COVID-19 detection using chest X-ray images based on Machine learning and Deep learning models: Further evidence from Data Augmentation

Submitted in partial fulfilment of the

requirements of the degree of

Master of Data Science

Shokoufeh Hosseini

(Student Number: 116276)

Supervisor: Associate Prof. Habib Ullah

Co-Supervisor: Associate Prof. Fadi Al Machot

Faculty of Science and Technology

Norwegian University of Life Science

May 2023

Declaration

This written submission represents my own ideas in my own words, and where others' ideas or phrases have been used, the original sources have also been appropriately cited. Furthermore, I declare that my submission complies with all academic integrity and honesty principles and does not contain any misrepresentation or fabrication of any idea, fact, or data. The university can take disciplinary action against those who violate the rules. This action can also elicit penalties from sources without adequately cited or whose permissions would have been obtained.

Shokoufeh Hosseini

Date: May 14, 2023

CERTIFICATE

It is certified the work contained in the thesis titled " COVID-19 detection using chest X-ray images based on Machine Learning and Deep Learning models: Further evidence from Data Augmentation" by " Shokoufeh Hosseini" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Habib Ullah
Associate Professor,
Dept. of Science and Technology
Norwegian University of Life Science

Dr. Fadi Al Machot
Associate Professor,
Dept. of Science and Technology
Norwegian University of Life Science

Date: May 14, 2023

Place: Norwegian University of Life Science

ACKNOWLEDGMENTS

I am delighted to express my respect and a deep sense of gratitude to my Master supervisor Habib Ullah & my Co-supervisor Fadi AL Machot Associate Professors, Department of REALTEK for their wisdom, vision, expertise, guidance, enthusiastic involvement, and persistent encouragement during the planning and development of this research work. I also gratefully acknowledge their painstaking efforts in thoroughly reviewing and improving the manuscripts, without which this work could not have been completed. I am incredibly thankful to my husband Sajad for his knowledge, love, and encouragement during this journey.

I wish to express my appreciation to my friends and thanks to the research fellows at the department for their help and motivation throughout my research work. I also would like to express my deep and sincere gratitude to my friends and all other persons whose names do not appear here. These people have directly or indirectly helped me in good times and hard times.

Finally, I am indebted and grateful to the Almighty for helping me in this endeavor.

Shokoufeh Hosseini

ABSTRACT

Detecting COVID-19 Coronavirus quickly and accurately was essential for preventing and controlling this pandemic through timely quarantine and medical treatment in the absence of vaccines. Because there were an increasing number of cases of COVID-19 worldwide and the limited number of detection kits available, it was difficult to identify the presence of this disease. Consequently, we needed to seek other alternatives at this time. Deep learning techniques in computer-aided medical diagnosis have surpassed state-of-the-art performance in computer-aided diagnosis among existing, widely accessible, and low-cost resources. This dissertation proposes an alternative diagnostic tool that utilizes available resources and advanced deep-learning techniques to detect COVID-19 cases. We used two sources of datasets in this study, one of which was a small dataset [1], and another was a large dataset [2] with X-ray images of normal, COVID-19, and viral pneumonia. Using chest X-ray images, we investigated the performance of COVID-19 detection using machine learning and deep learning methods. We extracted edge and pixel values using feature extraction and then used Random Forest and Support Vector Machine to classify the images. Furthermore, we used data augmentation (Brightness, Contrast, and Flipping) to deal with small data sets. We also used this method for large data sets after reducing the number of images due to an imbalanced dataset. In addition, we used VGG19, CNN, and Convolutional Auto Encoder for deep learning models to extract features and classify COVID-19 chest X-ray images. We used hyperparameter tuning and applied transfer learning to further improve these models. Results showed a significant improvement in the accuracy of the models, demonstrating the effectiveness of our methods. By calculating the accuracy of these models, we found that convolutional autoencoder and CNN models performed best. Overfitting, however, can impair models' generalizability and predictability, reducing their generalizability. There were times when we reduced this problem, but there were also times when we couldn't handle it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this chapter is to provide the background and motivation for the current research. What inspired us to do this research is explained in this part.

1.1 Background

An infectious disease caused by SARS-CoV-2 viruses is Coronavirus disease (COVID-19) [3]. This disease started spreading from Wuhan, a city in China, at the end of 2019 [4]. This illness has created an urgent situation throughout the world. As a result, the world decided to cope with this problem and take urgent action against this problem. The first problem to solve is diagnosing the disease as soon as possible. Diagnostic methods can be used in many ways, but X-ray images of the lungs are one of the fastest. Some studies used this approach to detect COVID-19 by combining deep learning and machine learning. We have extended and investigated other approaches to detecting COVID using deep learning and machine learning. In Figure 1.1 examples of images used in this study have been provided.

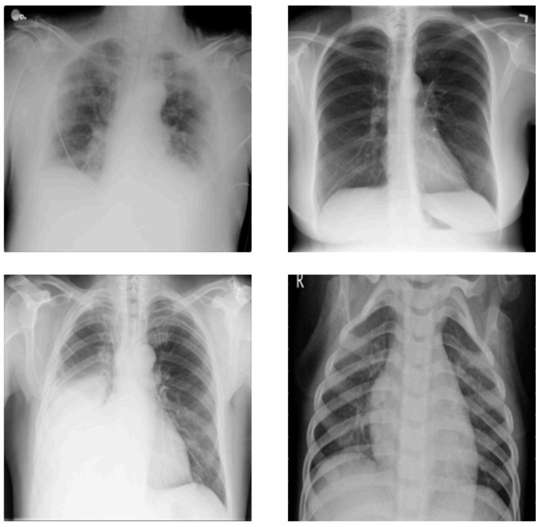


Figure 1.1: Examples of X-ray images used in this research and for modeling.

1.2 Motivation for the present research work

There have been millions of deaths caused by the COVID-19 pandemic, resulting in hundreds of thousands of deaths, and the global economy has been left adrift. It is crucial to detect and diagnose COVID-19 cases as soon as possible in order to manage and control the disease effectively. A chest X-ray is one of the diagnostic tools for COVID-19, which can provide valuable information about the severity and progression of the disease, as well as possible treatment options. This research aimed to explore the application of machine learning models and deep learning models to the detection of COVID-19, utilizing chest X-ray images by applying machine learning models and deep learning models. Medical imaging has seen remarkable success with using of machine learning and deep learning models [5]. These models include the detection and diagnosis of a wide variety of diseases. As a result of these models, we will be able to improve the way we approach medical diagnosis, enabling faster and more accurate diagnosis and ultimately improving patient outcomes. COVID-19 must be prevented from spreading, and timely treatment to patients infected with the disease; it is critical to provide rapid and accurate diagnoses to prevent the spread of the disease. As a diagnostic tool for respiratory diseases, such as COVID-19, chest X-ray imaging is one of the most widely used. Nevertheless, interpreting chest X-ray images is a complex task that requires the expertise of a radiologist in order to be successful. In case of an early-stage disease or poor image quality, the interpretation of chest X-rays can also be subjective and subject to error. It is thought that machine learning and deep learning models can help overcome these limitations by automating and objectively analyzing chest X-ray images, thereby improving diagnoses of COVID-19. It has been widely accepted that machine learning models such as Random Forest [6] and Support Vector Machines (SVM) [7] have been used in medical image analysis for many years, and both have shown promise in the analysis of medical images. Typically, these types of models work based on the concept of supervised learning, where the model is trained on a set of labeled data in order to learn the underlying patterns and relationships between input features and the labels assigned to the output features. In this way, the trained model can be used to predict the outcome of new, unknown data in the future. In the case of COVID-19 detection, using chest X-ray images, random forest, and SVM models can be trained on a dataset of labeled images, where each image is labeled as either COVID-19 positive or negative. The models can then learn to recognize the patterns and features that distinguish COVID-19 positive images from negative ones. Once the models are trained, they can be used to predict the COVID-19 status of new chest X-ray images. Deep learning models, particularly Convolutional Neural Networks (CNNs), have shown success in medical image analysis, including the detection and diagnosis of various diseases. CNNs are particularly suited for image analysis tasks, as they can learn complex features and patterns from the input images without requiring manual feature engineering [8]. In this research, we will explore the application of two CNN models for COVID-19 detection using chest X-ray images: VGG19 and a custom CNN. VGG19 is a widely used CNN model that has shown excellent performance in various image recognition tasks [9]. A custom CNN model can be designed specifically for COVID-19 detection, where the architecture and parameters are optimized for this task. A Convolutional Auto Encoder (CAE) is a type of neural network that can be used for image compression and reconstruction. CAEs are composed of two parts: a compression module that compresses an input image to a low-dimensional latent space and a decoder that reconstructs the original image from the latent space representation. CAEs can be trained on a dataset of chest X-ray images, where the encoder learns to capture the essential features of the images, and the decoder learns to reconstruct the images from the latent space representation [10]. In the case of COVID-19 detection using chest X-ray images, a CAE can be trained on a dataset of labeled images, where the encoder learns to compress the essential features that distinguish.

1.3 Organization of the thesis

The research work presented in the thesis is organized and structured in the form of seven chapters, which are briefly described as follows:

i) **Chapter 1** describes the introduction and motivation of the presented paper

ii) **Chapter 2** Provides a comprehensive review of relevant literature

iii) **Chapter 3** Information about datasets we have used and data augmentation presented in this study

iv) **Chapter 4** A description of the algorithm and methods used in this paper is presented

v) **Chapter 5** Analyzes the experimental results of the models used in this study and compares them

vi) **Chapter 6** Provides a Discussion section about the results of this study and the challenges it faced

vii) **Chapter 7** concludes the thesis with overall discoveries of the present research work. The scope for future work is also mentioned.

Chapter 2

Literature Review

The objective of this chapter is to present a survey of different papers and articles that have attempted to solve the COVID-19 detection problem using machine learning algorithms or deep learning algorithms. In order to apply Artificial Intelligence (AL) methods, they have used different approaches. Methods that involve traditional machine learning, deep learning, and hybrid methods can all be divided into three categories.

2.1 Traditional Machine Learning Models

Malathy Jawahar [11] developed a model that highly accurately analyzed chest X-ray images to detect COVID-19. Histogram Oriented Gradient (HOG) [12] was used to extract features, and the Gray-Level Co-Occurrence Matrix (GLCM) [13] was used to evaluate this extraction. These extracted features were then trained with Random Forest classification. The accuracy of this model was assessed against other traditional machine learning classifiers, including SVM, linear regression, and K-nearest Neighbors (KNN) [14].

Cesar Ortiz-Toro [15] incorporated three machine learning algorithms: Random Forest [16], K-Nearest Neighbors [14], and Support Vector Machines [17]. In this study, a dataset was constructed using images extracted from a repository of images developed primarily for the study of COVID-19 and based on pediatric chest X-rays. The tested methods proved to be reliable and easy-to-use auto diagnostic tools.

Abhishek Dixitn [18] proposed a 3-step procedure that includes K-means clustering [19] and feature extraction during the pre-processing stage. The selected features are optimized in the second step using a hybrid differential evolution algorithm [20] and particle swarm optimization [21]. Following the optimization of the features, the SVM classifier uses the features as input. According to their model, 8066 images of normal chest X-rays, 5551 images of pneumonia, and 358 images of COVID-19 chest X-rays achieved 99.34% accuracy. Their model is robust and sustainable in diagnosing individuals infected with COVID-19.

Hasoon [22] proposed a set of procedures including preprocessing (image noise removal, image thresholding, and morphological operation), identifying and segmenting regions of interest, obtaining features, analyzing local binary patterns (LBPs) [23], analyzing gradient histograms [12], and analyzing Haralick textures [24] and classifying data (K-Nearest

Neighbors (KNN) and Support Vector Machines (SVM)). As a result of the combination of feature extraction operators and classifiers, six models can be generated, including LBP-KNN [25], HOG-KNN [26], Haralick-KNN [27], LBP-SVM [28], HOG-SVM [29], and Haralick-SVM [30]. Test samples of 5,000 images were used to evaluate the six models, which were trained with 5-fold cross-validation. The results showed a high diagnostic accuracy of up to 98.66%. Based on its accuracy, sensitivity, specificity, and precision, the LBP-KNN [25] model outperforms the other models. In addition to providing an end-to-end structure without the need for manual feature extraction and manual selection methods, the proposed method for early detection and classification of COVID-19 by image processing using X-ray images has been shown to be usable as it provides an end-to-end structure for the detection and classification of COVID-19.

2.2 Deep Learning Based Methods

Mohamed Loey [31] proposed Convolutional neural networks (CNNs) for classifying COVID-19 artifacts in real-world situations based on chest X-ray images. Convolutional neural networks (CNNs) were proposed for recognizing chest X-ray images with Bayesian optimization [32]. Specifically, two components of the model were applied: the CNN for feature extraction and learning and the Bayesian optimizer for tuning its hyperparameters. In a large, balanced dataset, 10,848 images were analyzed (3616 COVID-19, 3616 normal cases, and 3616 pneumonia cases). The first study compared Bayesian optimization to three distinct scenarios; convergence charts and accuracy compared the two scenarios. Using Bayesian search [33] to generate optimal architectures, they found that 96% of their results were accurate. This comparison of research methods and theme analysis methods is designed to assist qualitative researchers in addressing their research questions methodologically.

Govardhan Jain [34] applied a transfer learning model using pre-trained Convolutional Neural Networks (CNNs) and a Residual Network architecture with 50 layers (ResNet50) [35] on 1215 chest X-ray images, then performed data augmentation to gain up to 1832 images to avoid overfitting and improve generalization. Training-validation-testing and five-fold cross-validation procedures were employed to demonstrate the method's effectiveness. The high accuracy of 97.77%, recall of 97.14%, and precision of 97.14% in the case of COVID-19 detection show the effectiveness of the proposed method.

Bhawna Nigam [36] developed Coronavirus diagnostic systems based on popular deep-learning architectures. This paper describes a set of architectures employed for the analysis: VGG16 [37], DenseNet121 [38], Xception [39], NASNet [40], and EfficientNet [41]. They used a multiclass classification that included COVID-19-positive patients, normal patients, and another class. Another type of X-ray is used to diagnose pneumonia, influenza, and other respiratory diseases. 79.01%, 89.96%, 88.03%, 85.03%, and 93.48% of the results were obtained for VGG16, DenseNet121, Xception, NASNet, and EfficientNet, respectively.

Harsh Panwar [42] proposed an algorithm based on deep transfer learning to detect COVID-19 cases using chest X-rays and CT scan images. Their research included three datasets: COVID-chest X-ray, SARS-COV-2 CT-scan, and Chest X-Ray Images (Pneumonia). According to the obtained results, the proposed deep learning model could detect COVID-19-positive cases in 2 seconds, which is faster than conventional RT-PCR tests. Additionally, they have established a relationship between COVID-19 patients and Pneumonia patients to investigate how Pneumonia radiology images correlate with COVID-19 radiology images. They used the Grad-CAM-based color visualization [43] approach in all experiments to interpret radiology images and take action for future comings.

Emtiaz Hussain [44] applied a CNN model called CoroDet [45] in this study to detect COVID-19 automatically on raw chest X-ray and CT scan images. This tool was designed to detect binary or multiclass models of COVID, normal and viral pneumonia, and bacterial pneumonia. COVID detection techniques were compared in terms of accuracy to the proposed model. The proposed model produced a classification accuracy of 99.1% for two classes, 94.2% for three classes, and 91.2% for four classes, clearly better than the state-of-the-art methods for COVID-19 detection. According to their knowledge, their method was evaluated using the largest dataset of x-ray images for COVID detection [46]. Shervin Minaee trained four convolutional neural networks using a subset of 2000 chest X-ray images, including ResNet18 [35], ResNet50 [35], SqueezeNet [47], and DenseNet-121 [38], to identify COVID-19 disease. The models were then tested on the remaining 3000 images, and most achieved a sensitivity rate of 98% while having a specificity rate of around 90%.

Shankar [48] provided a model that included preprocessing, feature extraction, and classification processes. Initially, images were preprocessed using Weiner filtering (WF) [49]. Following that, gray-level co-occurrence matrices, gray-level run-length matrices [50], and local binary patterns (LBP) were combined for the fusion-based feature extraction process. An optimal feature subset was then selected using the Salp Swarm Algorithm (SSA) [51]. An artificial neural network (ANN) was applied to classify infected and healthy patients.

Stefanos Karakanis [52] proposed deep-learning neural networks built on ResNet8 [35] and CNN models at the University of Aberdeen in the United Kingdom. Additionally, they tested their model bias using masked images from the dataset. Their study used a pre-trained ResNet8 model based on the ImageNet dataset and a CNN model without transfer learning for COVID detection.

Khabir Uddin Ahamed [53] developed a COVID-19 case detection model by analyzing chest CT scans and X-ray images. This deep learning model uses a modified ResNet50V2 architecture [35]. Data collected from a variety of publicly available sources were used to train the model. This data included four types of classes: confirmed COVID-19 cases, normal controls, and confirmed cases of bacterial and viral pneumonia. As part of the preprocessing, the aggregated dataset was preprocessed through a sharpening filter before being fed into the proposed model for further processing.

Sara Hosseinzadeh Kassani [54] evaluated popular deep-learning-based feature extraction algorithms for COVID-19 detection. A pool of deep convolutional neural networks was chosen among MobileNet [55], DenseNet [38], Xception [39], ResNet [35], InceptionV3 [56], InceptionResNetV2 [57], VGGNet [37], and NASNet [40] to obtain the most accurate feature. This is a crucial component of learning. Several machine learning classifiers were used to extract features from the extracted data to categorize subjects as COVID-19 cases or controls. For unseen data, this approach avoided task-specific data preprocessing methods. A publicly available chest X-ray and CT image dataset, COVID-19, was used to validate the proposed method. A hybrid ResNet50 [35] feature extractor trained by LightGBM outperformed a DenseNet121 [38] feature extractor with 98% accuracy. The second-best learner was a Bagging tree classifier with 99% accuracy.

Chaimae Ouchicha [58] introduced CVDNet, a deep convolution neural network (CNN) model for detecting COVID-19 infection in chest X-ray images. Based on residual neural networks, local and global input features are captured using two similar levels with different kernel sizes. The model was trained based on a dataset available on the internet that contained 219 COVID-19 chest x-ray images, 1341 normal chest x-ray images, and 1345 viral pneumonia chest x-ray images. As a result of their experiments, CVDNet demonstrated impressive classification performance on a small dataset, which can further be improved with more training data. On a small dataset, these results demonstrate promising classification performance. As a result, they believe their CVDNet model could serve as a valuable tool for radiologists to diagnose COVID-19 cases as early as possible.

Eduardo Luz [59] introduced new deep artificial neural network models that are highly accurate and have low footprints based on the EfficientNet family of deep neural networks. In the study, hierarchical classifiers were employed to take advantage of the underlying taxonomy. For training the proposed approaches and the other five competing architectures, X-ray images are divided into healthy, pneumonia-free, and COVID-19 patients. To evaluate the generalization power of the method, they proposed an open-dataset and cross-dataset evaluation. With 93.9% accuracy, 96.8% COVID-19 sensitivity, and a 100% positive prediction rate, the proposed approach produced a better model than the other tested approaches by using from 5 to 30 times fewer parameters. The cross-dataset evaluation indicated that even state-of-the-art models suffer from a lack of generalization power, which suggests that deep learning can assist physicians in detecting COVID-19 in X-ray images only with large and heterogeneous databases. COVIDx database [60] consists of 13,800 X-ray images, 183 of which are from patients with COVID-19. As far as efficiency and effectiveness are concerned, the reported results represent state-of-the-art results.Embedding a smartphone app into medical devices or even in physicians' equipment appears to be a promising option.

Sarra Guefrechi [61] developed a deep learning algorithm to detect COVID-19 in chest X-ray images using features extracted from the images. An enhanced dataset, comprising COVID-19 and standard chest X-ray images derived from several public databases, has been fine-tuned using three robust networks, including ResNet50, InceptionV3, and VGG16. Using data augmentation techniques such as Random Rotation at an angle between -10 and 10 degrees, random noise, and horizontal flips, they artificially generated many chest X-ray images. As a result of the proposed models, chest X-ray images were classified as Normal or COVID-19 with 97.20% accuracy for Resnet50 [35], 98.10% accuracy for InceptionV3 [56], and 98.30% accuracy for VGG16 [37]. COVID-19 detection methods were easily deployed and showed strong performance in the results, demonstrating the effectiveness of transfer learning.

2.3 Hybrid Methods

Aras M. Ismael [62] employed a deep-learning algorithm to classify COVID-19 and normal chest X-rays with the help of feature extraction, fine-tuning of pre-trained convolutional neural networks, and end-to-end training of a developed CNN model. Pretrained deep CNN models (ResNets with different layers, VGG16 and VGG19) were used for deep feature extraction. In the end, a Support Vector Machine (SVM) classifier was employed for deep feature classification. There were 180 chest X-rays from COVID-19 and 200 chest X-rays from normal X-rays used in this study for binomial classification.

