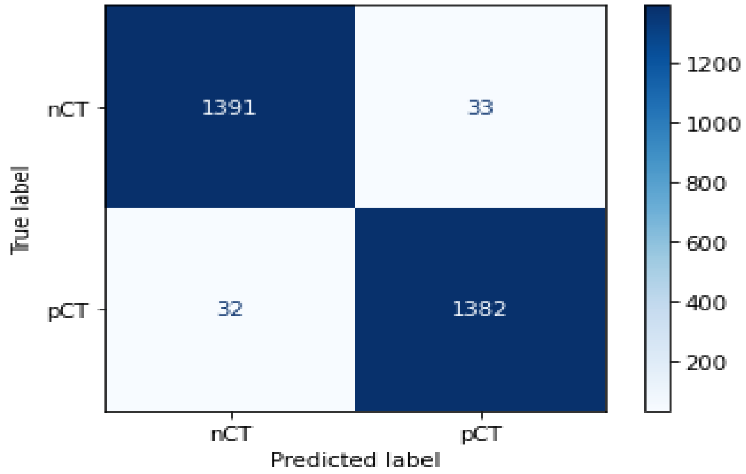


Figure 4.7: Confusion Matrix of SVM.

c. K-Nearest neighbor

The system predicted 77 Covid-19 photos and 14 non-Covid-19 photos incorrectly, while successfully predicting 1400 Covid-19 photos and 1347 non-Covid-19 photos based on the matrix's results, as seen in (Figure 4.8).

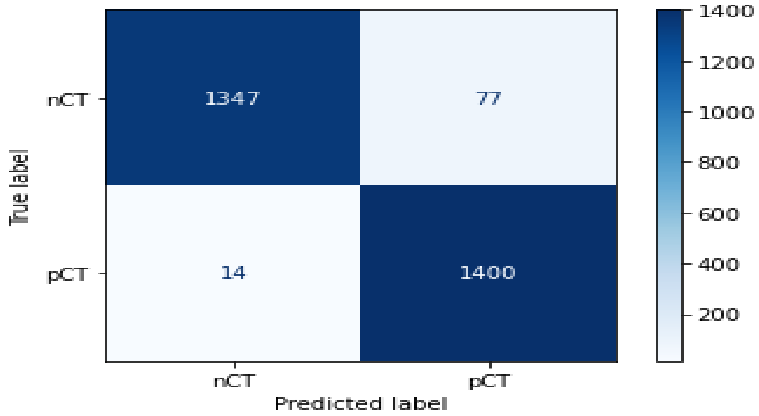


Figure 4.8: Confusion Matrix of KNN.

d. Random Forest

The system predicted 97 Covid-19 images and 111 non-Covid-19 images incorrectly, while successfully predicting 1303 Covid-19 images and 1327 non-Covid-19 images based on the matrix's results, as seen in (Figure 4.9).

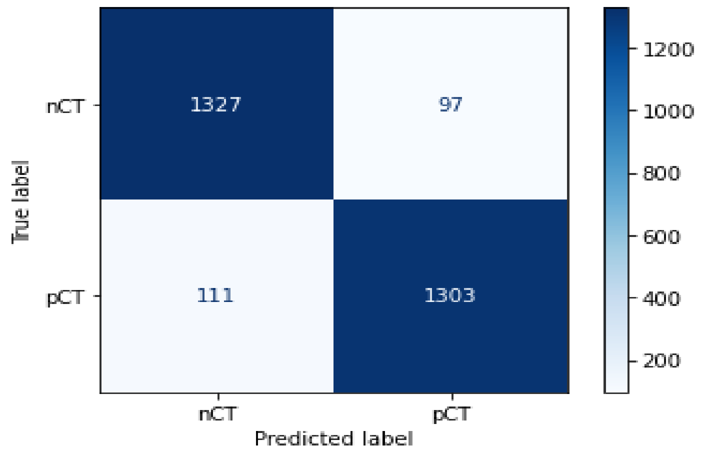


Figure 4.9: Confusion Matrix of Random Forest.

e. XGBoost

The system predicted 65 Covid-19 images and 17 non-Covid-19 images incorrectly, while successfully predicting 1397 Covid-19 images and 1359 non-Covid-19 images based on the matrix's results, as seen in (Figure 4.10).