Seyed Mohammad Jafar Jalali [63] developed an effective method to detect COVID-19 disease by applying a convolutional neural network (CNN) and combining it with the K-nearest neighbors at the end layer of the network, which increased the accuracy of the resulting method. This is done by replacing the last Softmax CNN layer with a K-nearest neighbors (KNN) classifier that takes into account the agreement of neighborhood labeling in order to replace the Softmax CNN layer.

Prottoy Saha [64] used a scheme called EMCNet to assess chest X-ray images to make an automated detection of COVID-19 patients that were suspected. X-ray images of patients were analyzed using a convolutional neural network so that more profound and higher-level features could be extracted. To detect COVID-19, binary classifiers such as Random Forest [16], Support Vector Machine [17], Decision Tree [65], and AdaBoost [66] were trained on the extracted features. An ensemble of classifiers was developed based on the outputs of these classifiers, ensuring better results for datasets of different sizes and resolutions.

Murugan Hemalatha [67] developed a hybrid Random Forest Deep Learning (HRFDL) [68] classifier based on Modified Heat Transfer Search (MOMHTS) [69]. CT scan images were used to identify COVID-19 and chest X-ray images were used to identify viral pneumonia. In this methodology, the objective is primarily to increase the speed at which IoT devices communicate as well as to improve COVID-19 detection by optimizing the HRFDL classifier to be suitable for resources that can handle minimal computation and storage requirements.

Mesut Togaçar [70] restructured the data classes by applying the FuzzyColor technique for the preprocessing step in which structured images were stacked with the original images. Then, the stacked dataset was trained with deep learning models such as MobileNetV2 [71] and SqueezeNet [47]. The features gained by the models were processed using the Social Mimic optimization method. Afterward, efficient features were combined and classified using Support Vector Machines (SVM).

Saddam Hussain Khan [72] presented two fresh approaches to deep learning frameworks: Deep Hybrid Learning (DHL) [73] and Deep Boosted Hybrid Learning (DBHL) [74] for COVID-19 detection in chest X-ray datasets. Through a machine learning (ML) classifier attached to a proposed DHL framework, the representation learning ability of two developed COVID-RENet-1 and 2 models is exploited separately, combined through a machine learning (ML) framework. They ensured that Region and Edge-based operations were carefully applied to the COVID-RENet models to learn regions' homogeneity and extract boundary features. COVID-RENet-1 and 2 were fine-tuned using transfer learning on chest X-rays in the proposed DBHL framework. After combining the penultimate layers of both models, a higher enriched boosted feature space is concatenated, resulting in a single enriched boosted feature space that contains information from both models. By using enriched feature spaces in conventional ML classifiers, COVID-19 detection performance was improved. Based on experiments, they found that the DBHL framework, which merges two-deep CNN feature spaces, achieved high accuracy, sensitivity, F-score, and precision.

Jawad Rasheed [75] performed this analysis using two of the most commonly used classifiers to classify areas in the dataset: logistic regression (LR) and convolutional neural networks (CNN). It was important to speed up and optimize the system as much as possible, and to accomplish that, the dimensionality of the dataset was also reduced through principal component analysis (PCA) in order to speed up the learning process further and to increase classification accuracy by selecting highly discriminating features. While conventional approaches require a large number of training samples, COVID-19 X-ray images lacked an adequate number of labeled training samples. To reduce overfitting, generative adversarial networks (GANs) were utilized for data augmentation. The X-ray images for this study were derived from the available online dataset and incorporated using GAN. In COVID-19 patient identification, CNN and LR showed promising results. Positive case identification accuracy was 95.2%-97.6% without PCA and 97.6%-100% with PCA for the LR and CNN models .

Nahida Habib [76] extracted features from given X-ray images using Convolutional Neural Networks (CNNs), CheXNet [77], and VGG-19. Random under sampling, random over sampling, and synthetic minority oversampling techniques were applied to the ensembled feature vector to overcome the problem of data irregularity. In the next step, several Machine Learning (ML) classification techniques (Random Forest, Adaptive Boosting, K-Nearest Neighbors) are used to classify the ensembled feature vector. Random Forest scored better on the standard dataset than the other methods. Compared to existing methods, the proposed method achieves improved classification accuracy and AUC values and outperforms all other models, predicting with 98.93% accuracy. When tested on a variety of datasets, the model also exhibited generalization capacity. The outcomes of this study can be used to diagnose pneumonia from chest X-ray images.

Danial Sharifrazi [78] proposed a method to detect COVID-19 using X-ray images that combines convolutional neural networks (CNN), support vector machines (SVM), and Sobel filters. The edges of the X-ray images were obtained by filtering a new dataset of X-ray images using a Sobel filter. Following the CNN deep learning model, an SVM classifier has been used to learn with relatively small amounts of data. This method is designed in such a way that it can learn with very little data. The proposed CNN-SVM with Sobel filter (CNN-SVM + Sobel) achieved 99.02% classification accuracy, 100% sensitivity, and 95.23% specificity in the automated detection of COVID-19. In this study, it was shown that CNN performance could be improved by using a Sobel filter. In contrast to most other research methods, this methodology does not use a pre-trained network. They validated their developed model with six public databases and obtained the best performance.

Chapter 3
Methodology

Image datasets are classified using Deep Learning Neural Networks and traditional machine learning algorithms in Artificial Intelligence. These methods have been tested on various datasets, with some advantages and disadvantages. This paper benchmarked traditional and deep learning algorithms for image classification. A description of the machine learning and deep learning algorithms and models used in this study is provided for both datasets. Traditional machine learning algorithms include Random Forests (RF) and Support Vector Machines (SVM), while CNN, VGG19, and Convolutional Auto Encoding algorithms were used in deep learning. The algorithms and hyperparameters used are explained in this chapter.

3.1 Random Forest

A random forest, or random decision forest, is an ensemble learning method that makes multiple decision trees at training time to solve classification, regression, and other tasks. To train random forests, bootstrap aggregation is applied, also known as bagging. By randomly selecting samples from the training set and replacing them with random samples with responses, repeatedly bagging (L times) obtains a random selection of samples for fitting trees to: In the case of i = 1, ..., L:

1) Then replace n of the training examples in X and Y with replacement examples; call these X_i and Y_i.

2) Use the X_i and Y_i as inputs for a classification or regression tree.

Using all the individual regression trees on dataset, predict unseen samples of dataset by averaging their predictions:

$$\hat{f} = \frac{1}{L} \sum_{i=1}^{L} f_i(dataset)$$

Alternatively, classification trees can be selected by majority vote [79]. In traditional machine learning algorithms, features must be extracted, and images should be classified based on those features. A Random Forest algorithm classifies images by converting images into numerical features that can then be input into the algorithm. In order to accomplish this, techniques like feature extraction can be used, in which pixel values, texture features, edge detection, and local binary patterns can be extracted from images. A random forest algorithm can be applied to classify the images using the extracted features. Random forest algorithms can also be applied to other classification problems, with the main difference being that the input data consists of numerical features instead of raw image data.

3.2 Support Vector Machine (SVM)

Support Vector Machines is a robust supervised learning algorithm for classification and regression analysis. Hyper planes are used to classify the dataset according to their optimal positions. Support Vector Machine (SVM) is a classification model to classify and analyze data. SVMs are also capable of classifying images and segmenting images. Once the relevance feedback process is completed, the accuracy of the SVM-based search exceeds that of traditional query refinement schemes. Applying the kernel trick to maximum-margin hyper planes allows nonlinear classifiers to be used for higher dimensional data like images since there are more than two features. We have three classes for our datasets in this study. As a result, the algorithm fits a maximum margin hyperplane in a transformed feature space by replacing each dot product with a nonlinear kernel function. Because there may be a nonlinear transformation and a high dimension in the transformed feature space, the classifier may not be linear in the original input space. Despite this, support vector machines perform well in higher-dimensional feature spaces given enough samples. The loss function for SVM is as following:

$$\min_{w} \lambda \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} max((0, 1 - y_i(w^T x_i - b))$$

Assuming x_i and y_i are samples and targets respectively, w is the normal vector, and $w^T x_i - b$ is the output. Additionally, λ ensures that the margin is on the right side of the margin while increasing the margin size [80].

3.3 Convolutional Neural Networks (CNN)

In deep learning neural networks, convolutional neural networks (CNNs) are particularly effective at analyzing visual imagery. The input data is processed by several layers, each responsible for a specific operation. CNNs are built by assembling layers that use convolutional neural networks as their building blocks. Using learnable kernels or filters, this layer identifies certain features or patterns in the image by comparing them to the input image. Each filter detects a different feature in the image, such as edges, corners, or color blobs. It slides across the input image and performs a mathematical operation known as convolution, which creates a feature map. Convolutional layers are usually combined with activation functions to introduce non-linearity. Activation functions commonly use Rectified Linear Units (ReLUs), which zero out negative values. Using a pooling layer, the output is downsampled after the convolutional and activation layers. By performing operations on non-overlapping regions of the feature map, the pooling layer reduces the spatial dimension of the feature map. As a result, the model has fewer parameters and is less likely to overfit. Input image features are detected in increasingly complex ways by repeating this convolution, activation, and pooling process multiple times. A fully connected layer is usually applied to the output of the convolutional layers to perform classification or regression. With backpropagation, the parameters of the network are adjusted in order to minimize the difference between the predicted and the accurate output during training. Image recognition, object detection, and other computer vision tasks benefit significantly from the convolutional neural network architecture since it is capable of learning features from the input data automatically [81]. Among the most common image classification models are convolutional neural networks, which in earlier layers extract blobs and edges that can be used to identify COVID-19 in chest X-rays. To detect COVID-19, we first developed a convolutional neural network (CNN) model to classify images. This approach used the Keras package to use different MaxPooling, Conv2D, dropout layers for regularization, and Dense layers with ReLu activation functions. Based on a multi-class problem, the SoftMax activation function was used for the last layer of output.

3.4 VGG19

The Visual Geometry Group (VGG) has developed a convolutional neural network architecture that has been trained over a large dataset of images for the purpose of classifying the images. It consists of 19 layers, which have been developed for deep learning on a wide range of datasets. There are 16 convolutional layers in the VGG19 architecture, followed by three fully connected layers divided into five blocks containing multiple convolutional layers. By applying the same padding to the input and using a 3x3 filter, VGG19 produces an output with the same spatial dimensions as the input. A ReLU activation function follows the convolutional layers, which introduces nonlinearity to the model. To prevent overfitting and reduce spatial input dimensions, the max-pooling layers are used to downsample feature maps. In VGG19, the fully connected layers produce output probabilities based on the output classes. The final layer is computed by using a softmax activation function. In supervised learning, VGG19 is trained by minimizing the subtraction of predicted and actual output with a loss function to determine the weights of the convolutional and fully connected layers. ImageNet, for example, is used to train a large dataset of images with well-known labels. It has achieved state-of-the-art performance in a wide range of image classification tasks thanks to its powerful image classification model, VGG19.

Complex features in images can be captured effectively with its deep architecture and small filters with the same padding. This model has an excellent generalization to various datasets in various fields and uses three-layered convolution filters and deeper nets [82]. The VGG19 CNN model is pre-trained on ImageNet and is pre-trained on a million images. Furthermore, it categorizes images based on their shapes and colors. Consequently, we used this model to classify both datasets and their augmented data to examine their performance. A few fully connected layers were used to extract features. In the Diagram 3.1, the phases of the VGG19 model for training have been provided.

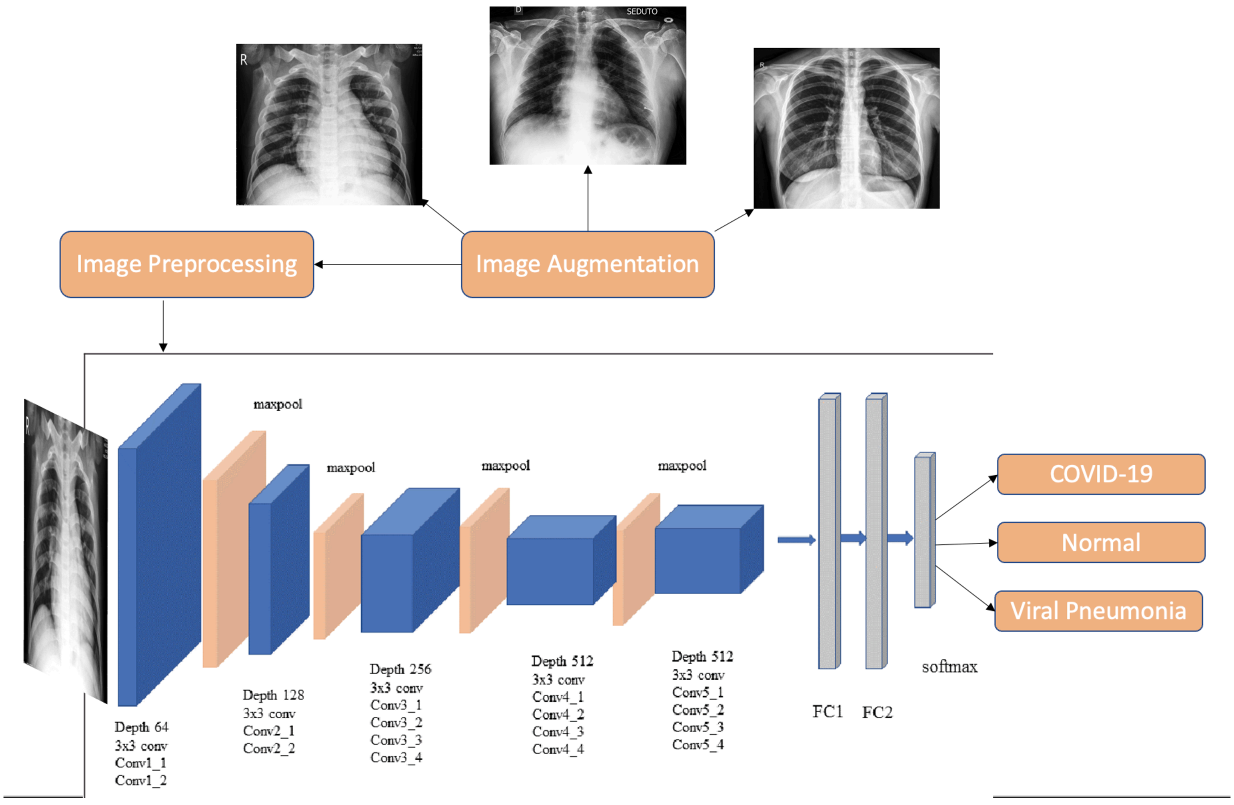


Figure 3.1: This flow diagram shows the different steps of the VGG19 model that have been applied in this study. In the first step, more training data were generated using data augmentation. Images were preprocessed to train the model, and finally, the SoftMax function was used to classify the images into COVID-19, Normal and Viral Pneumonia [83].

3.5 Convolutional Auto Encoder

The Convolutional Autoencoder is a type of neural network capable of compressing and reconstructing images. Generally, Conventional Auto Encoders (CAE) consist of two layers, encodes and decodes. The encoder learns to extract meaningful features from an input image through a series of convolutional layers. A compressed representation of these features is then created using pooling or upsampling layers. This compressed representation contains the most critical information about the input image and is called a ”latent code.” In order to reconstruct the original image, the decoder uses multiple deconvolutional layers. The deconvolutional layers learn to ”fill in” the missing details and create an output image as close to the original input. As part of training, an autoencoder uses a mean squared error or binary cross-entropy for the loss function to minimize the difference between input and output images. As a result of minimizing this loss function, the autoencoder learns how to encode input images into compact latent representations that capture their most important features. We minimize the following formula if $f_w(.)$ and $g_u(.)$ are encoder and decoder functions, respectively:

$$\min_{w,u} \frac{1}{n} \sum_{i=1}^{n} \|g_u(f_w(x_i)) - x_i\|_2^2$$

and fully connected layers for autoencoder:

$$f_w(x) = \sigma(wx) = h$$

$$g_u(h) = \sigma(uh)$$

which x and h are vectors and σ is an activation function like sigmoid or ReLU. It is common to use convolutional autoencoders to denoise images, compress them, and extract features from them. Additionally, Combining them with other neural network architectures, such as convolutional neural networks, can enhance their ability to classify images and detect objects [84]. In the Diagram 3.2, the phases of the Cnovolutional Auto Encoder model for training have been provided.

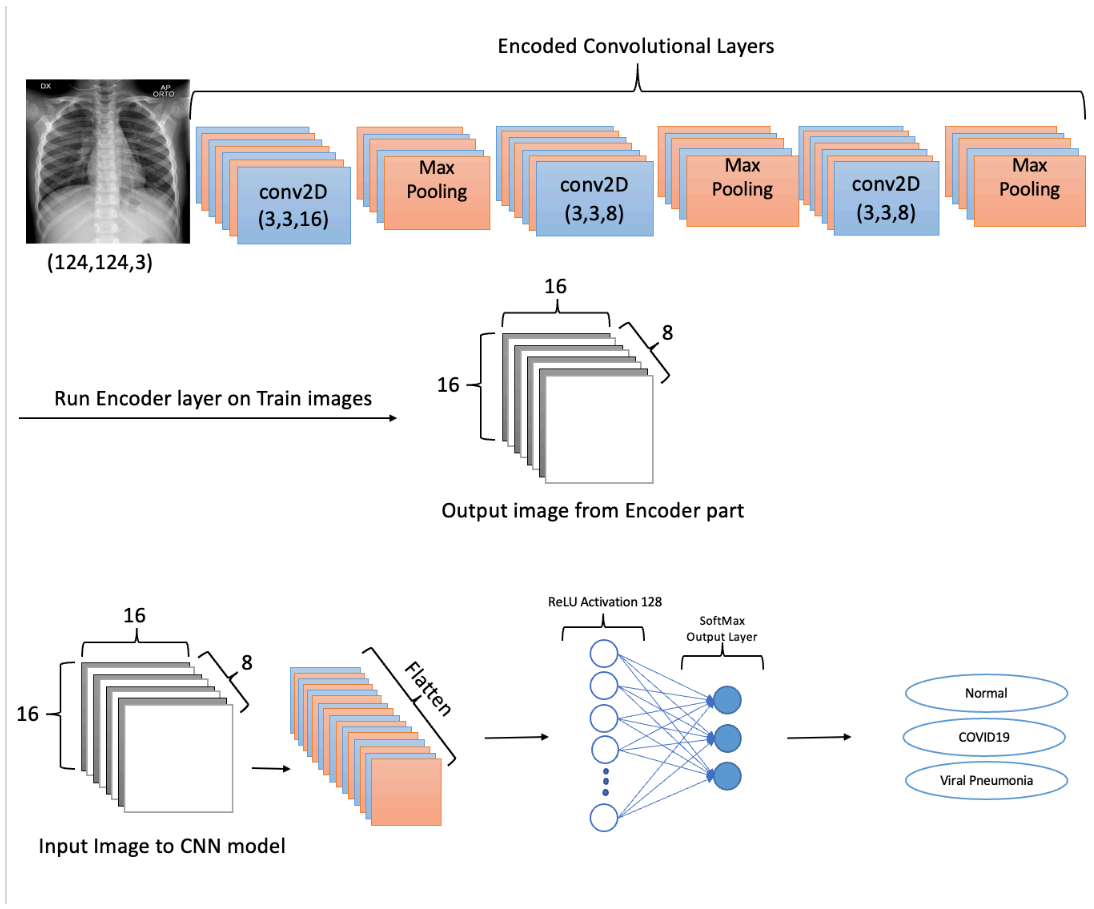


Figure 3.2: In this diagram, we demonstrate the phases of the Auto Encoder model that we used in this study. The input image with dimensions of (124,124,3) is passed to the encoder part, and after a few layers, we obtain an image with dimensions of (16,16,8), which we use as input for a CNN model for training.