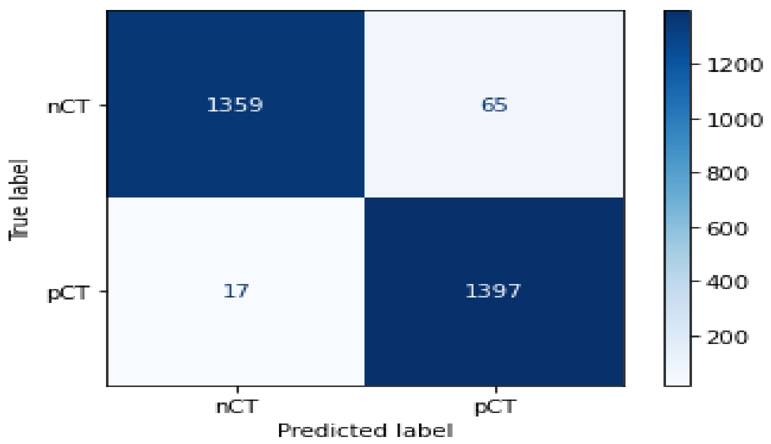


Figure 4.10: Confusion Matrix of XGBoost.

In Table (4.1), we show a comparison of the confusion matrix between the two models. Results from the VGG feature rating outperformed the results of the GLCM feature rating.

Table 4.1: Confusion Matrix Comparison For Two Models Using Five Algorithms.