Chapter 4

Data

For every research project, data is needed to work with and verify the results. We can use this to determine if the results are reliable and performance-oriented. Data Augmentation is also a way to meet particular needs in datasets, such as helping with small datasets and making accurate generalizations in training. The methods used in this study to meet these requirements, work with different data sets, and augment the data are varied. In studying different articles, we decided to try different methods that were not used in the papers we studied. We wanted to test different techniques and possibilities to see if they could improve model accuracy. By exploring techniques that have not been explored in the examined papers, we could gain insight that would assist in improving our study's results. Our experiment aimed to increase our models' accuracy by experimenting with various data augmentation techniques. A bright and contrast method was used in this study to augment data to test if the method assisted with enhancing. These methods were also mixed with the flipping method in order to obtain the results. We achieved the results by enhancing data simply with brightness and contrast and adding the flipping method. In order to apply these methods to our model, we had to create new images by using a package or class that applied these methods to our model. Compared with an image generator or an open CV package, Data Augmentor is a package that helps you see exactly how many images have been created and save them in a folder for easy reference. With the help of Data Augmentor, we were able to create new images by changing the brightness and contrast of them using the flipping method. Our model becomes more robust as a result of the exposure to more diverse data. As a result of the package, we could also keep track of the number of images we have created as well as organize them in an orderly manner. We have discussed the datasets as well as the augmented datasets used in this study through this chapter. We will also discuss what we did with them in order to archive better results and evaluate the performance of different data sources that were used. As such, it was essential to assess the data quality in order to be able to trust and justify the results of the research project. In this chapter, we have explained in detail how two datasets from two different sources were used in the study. This document outlines the process of evaluating both datasets, as well as possible limitations and issues based on our findings.

4.1 Data Augmentation

In this study, we used the package Augmentor to create new images from existing data. Among the existing data (Dataset 1 and Dataset 2), two augmented data sets were created, one based on brightness and contrast with 0.5 probability for each method. The min factor was adjusted to 0.3 and the max factor to 0.5. In another augmented data set, the brightness and contrast min factors and max factors were the same, but the probability changed when the flipping method was added; contrast had a probability of 0.4, brightness had a probability of 0.3, and flipping had a probability of 0.3. This adjustment of min and max factors ensures that the augmented data sets have the same overall brightness and contrast but with different probabilities for the different methods. By adjusting the probability of each method, the data sets can have different levels of brightness, contrast, and flipping, which helps create more diverse data sets for training.

4.2 Dataset 1

We received our first dataset from the University of Montreal in Canada. The dataset contains three types of chest X-ray images, Normal, Covid-19, and Viral Pneumonia images, examined in this study. A total of 317 images are included in this dataset, divided into two groups, training and test images. A total of 111 Covid-19 images and 70 images are provided separately for each Normal and Viral Pneumonia class. Furthermore, 26 Covid-19 images and 20 images for each class are included in the test set [1]. Our dataset in dataset 1 is relatively small, so we used augmented data to create additional images. This was done to avoid overfitting or underfitting issues and attempt to achieve better results. As part of the data augmentation, we used flipping, brightness, and contrast and made 2000 images for each class to perform the augmentation. A summary of dataset 1 has been provided in Table 4.1.

Table 4.1: Summary of dataset 1 used in this study.

	Train Data	**Test Data**
COVID-19	111	26
Viral Pneumonia	70	20
Normal	70	20

4.2.1 Data Augmentation for Dataset 1

As part of deep learning, data augmentation entails creating tweaked versions of the original data to expand a training dataset artificially. By using this technique, image classification tasks can be improved in terms of generalization and robustness. Data augmentation helps to create more variations of the original data, reducing overfitting and improving the model's ability to recognize patterns in unseen data. It also helps the model to learn more complex features that can assist it in classifying images more accurately. Various transformations can be applied to the initial images to produce images with different appearances. However, these images still represent the same underlying objects or scenes, including flipping, rotating, cropping, or zooming. The training data will be more diverse, which will make it easier for the model to learn robust, invariant features. Additionally, Deep neural networks can overfit when training data if they are too complex and have a small dataset. By artificially increasing the size of the training dataset, the model can generalize more efficiently to new, unseen data. This prevents the model from memorizing the exact training examples. By augmenting the training dataset, the model can learn more robust features that are not dependent on the actual training examples, thus avoiding overfitting. We used data augmentation to reduce overfitting and compensate for the small training dataset in this study. This helped us produce more reliable and robust results. The training data was split into two sections: training and validation. The first train data we augmented from them was 91 images for COVID-19, 50 images for each of normal and viral pneumonia. We then selected 20 images from each class for validation since augmented data cannot be used for validation. So we augmented images from the remaining data in the training set. In order to augment our data, we applied brightness, contrast, and flipping, each independently, using the Augmentor package. After applying brightness and contrast, we added flipping to the methods and tested them. As a result, we had two augmented datasets: one with brightness and contrast and one with brightness, contrast, and flipping. Through this way, we could analyze the algorithms' performance on various augmented datasets and determine which combination of augmentation techniques produced the best results. We managed to gain a better understanding of how the augmentation techniques affected algorithm performance by comparing the results of the different augmented datasets. We then adjusted the training process accordingly based on which combination of techniques produced the best results. A summary of augmented dataset 1 has been provided in Table 4.2.

Table 4.2: Summary of dataset 1 used in this study.

	train data	**validation data**	**test data**
COVID-19	2000	20	26
Viral Pneumonia	2000	20	20
Normal	2000	20	20

4.3 Dataset 2

Compared to dataset one, dataset two is much larger and contains a more significant number of images than dataset one. It was created by researchers from Qatar University in Doda, Dhaka University in Bangladesh, Malaysia, and Pakistan based on chest X-ray images of four different categories: normal images, Covid-19 images, viral pneumonia images, and lung opacity images. The first release included 219 COVID-19 chest X-rays, 1341 normal X-rays, and 1345 viral pneumonia X-rays. The second update included 1200 COVID-19 chest X-rays. The second update added 3616 COVID-19 positive cases, 10,192 Normal, 6012 Lung Opacity (Non-COVID lung infections), and 1345 Viral Pneumonia images and corresponding lung masks to the database. They will update the database as soon as the COVID-19 pneumonia images become available. Due to the limited number of classes in dataset 1, we did not use lung capacity images in this study, and we compared the results in each dataset and investigated the results in another dataset. In addition, as we had imbalanced data, some images were reduced to make balanced data, and data augmentation was used to check the model's performance on augmented, reduced data [2]. A summary of dataset 2 has been provided in Table 4.3.

Table 4.3: Summary of dataset 1 used in this study.

	train data	**test data**
COVID-19	10182	50
Viral Pneumonia	3652	50
Normal	1295	50

4.3.1 Data Augmentation for Dataset 2

Even if we have a large dataset to work with, data augmentation can still be helpful for image classification. In applications where the input images may differ in quality or appearance, data augmentation can help the model identify objects more robustly and invariably. The training data may not adequately represent certain image variations, even if the dataset is significant. By artificially increasing the diversity of training data, we can ensure that the model can recognize objects robustly and invariantly. In addition, large datasets may still have biases that can affect the model's performance. Those types of images, for example, might be more accurately recognized if the dataset consists primarily of images of a certain race or gender. With data augmentation, we can achieve a better distribution between the classes in training data and reduce model bias. Often, large datasets contain redundant images or similar images, which make the model overfit to the training data and perform poorly on new, unseen ones. By artificially increasing the diversity of training data, data augmentation can reduce overfitting and prevent the model from memorizing the exact training examples it is supposed to learn. In this way, the model can be more easily generalized to new, unknown images. As part of this study, we first reduced the number of images to like the number of Viral Pneumina images by 1300 images, then 260 images from this train data were analyzed for each class in the validation set and 260 images for the test set, with the remaining images being utilized for data augmentation to make 2000 images for the trainset. Our data augmentation was performed using the Augmentor package, as explained before. Using the Augmentor package, we created augmented images by applying brightness, contrast, and flipping, each tested separately. After applying brightness and contrast, we added flipping to the methods and tested the algorithms. Therefore, we had two augmented datasets based on brightness and contrast and one based on brightness, contrast, and flipping. This allowed us to compare the performance of the algorithms on the different augmented datasets and to see which combination of augmentation techniques produced the best results. By comparing the results of the different augmented datasets, we were able to understand better how the augmentation techniques were affecting the performance of the algorithms. This allowed us to determine which combination of techniques produced the best results and to make adjustments to the training process accordingly. Table 4.4 provides a summary of the data.

Table 4.4: Summary of dataset 1 used in this study.

	train data	**validation data**	**test data**
COVID-19	1300	260	260
Viral Pneumonia	1295	260	260
Normal	1300	260	260

Chapter 5
Experimental Results

We began with two datasets from two different resources to conduct this study. We examined our methods and algorithms to evaluate the effectiveness and ability of these methods and algorithms to achieve better results in the world of artificial intelligence. To assess the usefulness of these methods and algorithms, we compared the results of our analysis to the original datasets to determine their accuracy and reliability. During this chapter, we will describe how we have pre-processed the datasets we have used in this study and how we have used them in this research. In addition, we will present the experimental results we achieved in the previous chapter by applying the methods we discussed in that chapter. We have used various metrics, such as precision, recall, and F1-score, to evaluate our results. Furthermore, we have compared our results with benchmarks to show how our proposed methods and algorithms can provide better performance than existing methods. Finally, we have discussed the impact of using different data pre-processing techniques on the performance of our algorithms. Here, we have discussed the proposed models on both datasets that were mentioned in chapter 4 and the hyperparameters applied to these models. The models were evaluated based on the performance metrics such as accuracy, precision, and recall. The hyperparameters, such as learning rate and regularization strength, were tuned to optimize the performance of the models, and the results were compared across the two datasets to determine which model works best in each case. Furthermore, the results of the experiments confirmed that the model tuned with the correct hyperparameters achieved superior performance for both datasets, demonstrating the effectiveness of the proposed approach.

5.1 Image Pre-processing

The first step in analyzing images is preprocessing them. These methods include transformation, orientation, and resizing, which make images more appealing. The function of models requires that images be preprocessed before they can work correctly. As a result of this process, we can classify and work. In classical machine learning algorithms, images cannot be used as they are without first extracting features and saving them in a data frame. To extract features and detect edges, we used filters for feature extraction. For feature extraction in this study, we used three different filters, including pixel values, Gabor filters with different hyperparameters ($\gamma = 0.5$, and $\lambda = \frac{\pi}{4}\,\theta$, $\sigma \in (1,4)$) and the Sobel filter which is for edge detection. We can extract features with different theta, and sigma ranges from a function with a for loop. It is created by first 'for loop' theta in range(1,4) and then dividing this number into four times pi, and then by second for loop inside this loop, σ is in range(1,4), and then a Gabor filter with kernel size = 9 is created by using the $\lambda = \frac{\pi}{4}$ and the $\gamma = 0.5$. A Gabor filter with different orientations and scales can be created by looping through theta and sigma ranges. This allows for a more compelling feature extraction since the filter can be tuned to the specific features in the image that need to be extracted. The kernel size of 9 ensures that the filter captures enough of the image's features. The lambda and gamma parameters adjust the filter's orientation and scale, respectively. Additionally, image preprocessing can speed up the training and running of the model. We investigated resizing all the images because they were different sizes. The fully connected layers of convolutional neural networks require images of the same size to function. After resizing images to different sizes, like 100*100*3 and 150*150*3, we discovered that 50*50*3 was the most effective size for superior performance. Then we down-sampled the images to (50,50,3) sizes so that we could try to tune the model on the same size dataset. This would ensure that the model would be trained on the same size dataset and that the convolutional layers could properly detect the patterns in the images. This would lead to improved accuracy of the model and better performance. This dimension was used for VGG19 and CNN training; however, for Auto encoder training, this dimension (124,124,3) was used since auto encoders reduce the dimension themselves, and (50,50,3) was a low dimension. Autoencoders are used to compress data into a low-dimensional representation. To capture the most information, the initial dimension should be larger than the output dimension. Therefore, (124,124,3) was used as the input dimension to ensure that the auto-encoder had enough information to generate meaningful output.

5.2 Random Forest

5.2.1 Random Forest on Dataset 1

Based on feature extraction using all hyper parameters mentioned in image processing, Random Forest Classifiers without and with regularization (Lasso and ridge regression) were used on both datasets. Random forest with default parameters with 50 estimators (Gini loss function and without max-depth hyper parameter) was used. This was done because the Random Forest Classifier is robust and accurate compared to other classifiers. The use of 50 estimators was chosen to ensure high accuracy in the results. Additionally, image processing allowed extracting features that could be used to refine the model's accuracy further. The random forest method without regularization proved to achieve 81% accuracy in the results. When we ran this model on the second dataset, we received 58%

accuracy, which could have been better for the updated dataset. As we can see, the problem was classifying the Normal images as Covid-19 images. We observed that the model over-predicted Covid-19 images, classifying Normal images as Covid-19.

Then we tried a regularized Random forest model with 100 estimators, max-depth =10, class weight = balanced-subsample, and reached higher accuracy by 2% more, reaching 83%. This enhancement leads to one more percent accuracy when checking the second dataset by 59%. The Figure 5.1 contains information about confusion matrices of these performances and checking a training model on a second dataset to investigate how it performs on unseen data.

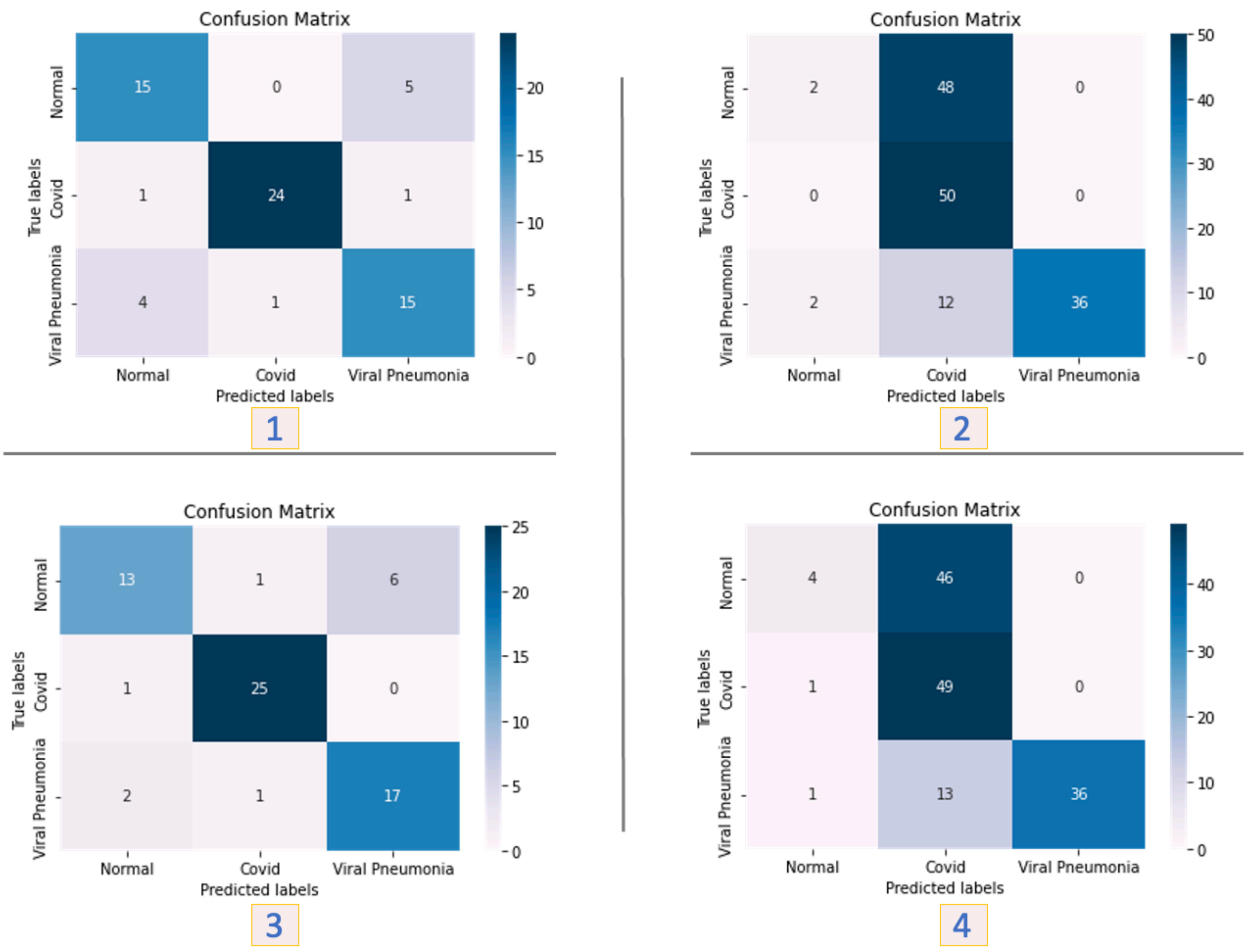


Figure 5.1: In this diagram, Confusion matrix (1) shows the performance of Random Forest without Regularization on Dataset 1, Confusion matrix (2) shows the investigation of Random Forest without Regularization on test data of Dataset 2, Confusion matrix (3) shows the performance of Random Forest with Regularization on Dataset 1, and Confusion matrix (4) shows the investigation of Random Forest with Regularization on test data of Dataset 2.

5.2.2 Random Forest on Dataset 2

Since dataset 2 is large, machine learning algorithms are unsuitable for large datasets, and training them takes a long time. Machine learning algorithms are computationally expensive and require many resources to train them. Training on large datasets can take excessive time and resources, so decreasing the number of images can help speed up the training process. Random forest on dataset 2 required us to reduce the number of Normal images due to the GPU and system running on the machine. In the same way, as with COVID-19 images, we reduced the number of Normal chest X-ray images to 3652. We, therefore, had 3652 images for Normal and COVID-19 for each class and 1295 images for Viral Pneumonia. We strived to maintain a balanced dataset, as this would allow us to make better predictions using random forests. Based on feature extraction using all hyperparameters mentioned in image processing, Random Forest Classifier with 50 estimators, Gini loss function, and without max-depth hyperparameter was used on Dataset 2. This was done because the Random Forest Classifier is robust and accurate compared to other classifiers. The use of 50 estimators was chosen to ensure high accuracy in the results. Additionally, image processing allowed extracting features that could be used to refine the model's accuracy further. This approach proved successful, with the Random Forest Classifier achieving 91% accuracy in the results. When we ran this model on the second dataset, we received 60% accuracy, which was not ideal for the updated dataset. As we can see, the problem was classifying the Normal images as Covid-19 images. We observed that the model over-predicted Covid-19 images, classifying Normal images as Covid-19. Moreover, we applied the algorithms to Dataset1, a larger dataset, and the result was 91% accuracy. However, we could not get an accurate result of 60% accuracy when we used the trained model on Dataset 1. The results of these tests can be seen in the confusion matrix provided in Figure 5.2, which can be seen in Confusion matrices 1 and 2. Also, we used a regularized random forest for this dataset too to examine that. We achieved 2% more accuracy by 93% in this trial. The model was further tested on the dataset and achieved 62% as shown in Figure 5.2 in Cofusion matrices 3 and 4. We could see that this is just like the previous experience with dataset1 with this difference that we got higher accuracy and prediction. This can be because of the increasing number of training data as we had a larger dataset. Also, a table has been provided to present a summary to show the performance of the Random Forest classifier on both datasets. This can be seen in Table 5.1. The random forest had better results with 91% accuracy on dataset 2, and this accuracy increased by 2% with regularization. However, both models on dataset one and Dataset 2 could not predict data from different sources effectively, with about 60% accuracy for both models.