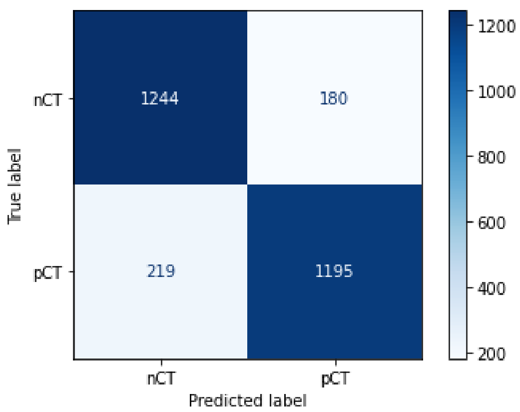
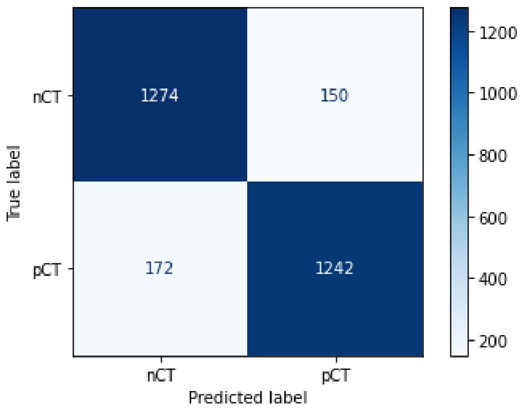
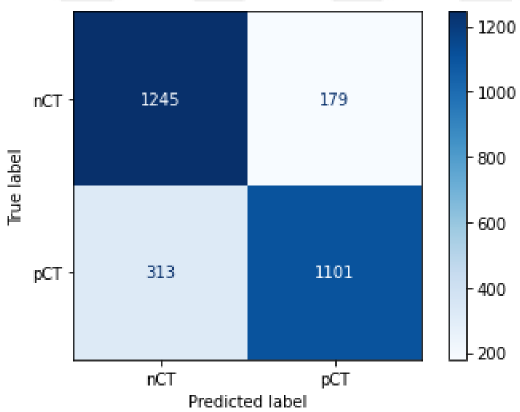
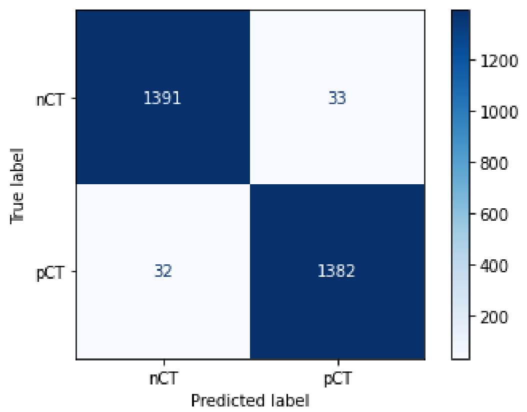
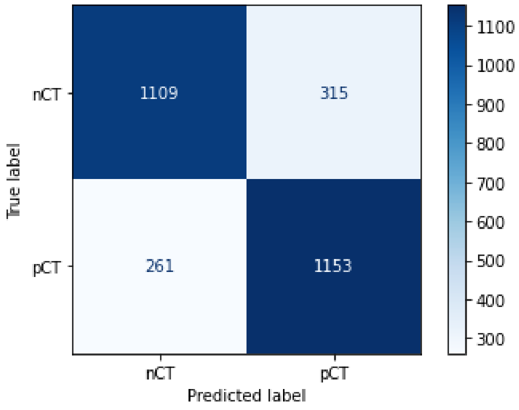
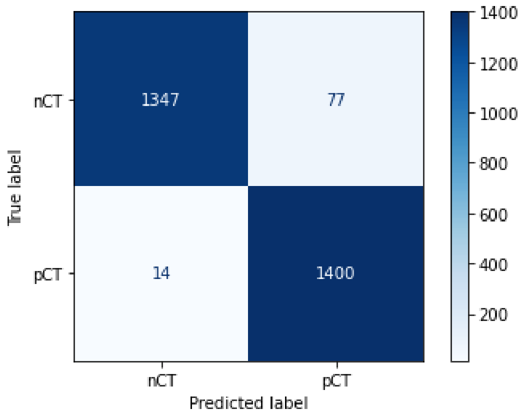
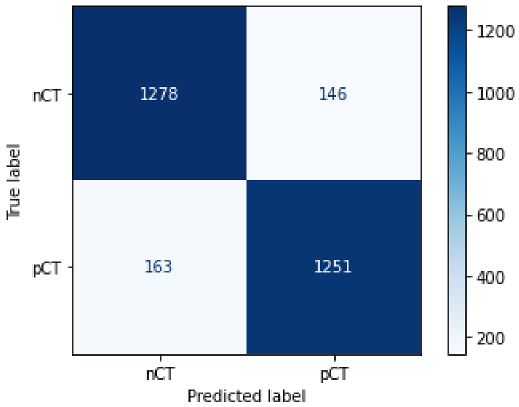
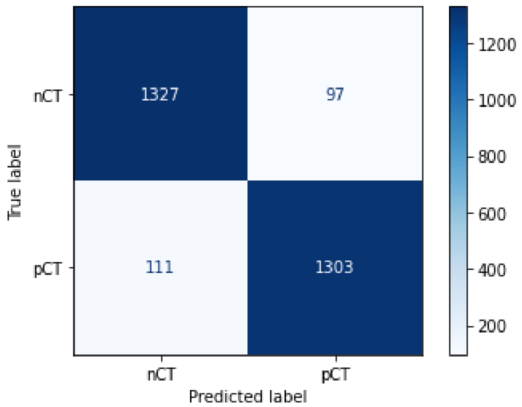
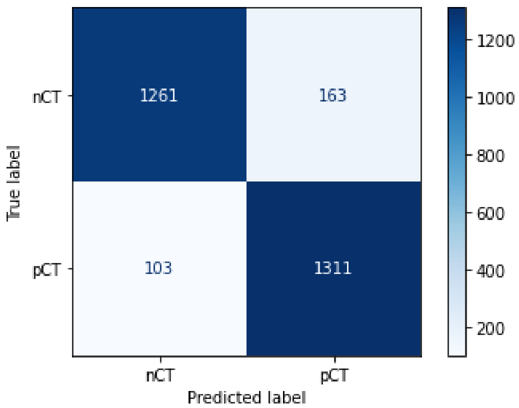
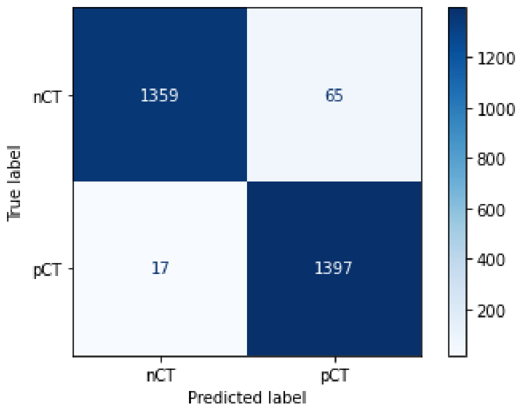
Algorithms	GLCM model	VGG16 model
DT		
SVM		

Table 4.1: Confusion Matrix Comparison For Two Models Using Five Algorithms "Table Continued".