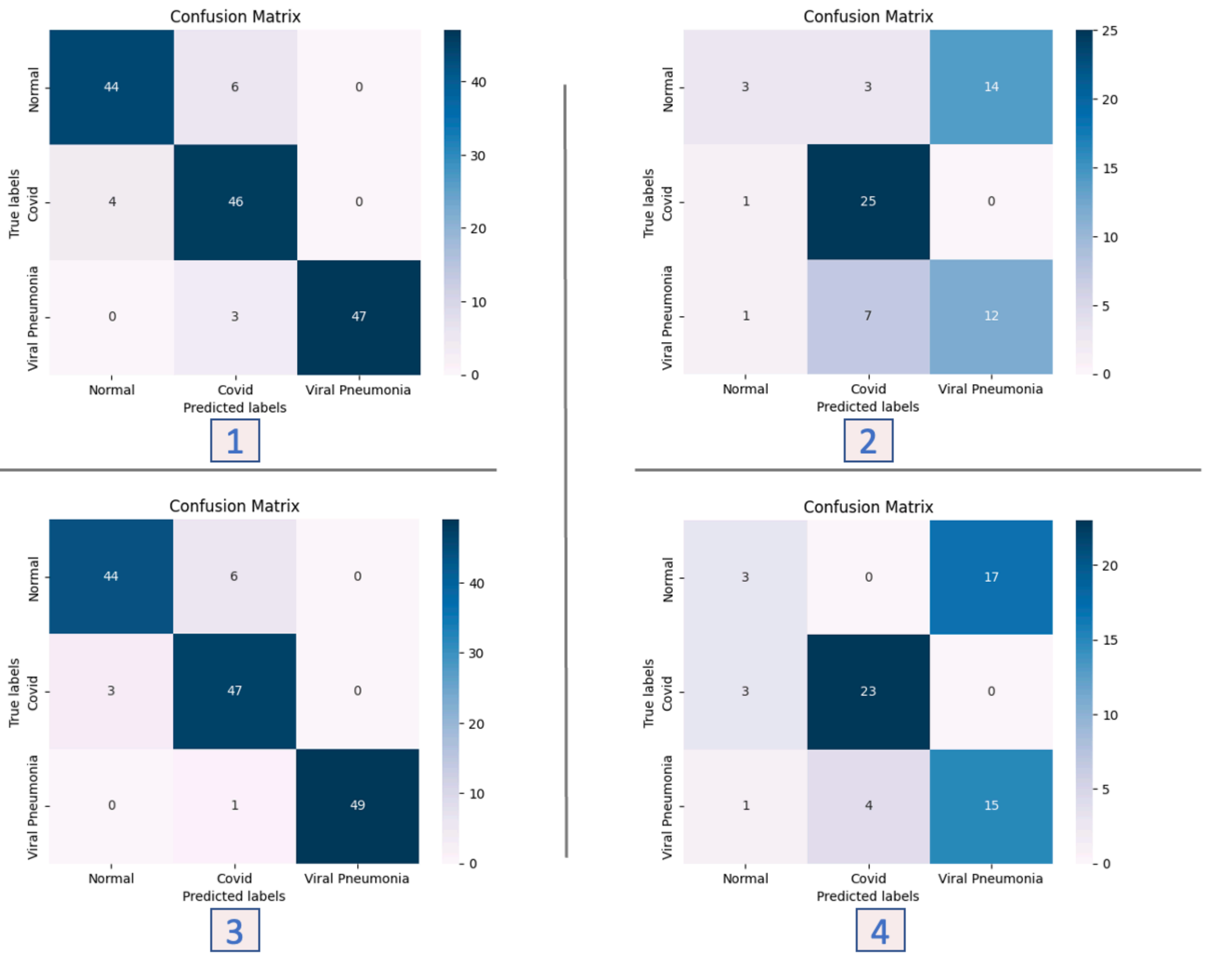


Figure 5.2: In this diagram, Confusion matrix (1) shows the performance of Random Forest without Regularization on Dataset 2, Confusion matrix (2) shows the investigation of Random Forest without Regularization on test data of Dataset 1, Confusion matrix (3) shows the performance of Random Forest with Regularization on Dataset 2, and Confusion matrix (4) shows the investigation of Random Forest with Regularization on test data of Dataset 1.

5.3 Support Vector Machine

5.3.1 Support Vector Machine on Dataset1

Using all hyperparameters mentioned in image processing, a Support Vector Machine Classifier was implemented for multi-class classification based on feature extraction. The number of classes* (Number of classes - 1) divided by two classifiers are constructed, and each class trains data from two classes. Using the decision function shape option, it can monotonically transform the results of ”one-versus-one” classifiers into a ”one-versus-the-rest” decision function of shape (n-samples, n-classes). ”One-versus-one” classifiers are binary classifiers that learn a discrimination function for each pair of classes. By combining these classifiers, a multi-class classifier can distinguish between all classes. The

Table 5.1: The table summarizes the Random Forest performance on Dataset1 and Dataset2.

Model	Dataset1	Dataset2	Test(Data1 on Data2)	Test(Data2 on Data1)
Without Regularization	81	91	58	60
With Regularization	83	93	59	62

decision function shape option allows the user to transform the output of the ”one-versus-one” classifiers into a ”one-versus-rest” decision function with a shape of (n-samples, n-classes), where n-samples is the number of images and n-classes is the number of image classifications. This model allows users to classify multiple classes with a single function easily. A further refinement of the model’s accuracy was possible through image processing, which allowed us to extract features from the image. The Support Vector Machine Classifier without regularization on Dataset 1 proved less successful, achieving 77% accuracy shown in the Confusion matrix(1) Figure 5.3 in the results. Also, we checked on the second dataset; however, we achieved 64% accuracy provided in the Confusion matrix (2) Figure 5.3, which could have been more optimal compared to Random Forest Model. As we can see, the problem was in classifying the normal images as COVID-19 images. The model overpredicted Covid-19 images and thus classified Normal images as Covid-19 images. Moreover, we applied regularization to the SVM model with kernel trick to check if this will enhance or not, and by using kernel = rbf, auto gamma, and C = 1, we received bad results by 39% shown in Confusion Matrix (3) Figure 5.3 dataset1 and when we checked the regularized model on test dataset 2 from the second resource we achieved 33% for accuracy provided in Confusion Matrix (4) Figure 5.3.

5.3.2 Support Vector Machine on Dataset2

Moreover, we applied the algorithms to Dataset2, a larger dataset. We achieved 90% accuracy for SVM without regularization, as shown in the Confusion Matrix (1) of Figure 5.4, but when we used the trained model on Dataset1 that provided 62% accuracy, as shown in the Confusion Matrix (2) of Figure 5.4, we were unable to achieve a good accuracy result.

A regularized SVM was also used to evaluate performance by regularizing the model with kernel trick and gamma = auto and C = 1. We did not improve performance but reduced performance by more than 30% for dataset 2. As shown in the Confusion Matrix (3) of Figure 5.4, the results when testing dataset 1 from different sources are 39% accuracy in the Confusion Matrix (4) of Figure 5.4. In addition, a table has been provided to summarize all the results of Support Vector Machines on both datasets, which can be found in Table 5.2.

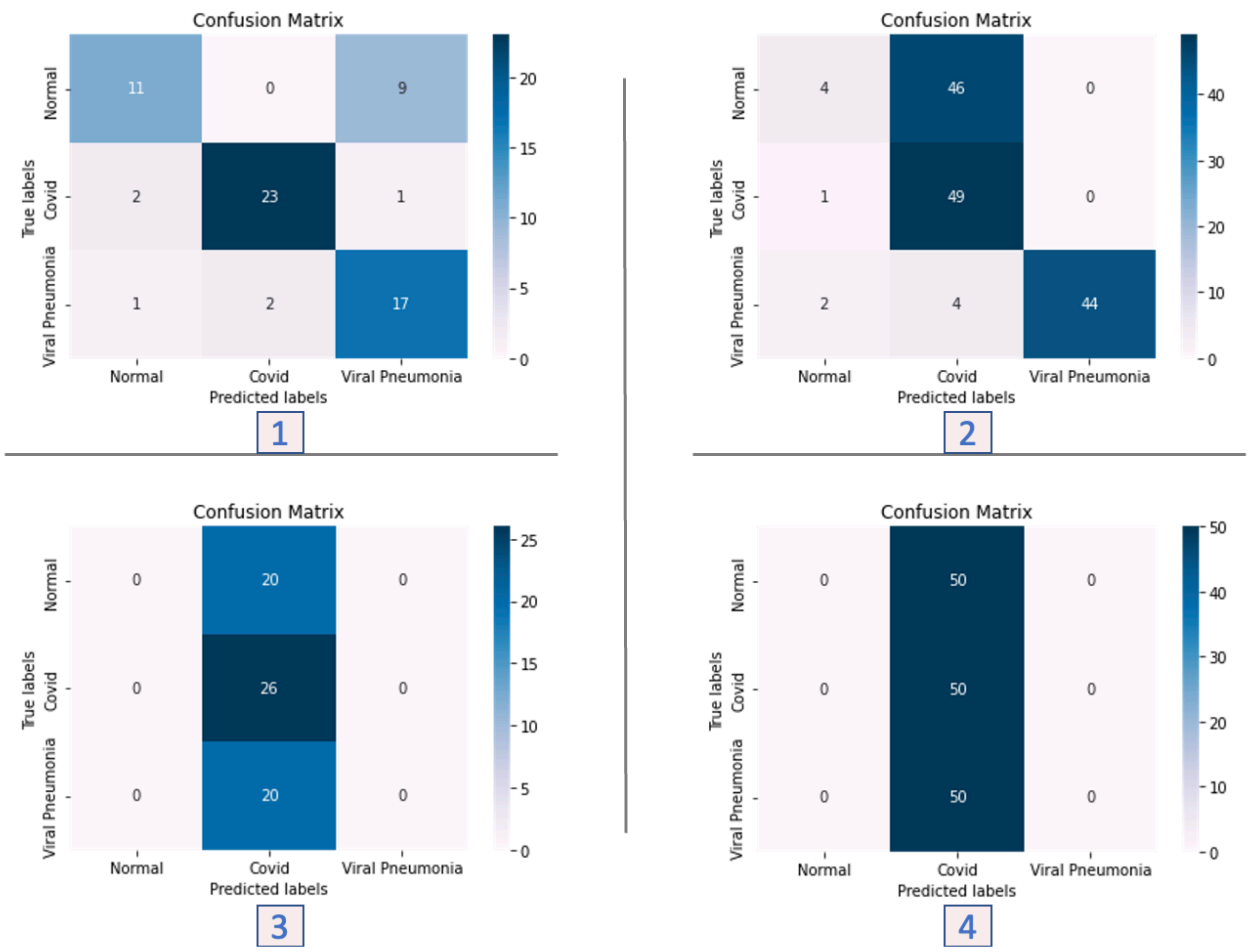


Figure 5.3: In this diagram, Confusion matrix (1) shows the performance of Support Vector Machine without Regularization on Dataset 1, Confusion matrix (2) shows the investigation of Support Vector Machine without Regularization on test data of Dataset 2, Confusion matrix (3) shows the performance of Support Vector Machine with Regularization on Dataset 2, and Confusion matrix (4) shows the investigation of Support Vector Machine with Regularization on test data of Dataset 1.

This table shows that Support Vector Machine performed best on Dataset 2 compared to Dataset 1 and regularized version, but this model could not reach nice results on different source datasets as well.

5.4 Convolutional Neural Networks (CNN)

A Convolutional Neural Network, which is a popular method of image classification, was presented in this paper. We tested the performance of Convolutional Neural Networks by modifying the model with different regularizations, including Ridge regression, Lasso regression, and dropout regularizations. In earlier layers, convolutional neural networks extract blobs and edges, which helps identify COVID-19 in chest X-rays. To detect COVID-19, we designed a convolutional neural network (CNN) as a first step. A Keras package

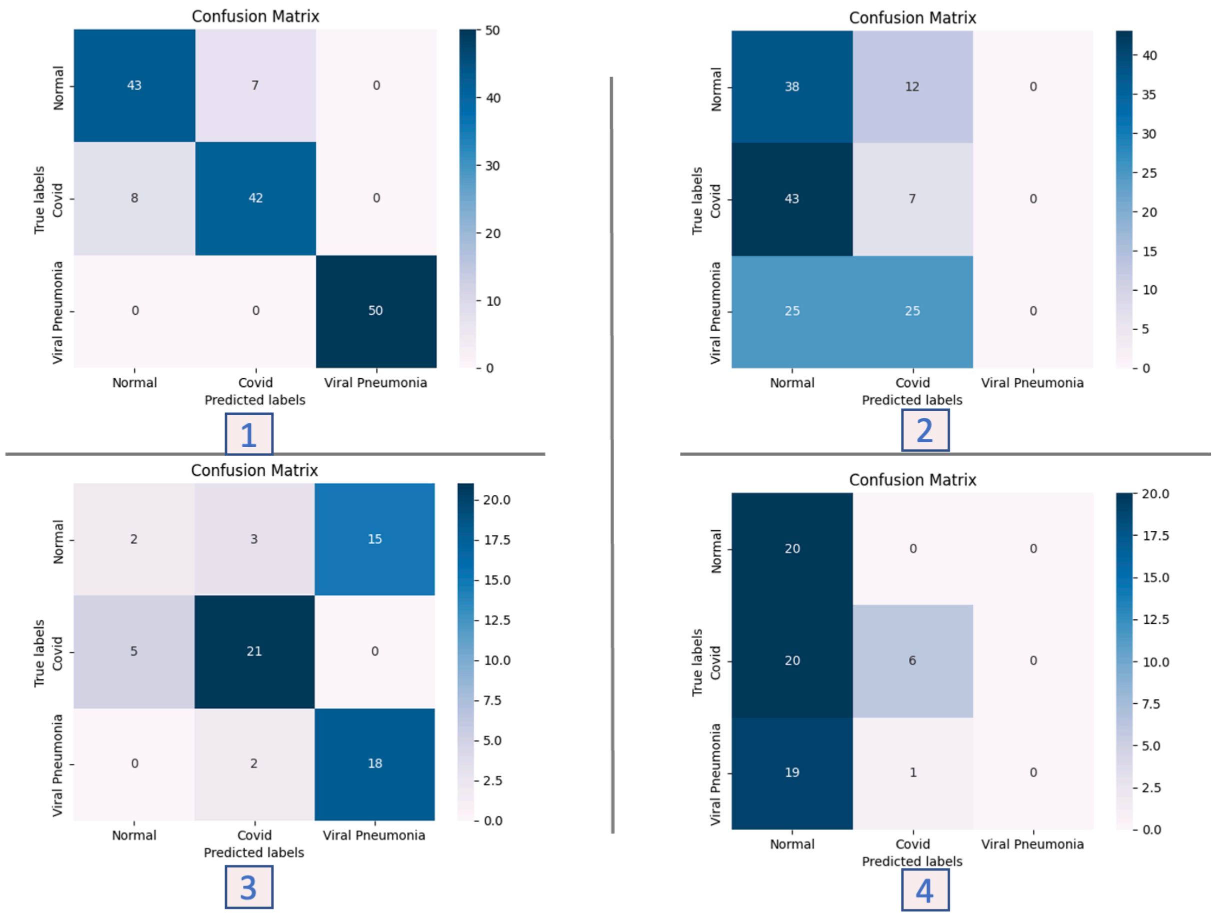


Figure 5.4: In this diagram, Confusion matrix (1) shows the performance of Support Vector Machine without Regularization on Dataset 1, Confusion matrix (2) shows the investigation of Support Vector Machine without Regularization on test data of Dataset 2, Confusion matrix (3) shows the performance of Support Vector Machine with Regularization on Dataset 2, and Confusion matrix (4) shows the investigation of Support Vector Machine with Regularization on test data of Dataset 1.

was applied along with MaxPooling, Conv2D, Dropout, and Dense layers activated with ReLu functions. The final output layer combined a multi-class problem with the SoftMax activation function. This model was designed with a sequential model with 16 layers. The input layer was a 50x50x3 image. We also applied five dropout layers with 0.25 parameters, Ridge Regression with 0.01 for kernel regularizer, and Lasso Regression with 0.01 for bias regularizer as penalty terms for regularizing the model. We also used batches of 16 sizes and 200 epochs. Using an Adam optimizer and a categorical cross-entropy loss function with parameters of from logits = True. The logit parameter indicates whether the activation function should be applied to the model output or not. In this case, the SoftMax activation function should be applied to the model output; hence the parameter is set to True. Using the proposed model described in this section, we achieved different accuracy through regularization methods. However, the models overfitted after some epochs. This means the model could accurately predict outcomes based on the data it was trained

Table 5.2: The table shows the performance of the Support Vector Machine on Dataset 1 and Dataset 2.

Model (SVM)	**Dataset1**	**Dataset2**	**Test(Data1 on Data2)**	**Test(Data2 on Data1)**
Without Regularization	77	90	64	62
With Regularization	39	30	33	39

on but could not generalize to untrained data. As a result, the model tended to ”memorize” the data it was trained on, causing it to overfit. We used dropout regularization, Lasso regression, or Ridge regression for regularization. This research found that dropout regularization helped increase accuracy and reduce overfitting, but it was insignificant. Moreover, we also applied the best-trained model that was trained on one dataset to another dataset from a different source in order to test it. In terms of accuracy, we did not achieve a significant level of accuracy, and what we found was that the Normal images, which made up about 90% of the images, were over classified as COVID-19 images. Additionally, there were cases in which COVID-19 and viral pneumonia were misclassified as Normal. There could be a reason for this, which may be the smaller size of the dataset compared to the second dataset, which might mean that the model needed to be trained on more examples from the dataset to be able to classify them accurately. There is also another reason for the inability to classify well in smaller datasets, which can be caused by a problem of overfitting in larger datasets. The training history for each regularization method and dataset is available. Also, the confusion matrix is provided when we applied a model to one dataset and checked it on another dataset from different sources in the figures.

5.4.1 CNN on Dataset 1

In this study, we applied three different regularization methods on three different forms of dataset 1. To begin with, we applied L1, L2, and Dropout regularization to the original dataset, which is a small dataset. With the CNN model that we developed using the L1 regularization on the original dataset 1, we were able to achieve 89% for accuracy, and the test on the dataset achieved 60%. The CNN deep learning model was trained on the original dataset 1, with L2 regularization. The model was then evaluated on a test set from the original and second datasets. Confusion matrices were generated to measure the accuracy of the model’s predictions on both datasets. Additionally, training plots for each epoch were provided to visualize the learning process. The deep learning model achieved 90% accuracy when evaluated on the test set from the original dataset L2 regularization.

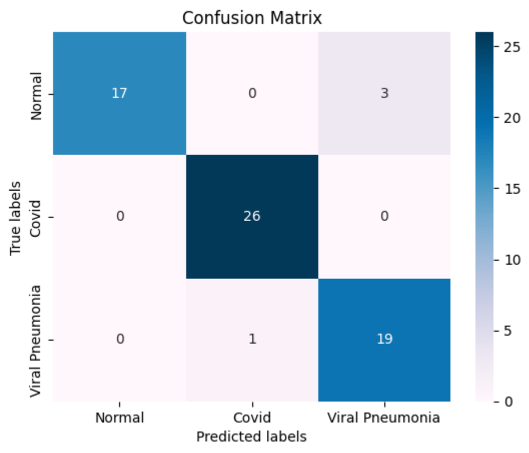


Figure 5.5: This diagram shows the confusion matrix of the CNN model with Dropout regularization algorithm's predictions for dataset1.

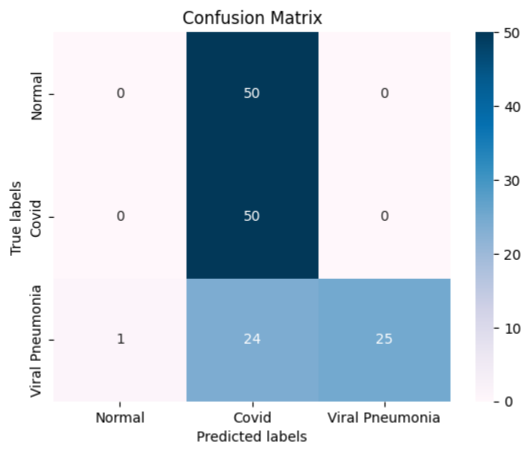


Figure 5.6: In this diagram, we show the confusion matrix for the CNN model based on dataset1's data for dataset2's test data.

However, when evaluated on the second dataset, accuracy dropped to 54%. On the original Dataset 1, dropout regularization had 93% accuracy for checking in the same test set and 50% accuracy when checking in Dataset 2. Figure 5.5 and Figure 5.6 show confusion matrices indicating the number of predictions. Figure 5.7 provides a plot of training for each epoch as well. Additionally, we applied all regularization methods to augmented data with brightness and contrast to see how the algorithms would be enhanced. The performance of CNN with L1 regularization worsened as the accuracy decreased to 39% and when tested on Dataset 2 the accuracy became 33%. In order to investigate the capabilities of L2 and Dropout, regularization techniques were applied on Augmented Dataset 1, which had been enhanced using brightness and contrast adjustments. Although the training accuracy of the models was not high, achieving only 60% and 63% accuracy for the best epoch, the testing accuracy on the test set was much better, with values of 93% and 84% for L2 and dropout regularization, respectively, as illustrated in Figure 5.10. The trained models were also tested on Dataset 2, and the results are presented in Figure 5.9. The confusion matrices of the number of predictions for both models are provided in Figure 5.8.

Specifically, we applied brightness, contrast, and flipping techniques to augment Dataset 1 and trained models on these augmented datasets to evaluate their performance. Our primary aim was to determine the effect of the flipping technique on the prediction model's enhancement. Additionally, we compared the performance of L1 regularization with augmented brightness and contrast to that of L2 regularization and dropout regularization. Our findings indicate that the performance of models with flipping was comparable to those without flipping. Moreover, the results obtained with L1 regularization with augmented brightness and contrast were similar to those obtained with L1 regularization augmenta-

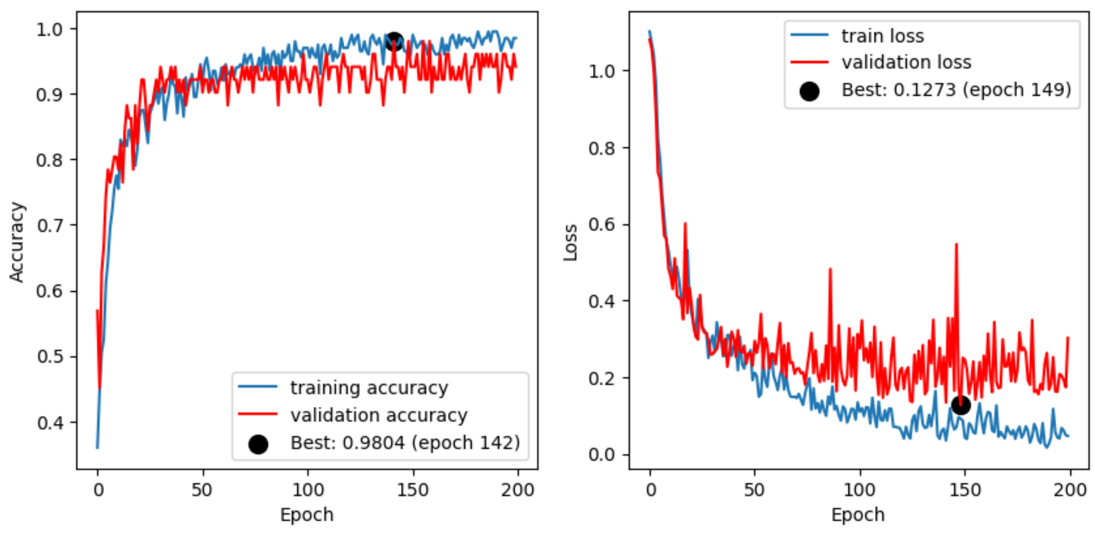


Figure 5.7: A chart showing loss and accuracy for CNNs with dropout regularization for each epoch, including the best loss and accuracy.

tion by brightness, contrast, and flipping. Additionally, the models trained with L2 and dropout regularization exhibited high accuracy of 98%, and 95% accuracy when dropout regularization was applied. However, the highest accuracy achieved during training was 65%, and 62% for L2 and dropout regularization test, respectively. We showed the in Figures 5.11, 5.12 and 5.13 the result of L2 method wich had better results. To summarize all results for Dataset 1, we presented a summary in Table 5.3. Our study highlights the importance of considering data augmentation techniques to improve the performance of predictive models.

Table 5.3: In the table, various regularization methods of CNN model are compared for different forms of Dataset1.

CNN				
	L1 Reg	**L2 Reg**	**Dropout**	**Test best one on Dataset2**
Without Augmentation	89	90	93	50
With Augmentation (Brightness, Contrast)	39	93	84	62
With Augmentation (Brightness, Contrast, Flipping)	39	98	95	65

CNN on Dataset 2

In this thesis, we also investigated the implementation of the CNN model with different regularization techniques on various forms of Dataset 2. The results of our experiments

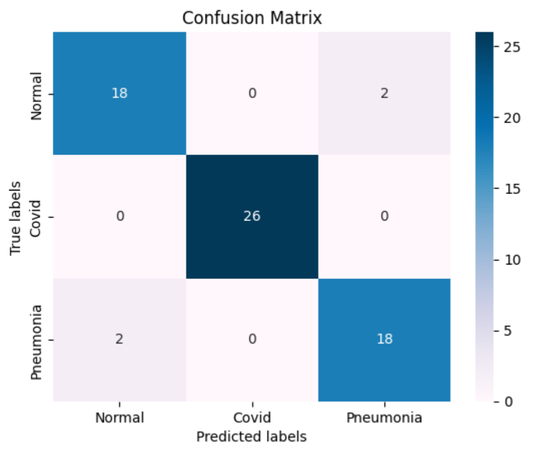


Figure 5.8: This diagram shows the confusion matrix resulting from the CNN model with the L2 regularization algorithm's predictions of the Augmented (Brightness and Contrast) dataset1's test data.

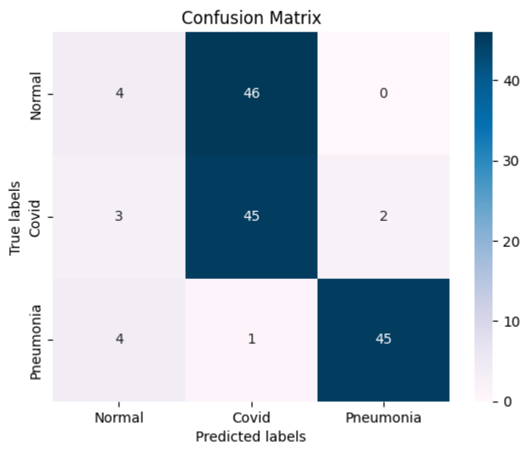


Figure 5.9: This diagram shows the confusion matrix of the CNN model with the L2 regularization algorithm's predictions of dataset2's test data based on Augmented (Brightness and Contrast) dataset1.

are presented in this section. Initially, we applied the CNN model on the original Dataset 2, which was an imbalanced and large dataset. Using L1 regularization, we achieved an accuracy of 69%. Subsequently, we tested the model to Dataset 1, resulting in an accuracy of 54%. This difference in accuracy may be due to the imbalance and size of the original Dataset 2. Therefore, it is essential to consider the characteristics of the dataset while selecting a regularization technique. Additionally, these findings highlight the importance of evaluating the performance of models on different forms of datasets to identify their generalization ability.

Additionally, a Convolutional Neural Network (CNN) model was employed on Dataset 2, which is imbalanced and voluminous. The model was subject to L2 regularization, which resulted in an 89% accuracy. The model was then applied to Dataset 1, producing a 77% accuracy. Following this, we applied a CNN model on the original dataset (Imbalanced and large dataset) and were able to achieve 91% accuracy for Dropout regularization, as shown in Figure 5.14, and 80% accuracy after applying the model on dataset 1, as shown in Figure 5.15. Furthermore, we also provided the history of this training in Figure 5.16.
Further, we have applied all CNN regularization models to the Balanced Dataset 2 and have provided the results of the L1 regularization. This showed a 93% accuracy on the same dataset and an 84% accuracy for testing on Dataset 1. In addition, we provide the results for L2 regularization which showed 94% accuracy on Dataset 2 and 84% accuracy for testing on Dataset 1. On Balanced Dataset 2, we applied the CNN model and obtained 95% accuracy with Dropout regularization illustrated in Figure 5.17. We also acquired 90% accuracy with Dropout regularization on dataset 1, which is displayed in Figure 5.18.

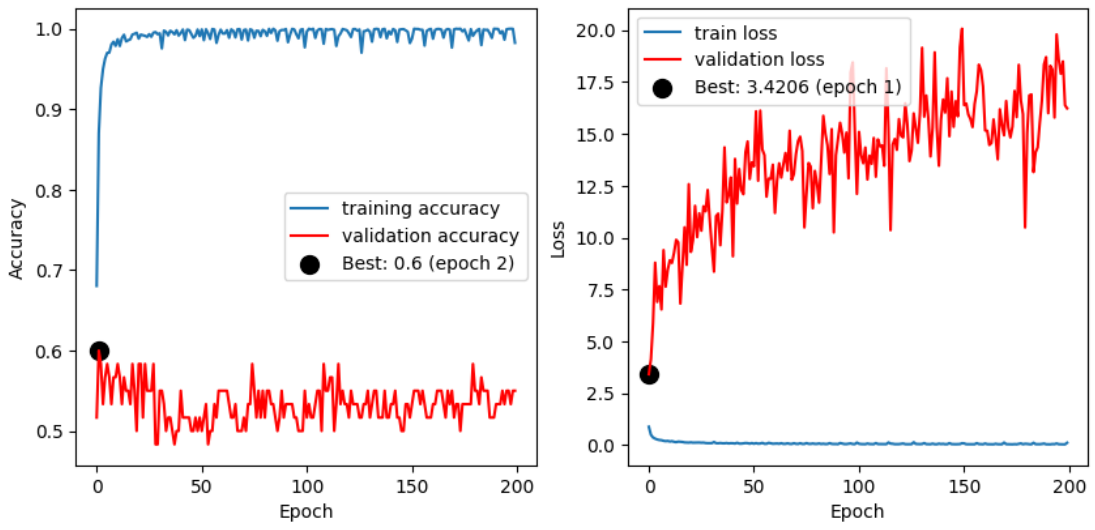


Figure 5.10: This is the loss and accuracy history for CNN with L2 regularization on the Augmented (brightness and contrast) dataset1.

Figure 5.19 shows the history of this training. On Augmented Dataset 2, we used CNN models with different regularization, using Brightness and Contrast, as well as combining Brightness, Contrast, and Flipping. The results of CNN models with L1 regularization on both Augmented datasets were not promising. The results show that the accuracy was reduced by 33 percent for augmented by Brightness and Contrast and augmented by these in addition to Flipping, respectively. As well we have provided the results for L2 regularization on Augmented (Brightness, Contrast) Dataset2, showing the accuracy of 88% on Dataset 2 and 72% on Dataset 1. In the next step, we used the CNN model to analyze Augmented(Brightness, Contrast) Dataset 2, achieving 96% accuracy with Dropout regularization with a Figure 5.21 to show its confusion matrix. In training, the best accuracy was 89%, and 89% accuracy was obtained using Dataset 1 as shown in figure 5.21. A history of this training can also be found in Figure 5.22. On Augmented(Brightness, Contrast, Flipping) Dataset 2, L2 and Dropout regularization provided similar results with 93 and 95 accuracies when training but the best epoch was less than this. When testing on Dataset 1, Dropout gained 90 accuracy, while the L2 gained 89 accuracy. In this study we provided illustration of Dropout regularization because of better results which can be seen in Figures 5.23, 5.24 and 5.25. A summary of the results of regularization methods for the CNN model on dataset 2 has been provided in Table 5.4.

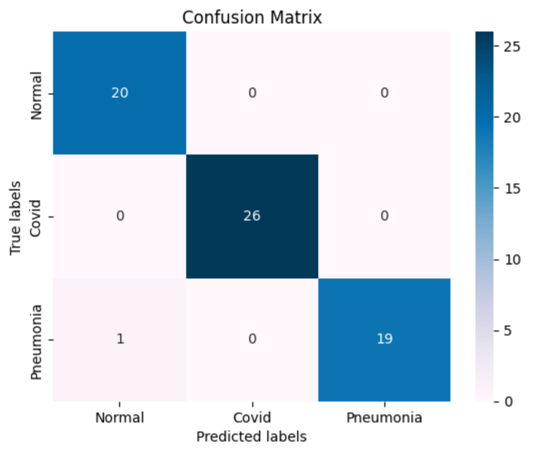


Figure 5.11: This diagram shows the confusion matrix of the CNN model with L2 regularization algorithm's predictions of the Augmented (Brightness and Contrast, Flipping) dataset1.

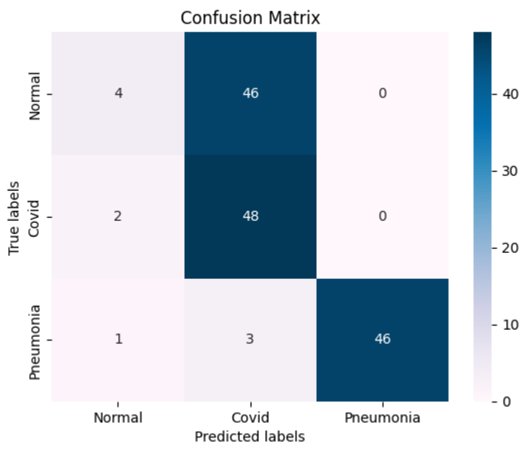


Figure 5.12: This diagram illustrates the confusion matrix of the CNN model with the L2 regularization algorithm's prediction of dataset2's test data based on Augmented (Brightness and Contrast and Flipping) dataset1.

5.5 VGG19

5.5.1 VGG19 on Dataset1

We trained the VGG19 model on the original and each augmented Dataset 1 separately and evaluated the performance on the validation set. The model was trained for 200 epochs with a batch size of 16 using the Adam optimizer and categorical cross-entropy loss. Table 5.5 summarizes the performance of VGG19 on each dataset: In the case of the original and augmented Dataset 1, there was not much difference between them when we applied VGG19 to them. Due to the fact that the accuracy of the original data was 87% as shown in the confusion matrix in Figure 5.26, 86% as shown in Figure 5.29 and 89% as shown in Figure 5.32 for Augmented(Brightness and Contrast) and Augmented (Brightness, Contrast and Flipping) respectively. In spite of this, the best accuracy rates in the training phase were significantly lower than these percentages. It can be seen that the testing results of Dataset 2 and the training are presented in Figures as follows: 5.28 for training on the original Dataset 1, 5.31 for training on the Augmented (Brightness, Contrast) dataset, 5.34 for training on Augmented (Brightness, Contrast, Flipping) dataset, 5.27, 5.30 and 5.33 illustrate the testing on Dataset2 using the training on Dataset1.

5.5.2 VGG19 on Dataset2

We trained the VGG19 model on each dataset separately and evaluated the performance on the validation set. The model was trained for 200 epochs with a batch size of 16 using

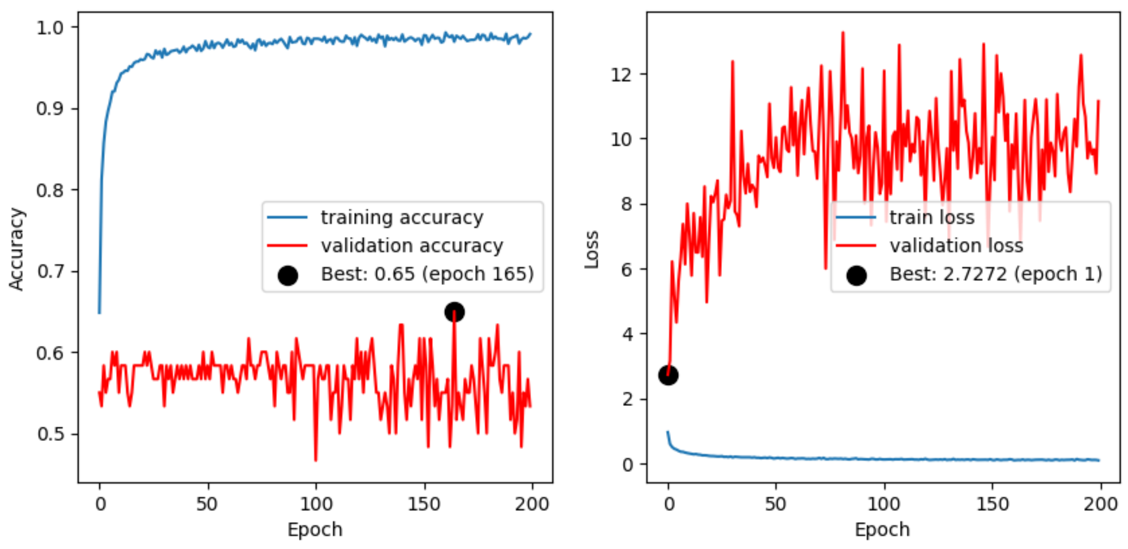


Figure 5.13: This is the accuracy and loss history for CNN with L2 regularization on the Augmented (Brightness, Contrast, and Flipping) dataset 1.

the Adam optimizer and categorical cross-entropy loss. Table 5.6 summarizes the performance of VGG19 on each dataset. The results of using VGG19 on Dataset 2 showed that there was a difference between Imbalanced data and balanced data with reduction and augmentation. With the VGG19 model applied to Imbalanced data, we obtained 76 percent accuracy, shown in Figure 5.35. However, when evaluating the test data of Dataset 1, we obtained 59 percent accuracy, shown in Figure 5.36. There is also a training history provided in Figure 5.37. By reducing the number of normal images and using brightness, contrast, and flipping, VGG19 improved accuracy from 76% to 91-93% when tested on the same dataset as well as from 59% to 81-83% when tested on Dataset1. Figures 5.38, 5.39, and 5.40 explain how to use the model without augmentation on balanced data. Moreover, 5.41, 5.42, and 5.43 describe how they to be applied the model to balanced data with brightening and contrast augmentation. Finally, figures 5.44, 5.45, and 5.46 illustrate how the model was applied to balanced data augmented by brightness, contrast, and flipping.

5.6 Convolutional Auto Encoder

5.6.1 Auto Encoder on Dataset 1

We trained the Convolutional Auto Encoder model on the original and augmented dataset separately and evaluated the performance on the validation set. The model was trained for 200 epochs with a batch size of 16 using the Adam optimizer and categorical cross-entropy loss. The Table 5.7 summarizes the performance of the Convolutional Auto Encoder on Original Dataset 1 and each Augmented Dataset 1. The Auto Encoder algorithms were tested with Original Dataset 1 and Augmented by Brightness and Contrast, and the results

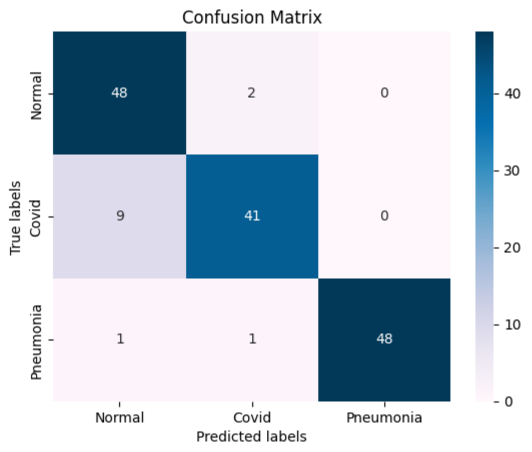


Figure 5.14: This diagram shows the confusion matrix of CNN model predictions using the Dropout regularization algorithm on original Dataset2 test data.

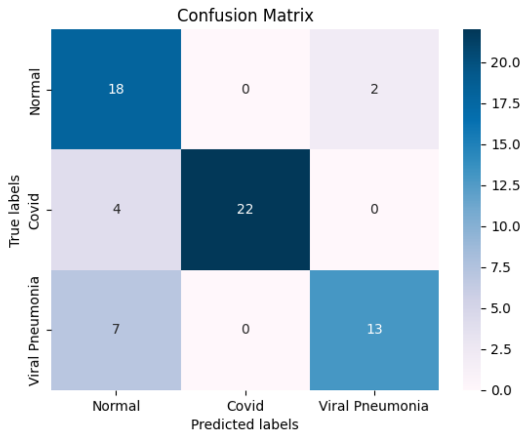


Figure 5.15: On this diagram, we show a confusion matrix representing predictions of Dataset1's test data based on original Dataset2's data using a CNN model with the Dropout regularization algorithm.

were the same, with 87 percent accuracy for the same dataset and 60 percent accuracy on Dataset 2. Adding the flipping method for augmentation did not yield promising results, with 57 percent accuracy for the same data and 35 percent for Dataset 2. Figures 5.47 , 5.48 and 5.49 provide results for the original dataset. Figures 5.50 , 5.51 and 5.52 provide results for Augmented(Brightness, Contrast) dataset. Figures 5.53 , 5.54 and 5.55 provide results for Augmented(Brightness, Contrast , Flipping) dataset.

5.6.2 Auto Encoder on Dataset 2

The results of Convolutional Auto Encoder are summarized in Table 5.8 for Dataset 2 and its Augmented data. Moreover, the results of the Reduced Dataset 2 are shown in Figures5.56, 5.57 and 5.58, and the results of the Augmented by Brightness and Contrast Dataset 2 are provided in Figures 5.59, 5.60 and 5.61. Also, the information of the results for Augmented by Brightness, Contrast, and Flipping is illustrated in Figures 5.62, 5.63 and 5.64. In our experiment, we trained a convolutional autoencoder on a dataset of images intending to reconstruct the input images. The autoencoder was implemented using Keras with a convolutional encoder and decoder architecture, with max pooling and up-sampling layers for dimensionality reduction and expansion. During training, we noticed that the model was overfitting from the first few epochs. The loss on the training set decreased rapidly, while the loss on the validation set remained relatively constant. We monitored the training progress using the training and validation loss metrics and found that the model achieved a training loss of 0.05 and a validation loss of 0.20 after just five epochs. We analyzed the model's performance on a held-out test set of images to investigate the overfitting issue. We found that the model's reconstruction performance on the

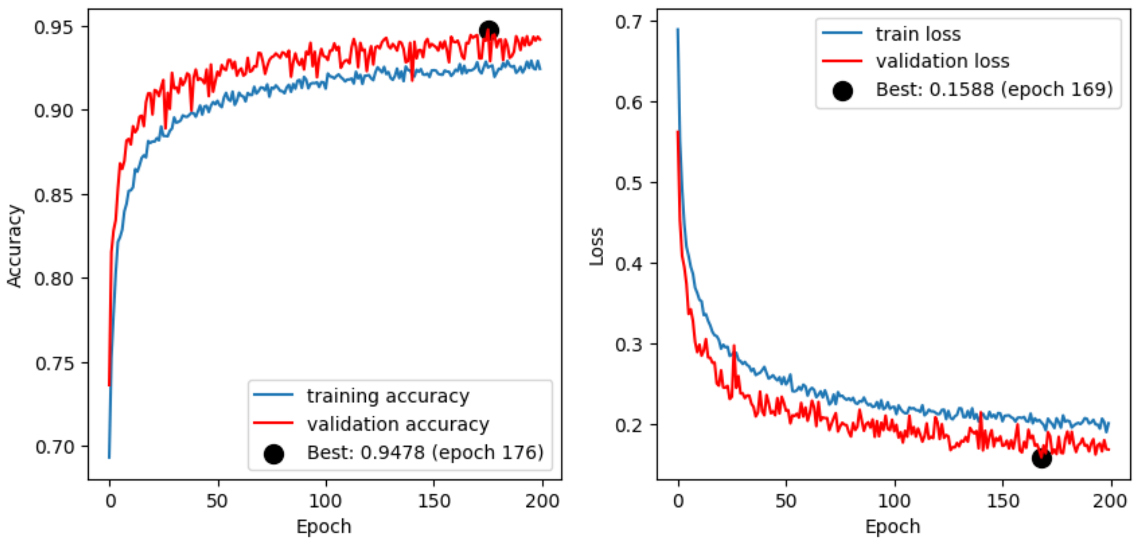


Figure 5.16: This is the history of loss and accuracy for CNN with Dropout regularization on original Dataset 2, including the best epoch for loss and accuracy for each epoch.

test set was much worse than its performance on the training set, indicating that the model had learned to memorize the training data rather than learn valuable features that generalize to new data. In conclusion, our experiment showed that convolutional autoencoders could overfit from the first few epochs, and the model is a complex training dataset size. Regularization techniques may only sometimes effectively mitigate overfitting, and reducing model complexity may result in decreased performance. Therefore, it is essential to carefully balance model complexity and regularization to achieve optimal performance and generalization in image classification tasks.