<table>
<tr><td>KNN</td><td>
</td><td>
</td></tr>
<tr><td>Random
Forest</td><td>
</td><td>
</td></tr>
<tr><td>XGBoost</td><td>
</td><td>
</td></tr>
</table>

In Table (4.2), which compares the precision, recall, accuracy, and f1 score of several classifiers based on GLCM features, the XGBoost technique stands out with a classification accuracy of 91.98%.

Table 4.2: Precision, Recall, Accuracy, and F1 Score of Different Classifiers on GLCM Features.

No.	Classifier	Precision (%)	Recall (%)	F-score (%)	Accuracy (%)
1	Decision Tree	87.11	85.90	86.33	86.58
2	XGBoost	91.75	90.99	92.00	91.98
3	Random forest	90.21	88.87	89.43	89.87
4	SVM	84.69	84.22	82.78	83.97
5	KNN	79.90	79.99	80.51	80.99

The precision, recall, accuracy, and f1 score of several classifiers based on VGG16 features are compared in Table (4.3), The support vector machine method shines at 98.22% classification accuracy and produces findings in a very short time frame.

Table 4.3: Precision, Recall, Accuracy, and F1 Score of Different Classifiers on VGG16 Features.

No.	Classifier	Precision (%)	Recall (%)	F-score (%)	Accuracy (%)
1	Decision Tree	89.43	90.33	90.01	89.55
2	XGBoost	97.77	97.12	96.99	97.72
3	Random forest	94.22	93.99	92.21	93.12
4	SVM	98.34	97.90	98.66	98.22
5	KNN	97.44	97.70	96.98	96.99

The VGG feature rating produced better outcomes than the GLCM feature rating. 98.22% accuracy was attained using the SVM method and VGG16 features.

4.2 SUMMARY

The results of the tests and experiments on the completed models are explained in this chapter. Based on the features of GLCM and VGG16, we identified chest CT images using five machine learning algorithms (SVM, KNN, RF, DT, and XGBoot). Finally, we compared the outputs of the algorithms to establish which was more accurate. The findings demonstrated that, in SVM algorithms, VGG16 features attained an accuracy of 98.22% while GLCM produced less precise outcomes. The GLCM algorithm from XGBoost has the highest accuracy 91%.

5. DISCUSSION AND CONCLUSIONS

Future technologies like artificial intelligence and machine learning can accelerate the development of businesses across multiple industries. The Covid-19 dataset (14486 samples) from the Kaggle website was used for this work, and a variety of machine learning approaches were employed. The results showed how important it is to use machine learning approaches to minimize the challenges that COVID-19 poses for the healthcare sector. It also showed how a trained algorithm might determine if a person is healthy or infected with the coronavirus.

In order to create five well-known machine learning classifiers (SVM, KNN, RF, XGBoost, and DT), features were extracted using VGG16 and GLCM techniques. The performances of the classifiers were calculated using the metrics Accuracy, Recall, F1-Score, Precision, and confusion matrix. The study evaluates the classification success of different machine learning techniques. In terms of grading metrics, we discovered that SVM and VGG16 features performed better than the other classifiers. This study also found that a feature significantly affects how well a model performs. The study's findings promote the creation of more precise diagnosis and detection systems for controlling the coronavirus pandemic as well as the emergence of reliable and efficient imaging information diagnostic equipment that is integrated with clever strategies to assist medical professionals when the epidemic spreads.

Because we analyzed two groups of features with multiple methods, our research stands apart from the competition. This group was not compared in earlier research, and it is distinguished by the abundance of data.

5.1 FUTURE STUDY

A future study will also take lung problems into account, it might group many lung conditions such as pulmonary tuberculosis, asthma, and lung cancer with Covid-19. A bigger dataset will be utilized. Additionally, the created model might be tested using additional COVID-19 scans taken using different modalities (such as X-ray or MRI), or even multimodalities (CT scans and MRI) GLCM and VGG16 properties can also be mixed and use the same methods for classification.