Using all the models we used and all the validations performed on them, we concluded that the large dataset helped improve the models' performance since, as we can see from the tables, accuracy decreased when performing on Dataset 2. In addition, Convolutional Auto Encoding and Convolutional Neural Networks were the most effective models in the analysis. However, we could see that there was overfitting in the first epochs. However, we could see that using Dataset 2 instead of Dataset 1 as a training set could improve COVID detection in Dataset 1. The Convolutional Auto Encoding and Convolutional Neural Networks had better results due to their ability to detect data patterns and make more accurate predictions. The overfitting issue could not be addressed by using Dataset 2 as the training set, which allowed for the model to better detect COVID in Dataset 1 by learning more relevant features.

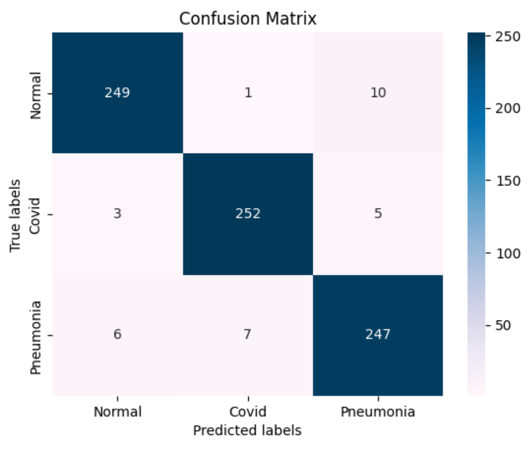


Figure 5.17: This diagram shows the confusion matrix generated by the CNN model with the Dropout regularization algorithm's predictions of the Balanced Dataset2 test data.

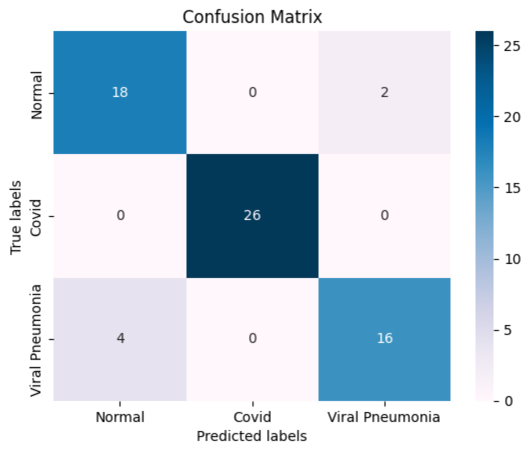


Figure 5.18: This diagram shows the confusion matrix generated by CNN model with the Dropout regularization algorithm's predictions of Dataset1's test data based on Balanced Dataset2.

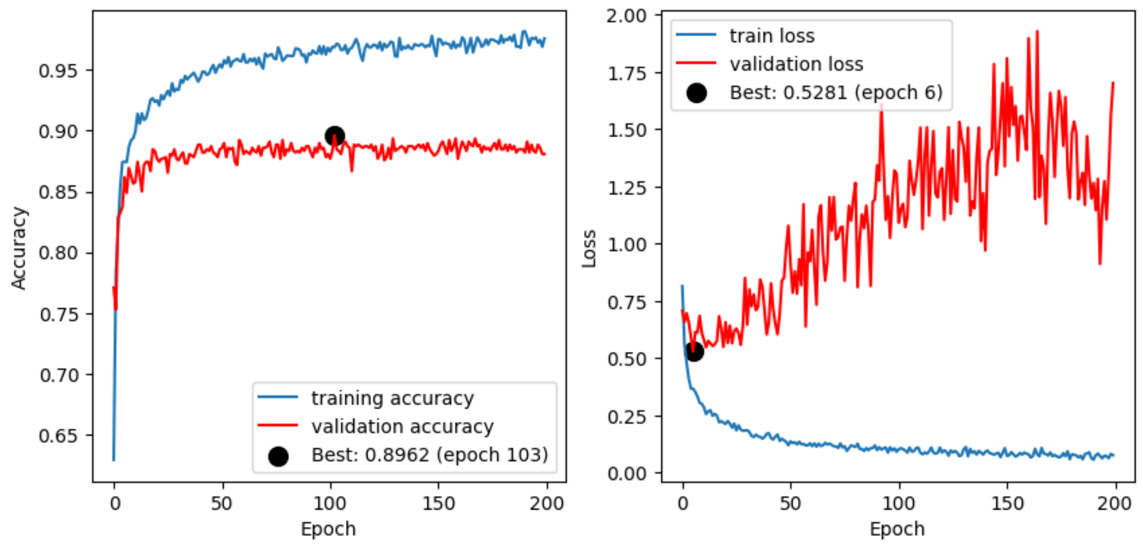


Figure 5.19: This is the history of loss and accuracy for each epoch, including the best accuracy and loss epoch for CNN with Dropout regularization on the original Balanced Dataset 2.

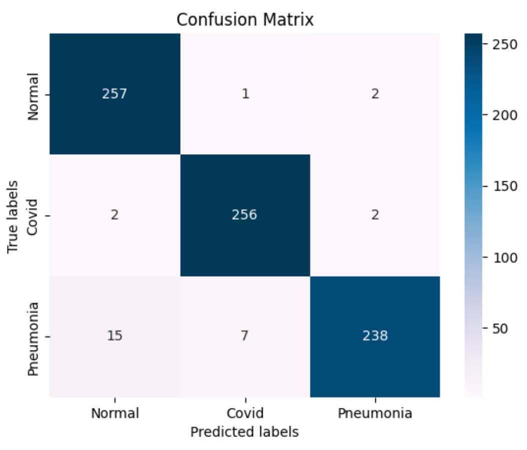


Figure 5.20: This diagram shows the confusion matrix generated by the CNN model with the Dropout regularization algorithm's predictions of the Augmented (Brightness and Contrast) Dataset2 test data.

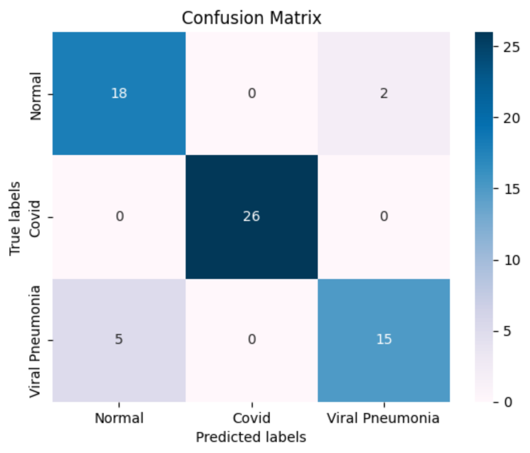
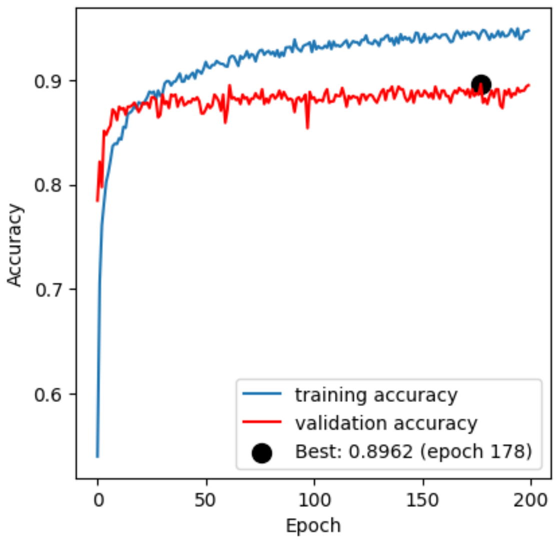


Figure 5.21: This diagram shows the confusion matrix generated by the CNN model with the Dropout regularization algorithm's predictions of Dataset1's test data based on the Augmented (Brightness and Contrast) Dataset2.



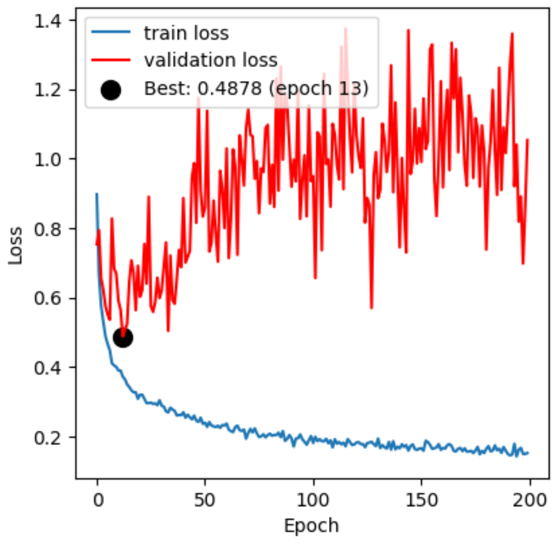


Figure 5.22: This is the history of loss and accuracy for each epoch, including the best accuracy and loss epoch for CNN with Dropout regularization on the Augmented (Brightness and Contrast) Dataset 2.

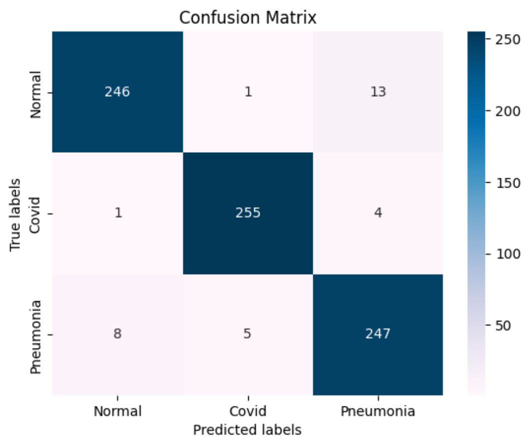


Figure 5.23: This diagram shows the confusion matrix generated by the CNN model with the Dropout regularization algorithm's predictions of the Augmented (Brightness, Contrast, and Flipping) Dataset2 test data.

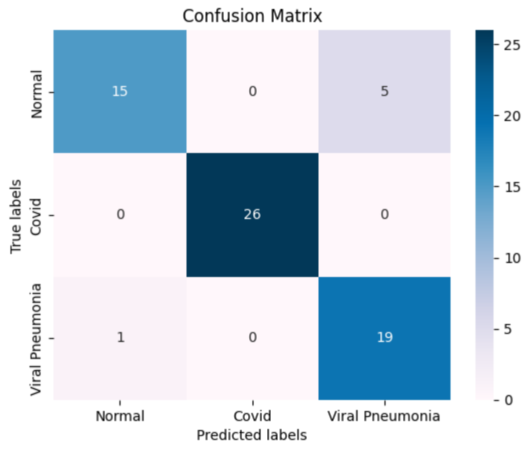
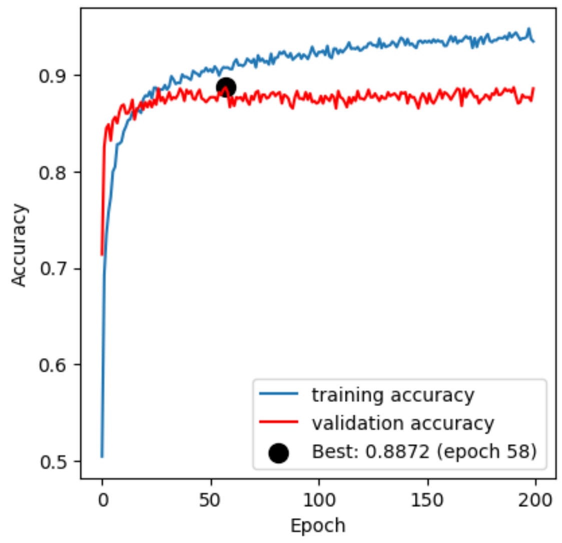


Figure 5.24: This diagram shows the confusion matrix generated by the CNN model with the Dropout regularization algorithm's predictions of Dataset1's test data based on Augmented (Brightness, Contrast, and Flipping) Dataset2.



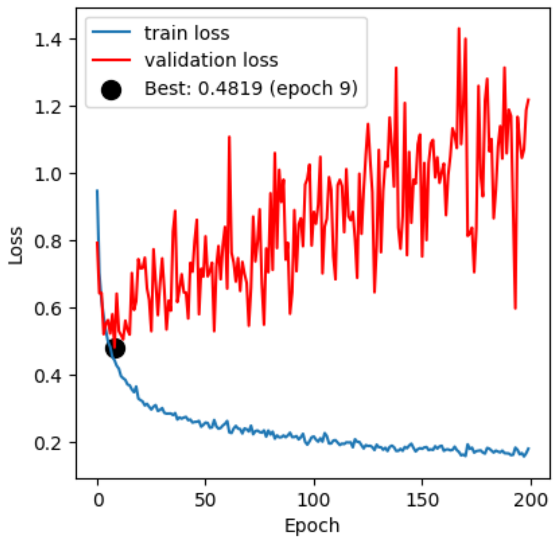


Figure 5.25: This is the history of loss and accuracy for each epoch, including the best accuracy and loss epoch for CNN with Dropout regularization on the Augmented (Brightness, Contrast, and Flipping) Dataset 2.

Table 5.4: In the table, various regularization methods of CNN are compared for different forms of Dataset2

CNN				
	L1 Reg	**L2 Reg**	**Dropout**	**Test best one on Dataset1**
Without Augmentation (Imbalanced)	69	89	91	80
Without Augmentation (Balanced)	93	94	95	90
With Augmentation (Brightness, Contrast)	33	88	96	89
With Augmentation (Brightness, Contrast, Flipping)	30	92	95	90

Table 5.5: This table presents the summary of the VGG19 model for Dataset1 and its augmented data with the main features of the model.

VGG19		
Dataset1	**Accuracy**	**Test on Dataset2**
Without Augmentation	87	58
With Augmentation (Brightness, Contrast)	86	58
With Augmentation (Brightness, Contrast, Flipping)	89	54

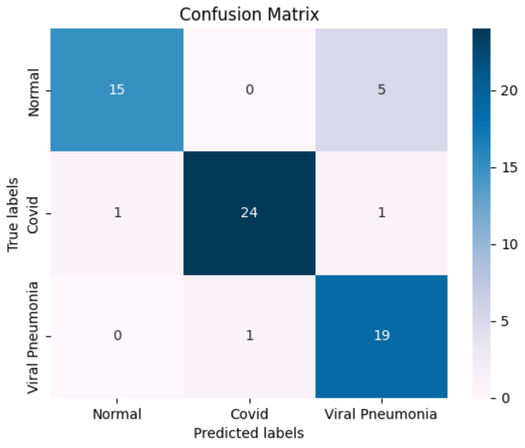


Figure 5.26: The diagram shows the confusion matrix of VGG19's predictions for Dataset 1's test data.

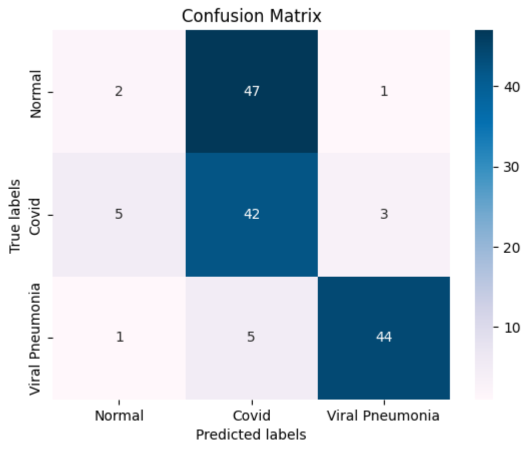


Figure 5.27: A confusion matrix is displayed in this diagram showing the VGG19 algorithm's predictions of Dataset2's test data based on Original Dataset 1.

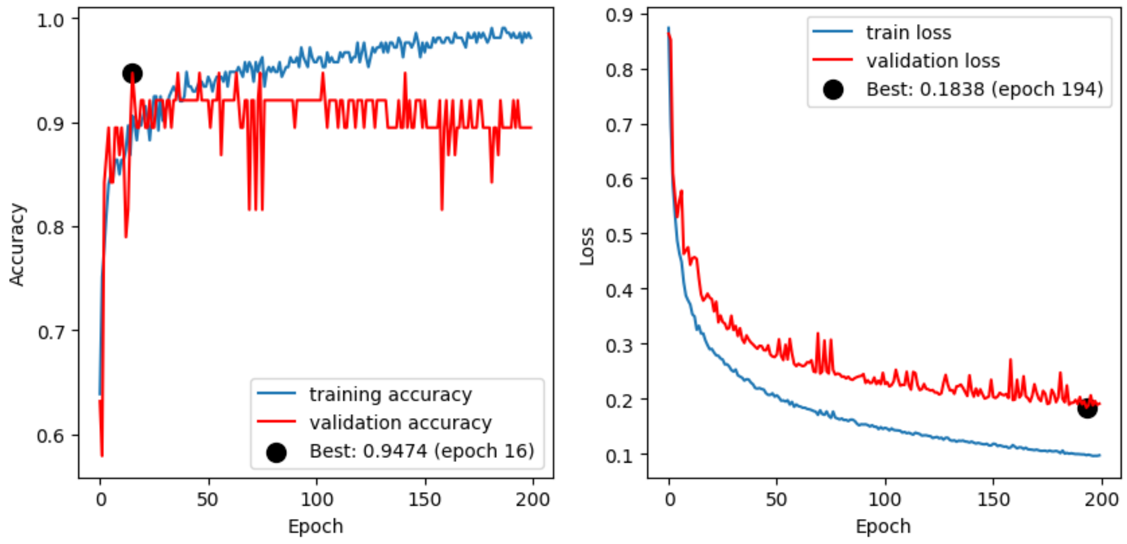


Figure 5.28: A history of loss and accuracy for each epoch, including the best accuracy epoch and loss epoch for the VGG19 model on Original Dataset1.

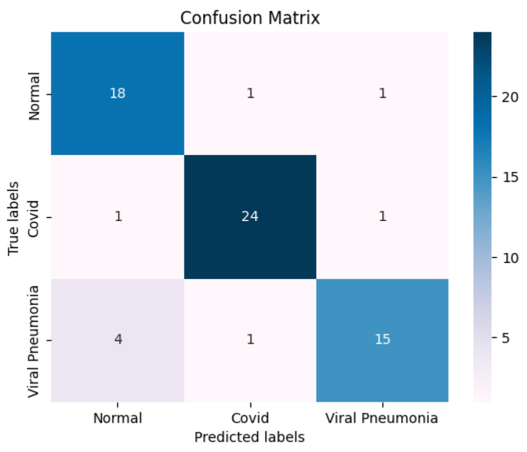


Figure 5.29: The diagram shows the confusion matrix of VGG19's predictions for Augmented (Brightness and Contrast) Dataset 1's test data.

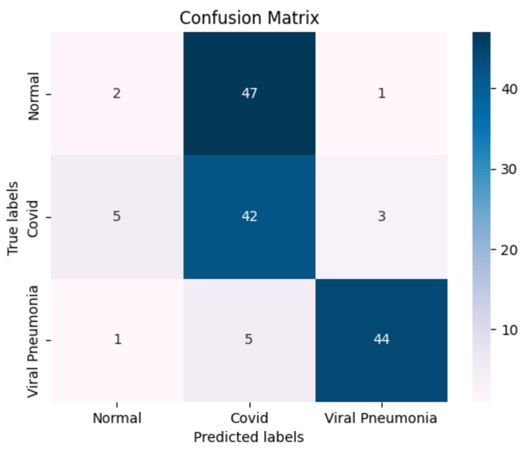


Figure 5.30: A confusion matrix is displayed in this diagram showing the VGG19 algorithm's predictions of Dataset2's test data based on Augmented (Brightness and Contrast) Dataset 1.

Table 5.6: The table shows the summary of the VGG19 model on Dataset 2 and its augmented data.

VGG19		
Dataset2	**VGG19**	**Test on Dataset1**
Without Augmentation(Imbalanced)	76	59
Without Augmentation (Blanaced)	91	81
With Augmentation (Brightness, Contrast)	92	80
With Augmentation (Brightness, Contrast, Flipping)	93	83

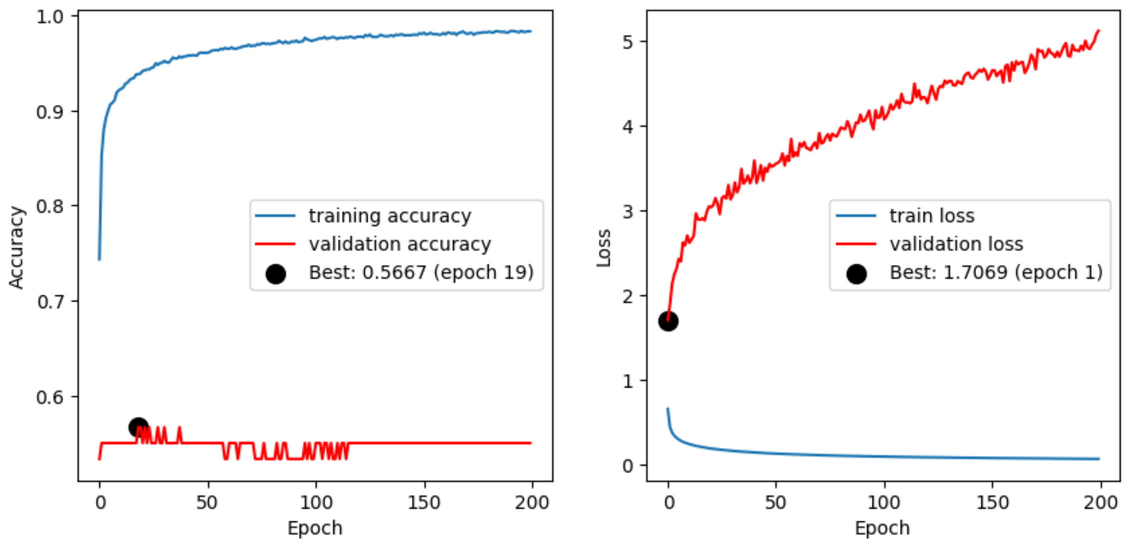


Figure 5.31: A history of loss and accuracy for each epoch, including the best accuracy epoch and loss epoch for the VGG19 model on Augmented (Brightness and Contrast) Dataset1.

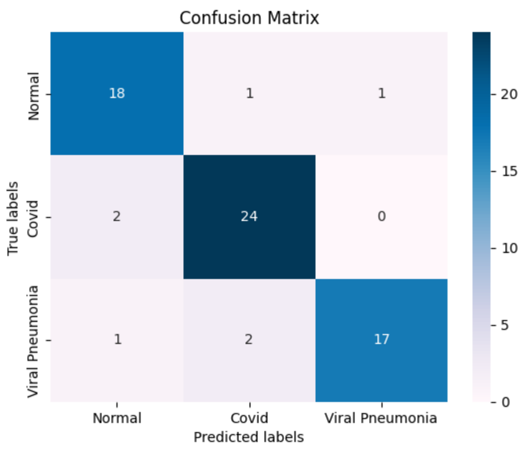


Figure 5.32: The diagram shows the confusion matrix of VGG19's predictions for Augmented (Brightness, Contrast, and Flipping) Dataset1's test data.

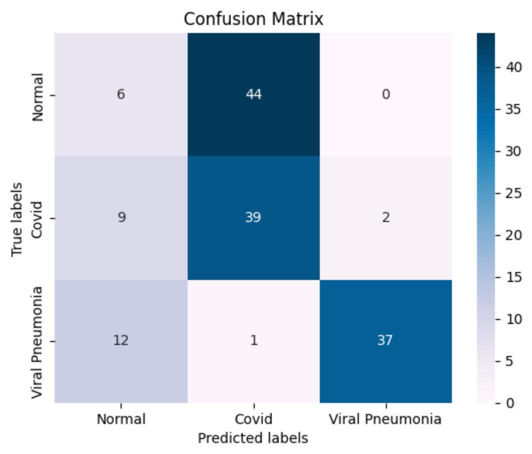


Figure 5.33: A confusion matrix is displayed in this diagram showing the VGG19 algorithm's predictions of Dataset2's test data based on Augmented (Brightness, Contrast, and Flipping) Dataset 1.

Table 5.7: The table shows the summary of the Convolutional Auto Encoder model on Dataset 1 and its augmented data.

Convolutional Auto Encoder		
Dataset1	**Accuracy**	**Test on Dataset2**
Original	87	61
With Augmentation (Brightness, Contrast)	87	62
With Augmentation (Brightness, Contrast, Flipping)	90	61

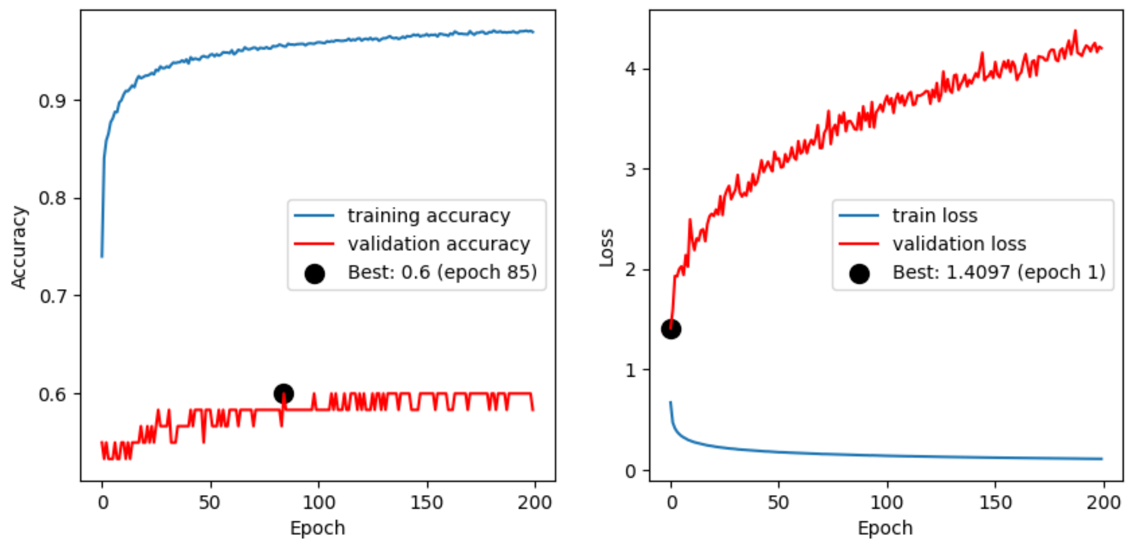


Figure 5.34: A history of loss and accuracy for each epoch, including the best accuracy epoch and loss epoch for the VGG19 model on Augmented (Brightness, Contrast and Flipping) Dataset1.

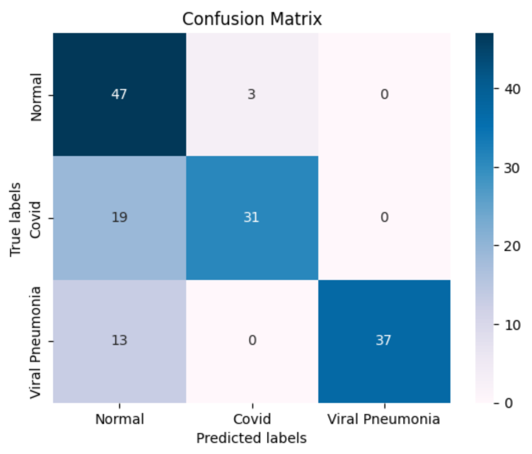


Figure 5.35: This diagram shows the confusion matrix of the VGG19 model's predictions for Original Dataset2.

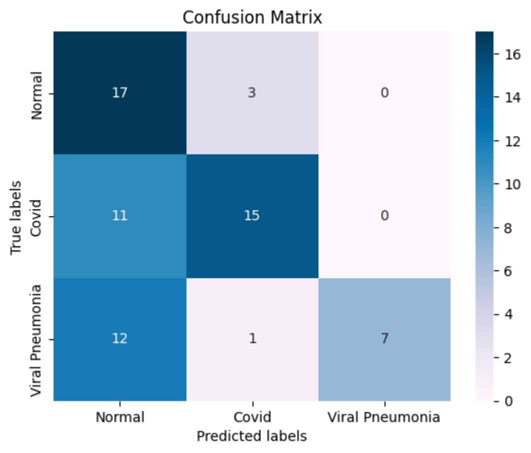


Figure 5.36: The confusion matrix presented here shows the VGG19 algorithm's predictions based on Original Dataset2's data for Dataset1's test data.

Table 5.8: The table shows the summary of the Convolutional Auto Encoder model on Dataset 2 and its augmented data.

Convolutional Auto Encoder		
Dataset2	**Accuracy**	**Test on Dataset1**
Balanced (Reduced)	95	84
With Augmentation (Brightness, Contrast)	93	83
With Augmentation (Brightness, Contrast, Flipping)	94	84

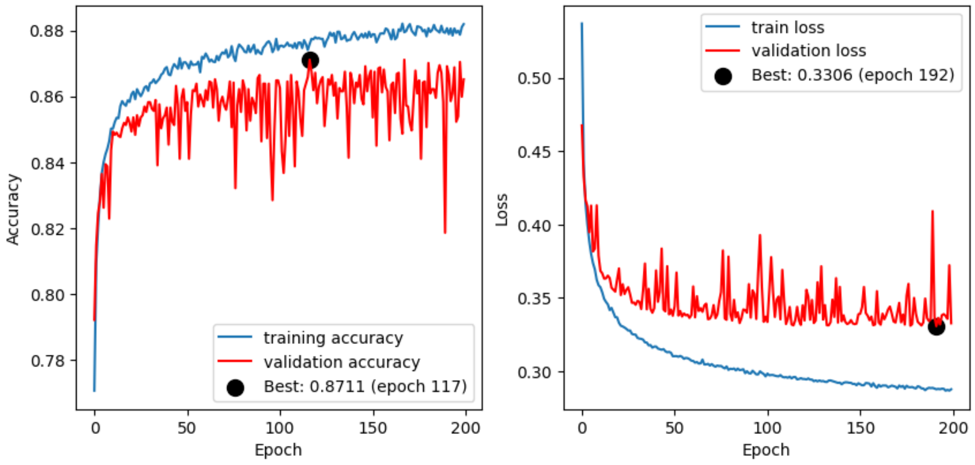


Figure 5.37: This is the history of loss and accuracy on Original Dataset2 for each epoch, including the best epoch for accuracy and loss.

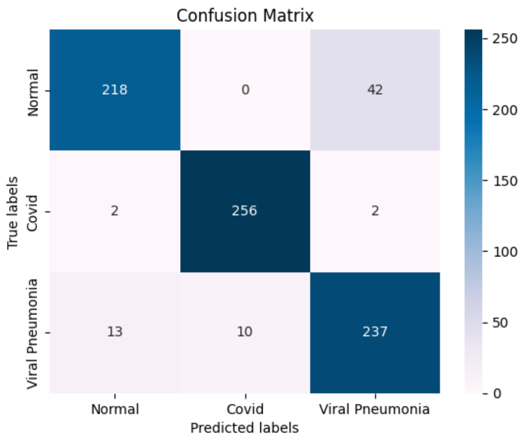


Figure 5.38: This diagram shows the confusion matrix of the VGG19 model's predictions for Balanced Dataset2.

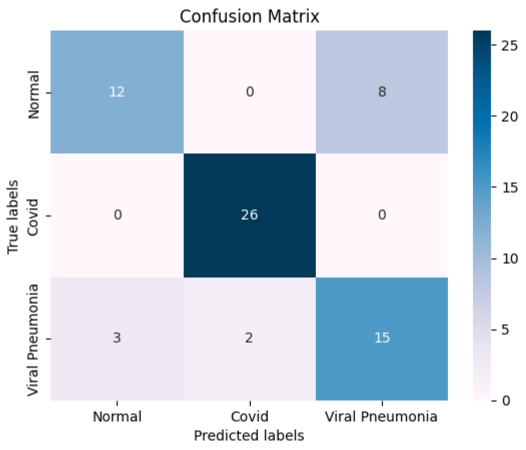


Figure 5.39: The confusion matrix presented here shows the VGG19 algorithm's predictions based on Balanced Dataset2's data for Dataset1's test data.

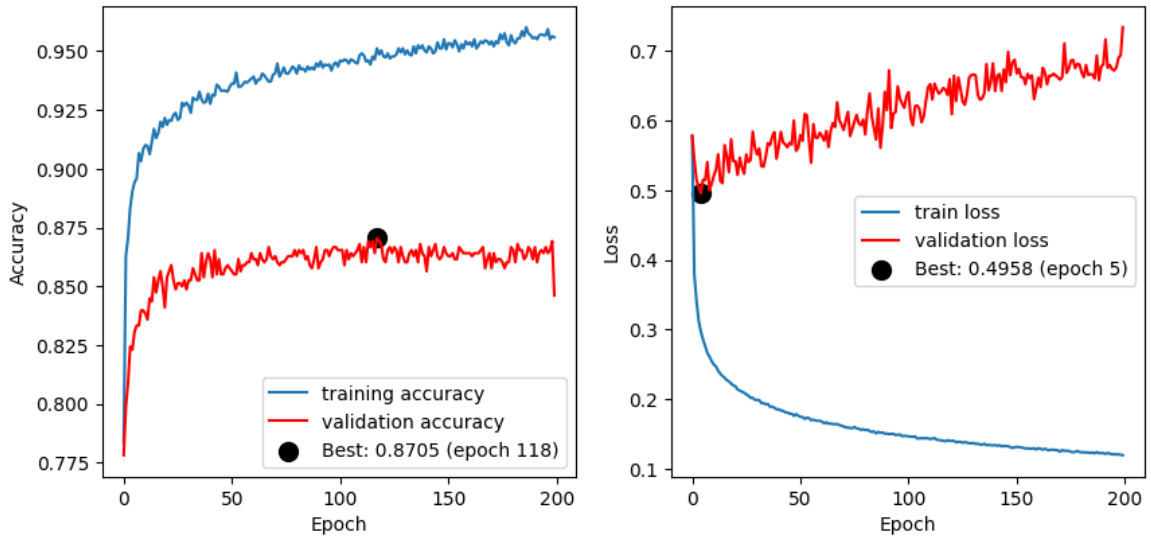


Figure 5.40: This is the history of loss and accuracy on Balanced Dataset2 for each epoch, including the best epoch for accuracy and loss.

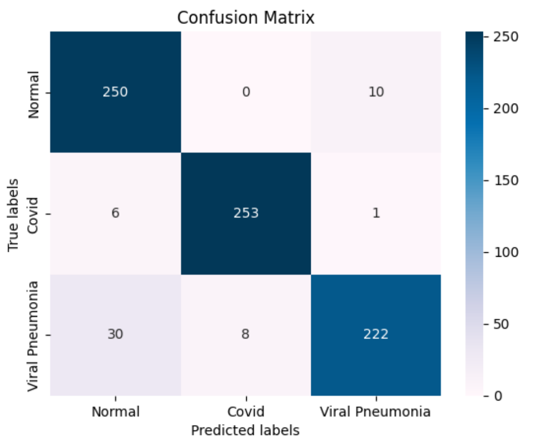


Figure 5.41: This diagram shows the confusion matrix of the VGG19 model's predictions for Augmented (Brightness and Contrast) Dataset2.

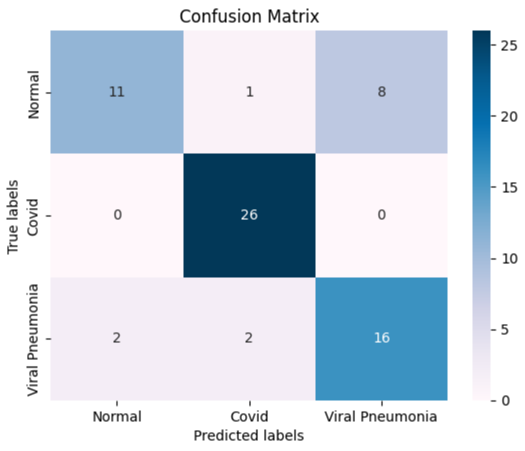


Figure 5.42: The confusion matrix presented here shows the VGG19 algorithm's predictions based on Augmented (Brightness and Contrast) Dataset2's data for Dataset1's test data.

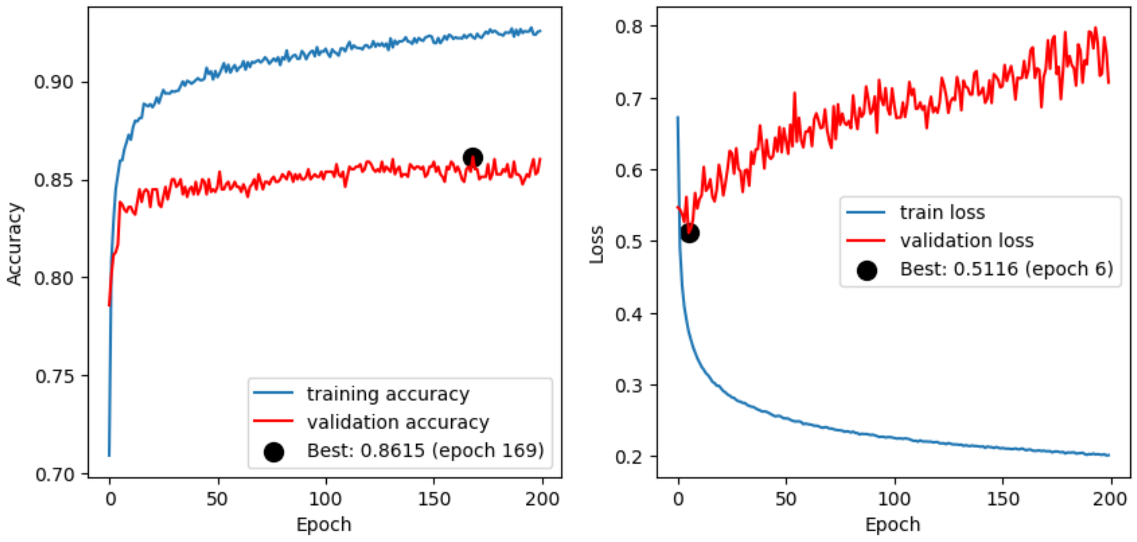


Figure 5.43: This is the history of loss and accuracy on Augmented (Brightness and Contrast) Dataset2 for each epoch, including the best epoch for accuracy and loss.

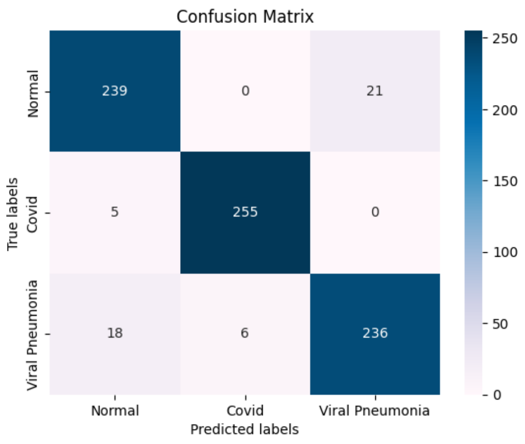


Figure 5.44: This diagram shows the confusion matrix of the VGG19 model's predictions for Augmented (Brightness, Contrast, and Flipping) Dataset2.