REFERENCES

[1] M. Gour and S. Jain, “Stacked Convolutional Neural Network for Diagnosis of COVID-19 Disease from X-ray Images,” Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.13817

[2] S. Osler, *Coronavirus Outbreak: All the secrets revealed about the Covid-19 pandemic. A complete rational guide of its Evolution, Expansion, Symptoms and First Defense*. AB Lawal, 2020.

[3] G. Rong, Y. Zheng, Y. Chen, Y. Zhang, P. Zhu, and M. Sawan, “COVID-19 diagnostic methods and detection techniques: a review,” *Ref. Modul. Biomed. Sci.*, 2021.

[4] J. F. Abdulkareem and H. K. Aljobouri, “Chest CT images analysis with deep learning algorithms for COVID-19 diagnostic for Iraqi center,” in *AIP Conference Proceedings*, 2023, vol. 2414, no. 1, p. 60004.

[5] F. Chollet, *Deep learning mit python und keras: das praxis-handbuch vom entwickler der keras-bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.

[6] B. Wicht, “Deep learning feature extraction for image processing thesis,” *Phd, Dep. Informatics, Univ. Fribg.*, 2017.

[7] T. Beysolow II, *Introduction to deep learning using R: A step-by-step guide to learning and implementing deep learning models using R*. Apress, 2017.

[8] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, *Advances in deep learning*. Springer, 2020.

[9] C. A. Charu, “Neural networks and deep learning: a textbook.” Spinger, 2018.

[10] M. Paterna, “Timbre modification using deep learning.” MA thesis. Copenhagen, Denrmak: Department of Architecture, Design, and …, 2017.

[11] G. Polat, “Classification of lung nodules in CT images using convolutional neural networks.” Middle East Technical University, 2018.

[12] F. Farhadi, *Learning activation functions in deep neural networks*. Ecole Polytechnique, Montreal (Canada), 2017.

[13] L. B. Godfrey, *Parameterizing and aggregating activation functions in deep neural networks*. University of Arkansas, 2018.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[15] R. Venkatesan and B. Li, *Convolutional neural networks in visual computing: a concise guide*. CRC Press, 2017.

[16] M. Umberto, "Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks," *Appl. Deep Learn. Berkeley, California, USA Apress*, pp. 185–216, 2018.

[17] M. A. Uthaib and M. S. Croock, "Multiclassification of license plate based on deep convolution neural networks," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, p. 5266, 2021.

[18] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annu. Rev. Biomed. Eng.*, vol. 19, pp. 221–248, 2017.

[19] T. H. Davenport, T. Hongsermeier, and K. A. Mc Cord, "Using AI to improve electronic health records," *Harv. Bus. Rev.*, vol. 12, pp. 1–6, 2018.

[20] E. Kuo, "AI in healthcare: industry landscape," *Techburst Rep.*, 2017.

[21] B. Meskó, "FDA approvals for smart algorithms in medicine in one giant infographic. The Medical Futurist. 2019." 2019.

[22] Ü. Yavuz and M. N. Dudak, "Classification of covid-19 dataset with some machine learning methods," *J. amasya Univ. Inst. Sci. Technol.*, vol. 1, no. 1, pp. 30–37, 2020.

[23] A. Shamsi *et al.*, "An uncertainty-aware transfer learning-based framework for COVID-19 diagnosis," *IEEE Trans. neural networks Learn. Syst.*, vol. 32, no. 4, pp. 1408–1417, 2021.

[24] P. K. Sethy, S. K. Behera, K. Anitha, C. Pandey, and M. R. Khan, "Computer aid screening of COVID-19 using X-ray and CT scan images: An inner comparison," *J. Xray. Sci. Technol.*, vol. 29, no. 2, pp. 197–210, 2021.

[25] M. A. Ahmed, Z. T. Al-qaysi, M. L. Shuwandy, M. M. Salih, and M. H. Ali, "Automatic COVID-19 pneumonia diagnosis from x-ray lung image: A Deep Feature and Machine Learning Solution," in *Journal of Physics: Conference Series*, 2021, vol. 1963, no. 1, p. 12099.

[26] A. A. S. MOL and M. K. Sabu, "CLASSIFICATION OF COVID-19 PATIENT'S CHEST X-RAY IMAGES USING MACHINE LEARNING TECHNIQUES–A COMPARATIVE ANALYSIS," *Adv. Appl. Math. Sci.*, pp. 3017–3026, 2021.

[27] M. A. Khan *et al.*, "COVID-19 case recognition from chest CT images by deep learning, entropy-controlled firefly optimization, and parallel feature fusion," *Sensors*, vol. 21, no. 21, p. 7286, 2021.

[28] O. Çinare and M. YaĞAnoĞLu, "Determination of Covid-19 Possible Cases by Using Deep Learning Techniques," *Sak. Univ. J. Sci.*, vol. 25, no. 1, pp. 1–11, 2021.

[29] S. V Kogilavani *et al.*, "COVID-19 detection based on lung CT scan using deep learning techniques," *Comput. Math. Methods Med.*, vol. 2022, 2022.

[30] A. S. Mubarak, S. Serte, F. Al-Turjman, Z. S. Ameen, and M. Ozsoz, "Local binary pattern and deep learning feature extraction fusion for COVID-19 detection on computed tomography images," *Expert Syst.*, vol. 39, no. 3, p. e12842, 2022.

[31] S. Ankalaki, K. Shorya, and J. Majumdar, "Deep Transfer Learning Model-Based Automated Detection of COVID-19 from X-ray Images and Interpretation of COVID-19 Images Using GLCM Texture Features," in *Emerging Research in Computing, Information, Communication and Applications: ERCICA 2020, Volume 2*, 2022, pp. 581–598.

[32] E. A. Abbood and T. A. Al-Assadi, "GLCMs Based multi-inputs 1D CNN Deep Learning Neural Network for COVID-19 Texture Feature Extraction and Classification," *Karbala Int. J. Mod. Sci.*, vol. 8, no. 1, pp. 28–39, 2022.

[33] B. Jehangir and S. R. Nayak, “GLCM Based LGBM classification for COVID-19 Analysis using Chest X-ray Images,” in *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2022, pp. 477–481.

[34] L. Kong and J. Cheng, “Classification and detection of COVID-19 X-Ray images based on DenseNet and VGG16 feature fusion,” *Biomed. Signal Process. Control*, vol. 77, p. 103772, 2022.

[35] O. Jung, D. U. Kang, G. Kim, and S. Y. Chun, “Adaptive GLCM sampling for transformer-based COVID-19 detection on CT,” *arXiv Prepr. arXiv2207.01520*, 2022.

[36] G. A. Riyanto, “Classification Covid-19 Based on X-Ray Using GLCM and ANN Backpropagation,” *J. Transform.*, vol. 19, no. 2, 2022.

[37] D. Kuzinkovas and S. Clement, “The Detection of COVID-19 in Chest X-Rays Using Ensemble CNN Techniques,” *medRxiv*, pp. 2011–2022, 2022.

[38] M. Maftouni. Large COVID-19 CT scan slice dataset [Online] Available: https://www.kaggle.com/dsv/2321803

[39] P. Lubaib and K. V. A. Muneer, “The heart defect analysis based on PCG signals using pattern recognition techniques,” *Procedia Technol.*, vol. 24, pp. 1024–1031, 2016.

[40] P. Theerthagiri, I. J. Jacob, A. U. Ruby, and Y. Vamsidhar, “Prediction of COVID-19 possibilities using KNN classification algorithm,” 2020.

[41] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[42] N. K. Sinha, M. Khulal, M. Gurung, and A. Lal, “Developing a web based system for breast cancer prediction using xgboost classifier,” *Int. J. Eng. Res. Technol.*, vol. 9, 2020.

[43] D. Gadkari, “Image quality analysis using GLCM,” 2004.

[44] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: analysis, applications, and prospects,” *IEEE Trans. neural networks Learn. Syst.*, 2021.

[45] P. D. Barua *et al.*, “Automatic COVID-19 detection using exemplar hybrid deep features with X-ray images,” *Int. J. Environ. Res. Public Health*, vol. 18, no. 15, p. 8052, 2021.

[46] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 international conference on engineering and technology (ICET)*, 2017, pp. 1–6.

[47] E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos, and P. de Geus, “Malicious software classification using VGG16 deep neural network’s bottleneck features,” in *Information Technology-New Generations: 15th International Conference on Information Technology*, 2018, pp. 51–59.

[48] A. A. Purwita and N. N. Qomariyah, “Impact of Aligning Saliency Maps on COVID-19 Disease Detection Using Chest X-Ray Images,” in *2021 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2021, pp. 396–401.