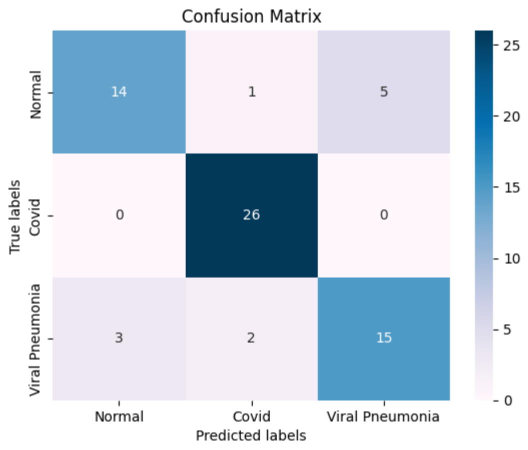


Figure 5.45: The confusion matrix presented here shows the VGG19 algorithm's predictions based on Augmented (Brightness, Contrast, and Flipping) Dataset2's data for Dataset1's test data.

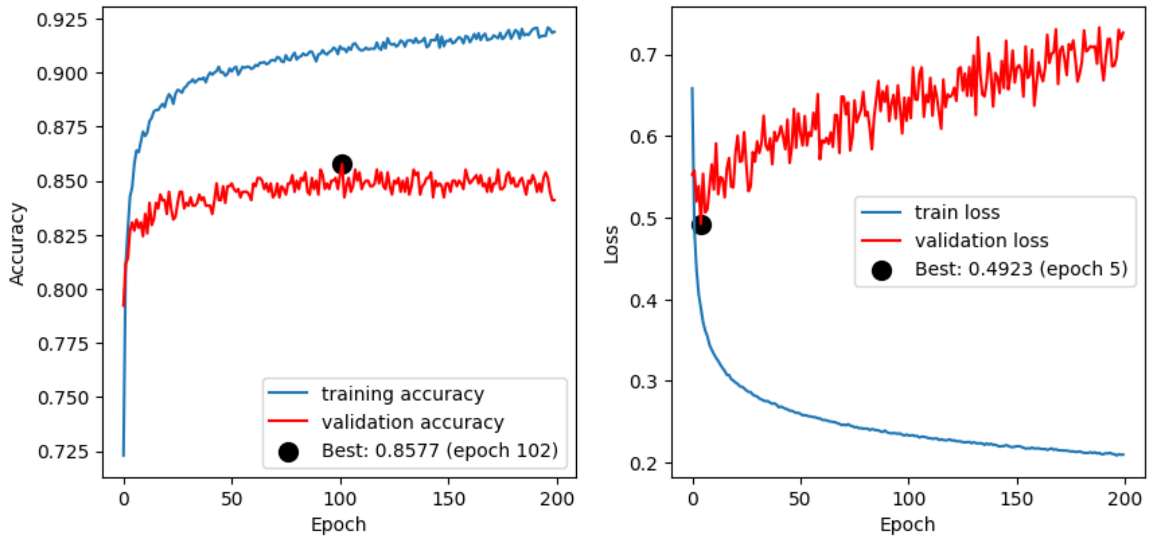
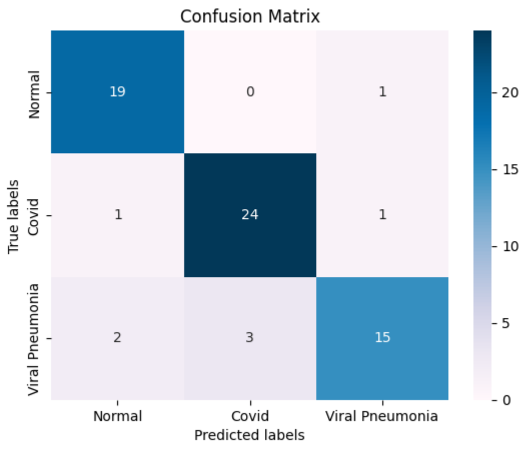
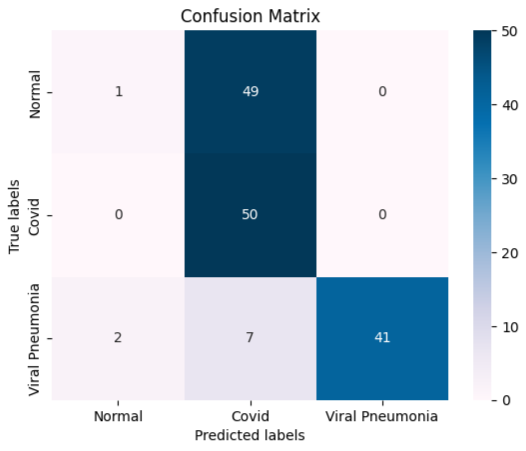


Figure 5.46: This is the history of loss and accuracy on Augmented (Brightness, Contrast, and Flipping) Dataset2 for each epoch, including the best epoch for accuracy and loss.



	Normal	Covid	Viral Pneumonia
Normal	19	0	1
Covid	1	24	1
Viral Pneumonia	2	3	15

Figure 5.47: In this diagram, we show the confusion matrix of the Auto Encoder model algorithm's predictions of original Dataset 1 test data.



	Normal	Covid	Viral Pneumonia
Normal	1	49	0
Covid	0	50	0
Viral Pneumonia	2	7	41

Figure 5.48: We show in this diagram the confusion matrix of the Auto Encoder model algorithm's predictions of Dataset2's test data based on the original Dataset1's data.

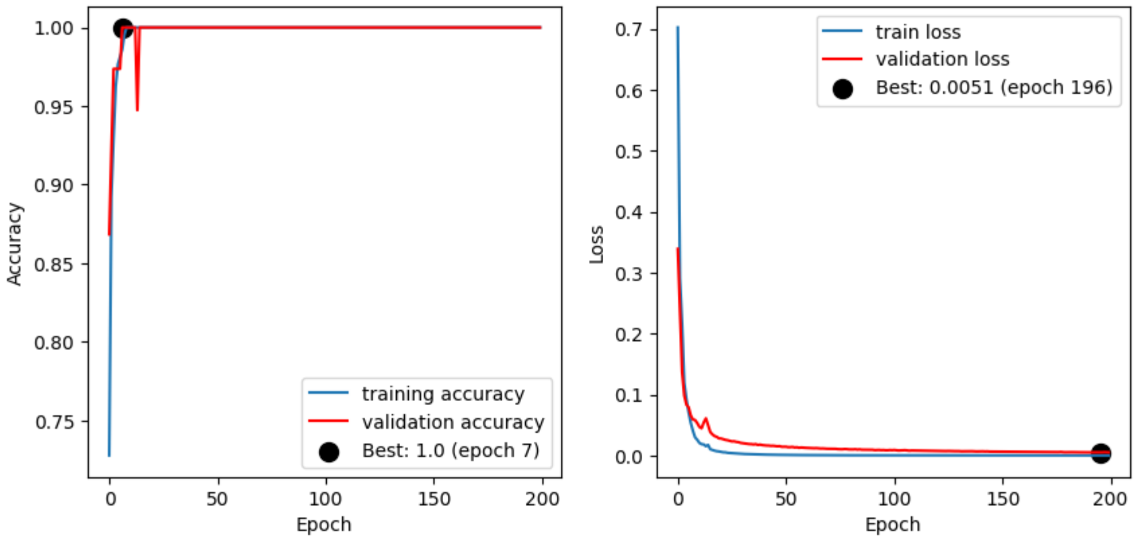


Figure 5.49: This is the history of loss and accuracy for each epoch, including the best accuracy and loss epoch for the Auto Encoder model on original Dataset1.

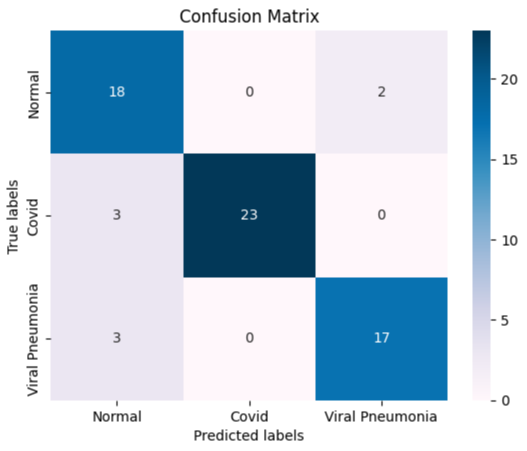


Figure 5.50: In this diagram, we show the confusion matrix of the Auto Encoder model algorithm's predictions of Augmented (Brightness and Contrast) Dataset1 test data.

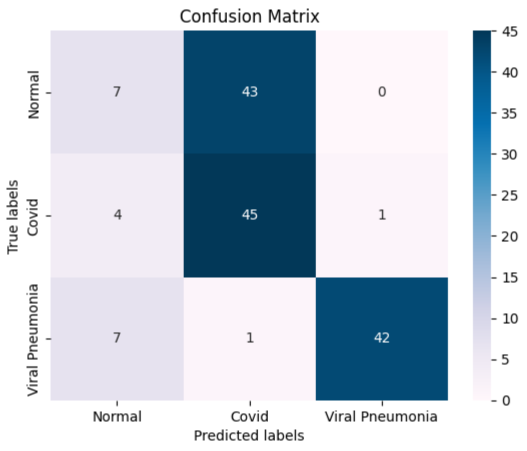


Figure 5.51: We show in this diagram the confusion matrix of the Auto Encoder model algorithm's predictions of Dataset2's test data based on Augmented (Brightness and Contrast) Dataset1's data.

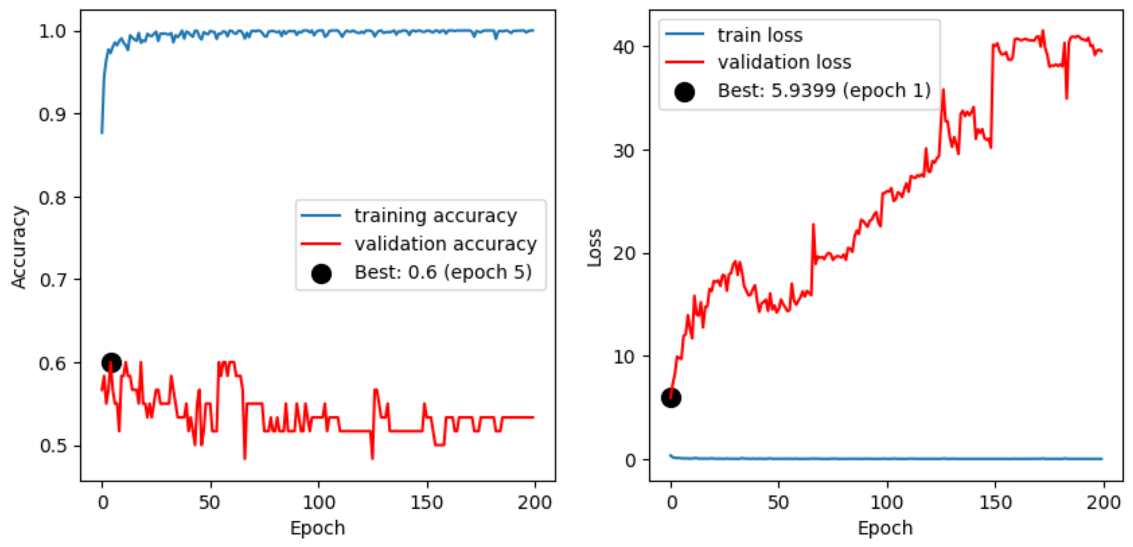


Figure 5.52: This is the history of loss and accuracy for each epoch, including the best accuracy and loss epoch for the Auto Encoder model on Augmented (Brightness and Contrast) Dataset1.

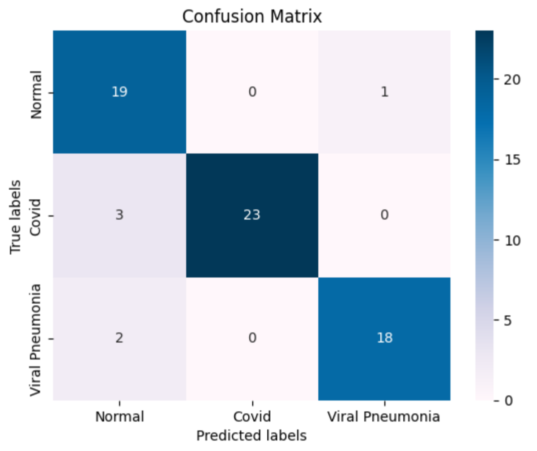


Figure 5.53: In this diagram, we show the confusion matrix of the Auto Encoder model algorithm's predictions of Augmented (Brightness, Contrast and Flipping) Dataset 1 test data.

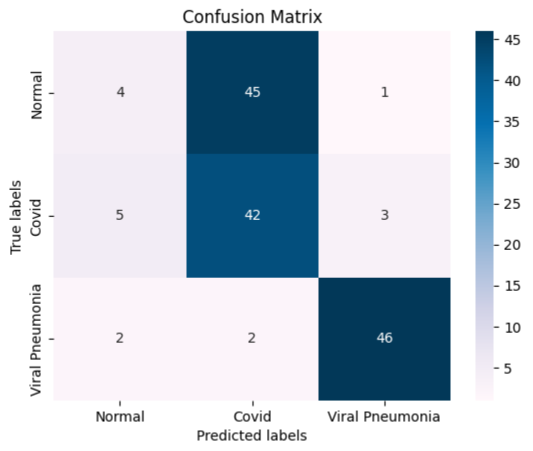


Figure 5.54: We show in this diagram the confusion matrix of the Auto Encoder model algorithm's predictions of Dataset2's test data based on Augmented (Brightness, Contrast, and Flipping) Dataset1's data.

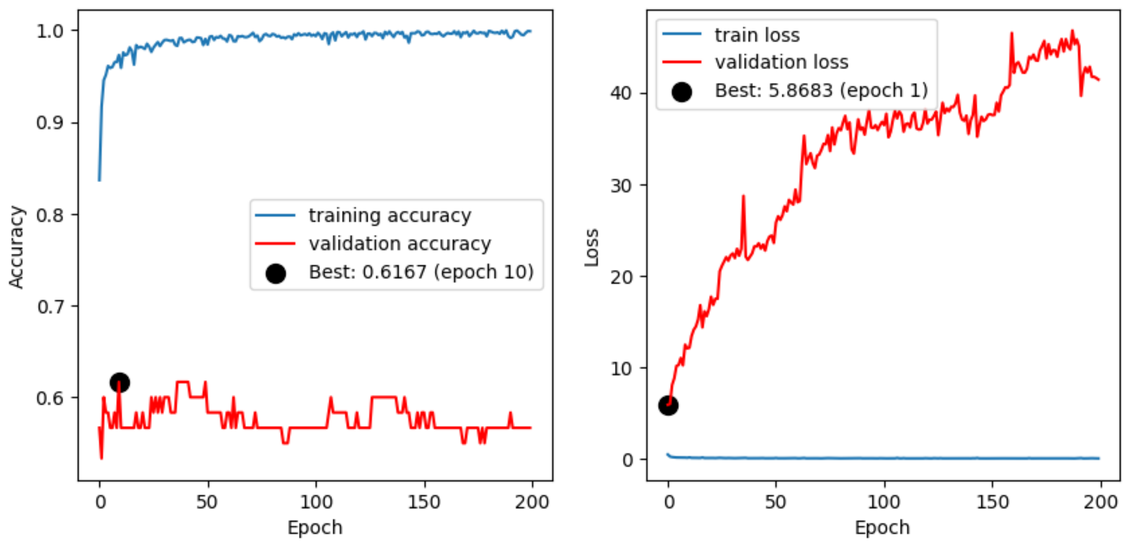
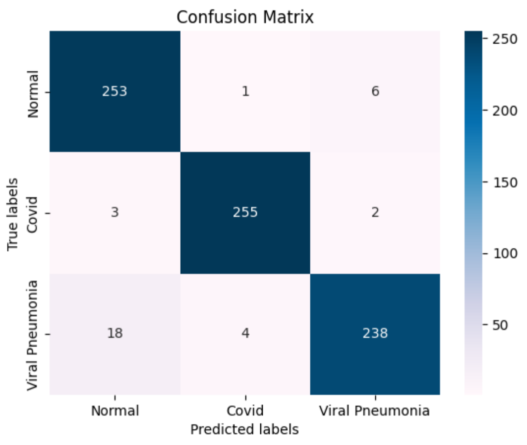
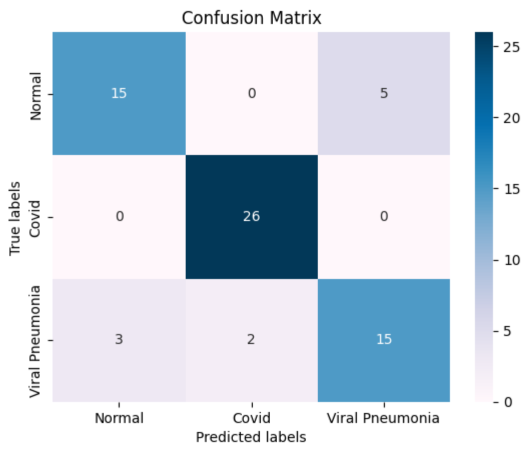


Figure 5.55: This is the history of loss and accuracy for each epoch, including the best accuracy and loss epoch for the Auto Encoder model on Augmented (Brightness, Contrast, and Flipping) Dataset1.



	Normal	Covid	Viral Pneumonia
Normal	253	1	6
Covid	3	255	2
Viral Pneumonia	18	4	238

Figure 5.56: In this diagram, we show the confusion matrix of the Auto Encoder model algorithm's predictions of Reduced Dataset2 test data.



	Normal	Covid	Viral Pneumonia
Normal	15	0	5
Covid	0	26	0
Viral Pneumonia	3	2	15

Figure 5.57: We show in this diagram the confusion matrix of the Auto Encoder model algorithm's predictions of Dataset1's test data based on Reduced Dataset2's data.

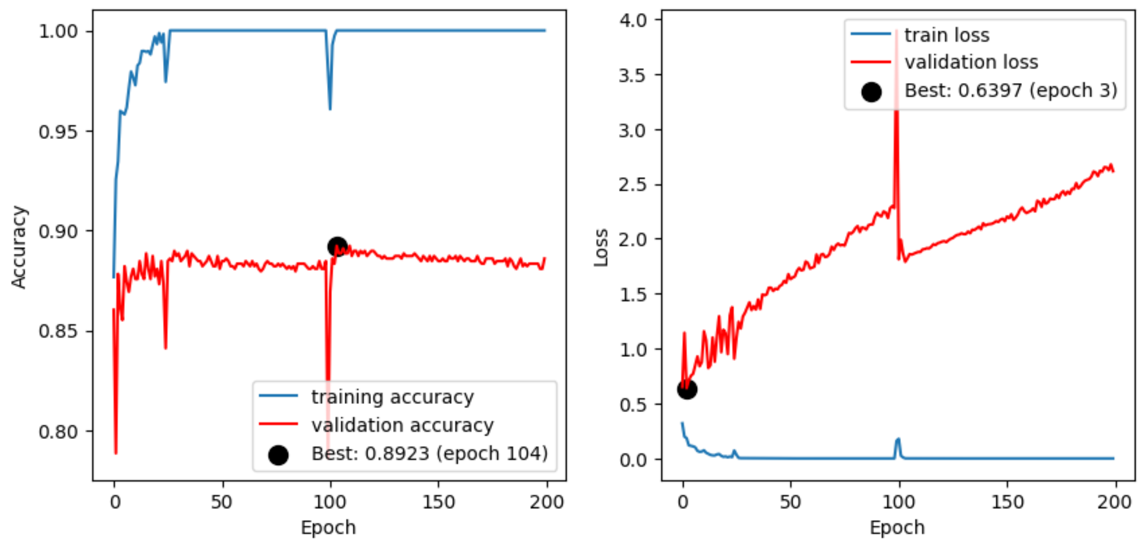


Figure 5.58: This is the history of loss and accuracy for each epoch, including the best accuracy and loss epoch for the Auto Encoder model on Reduced Dataset2.

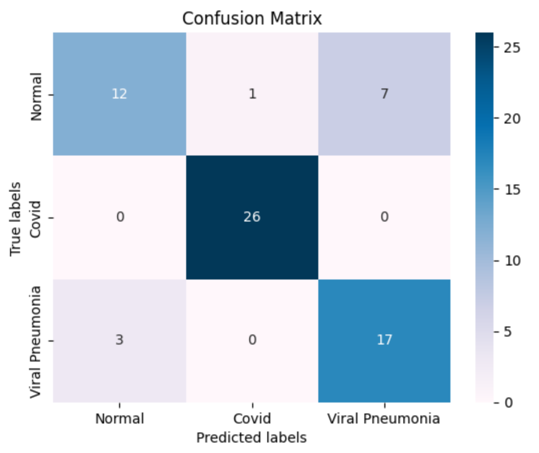


Figure 5.59: In this diagram, we show the confusion matrix of the Auto Encoder model algorithm's predictions of Augmented (Brightness and Contrast) Dataset2 test data.

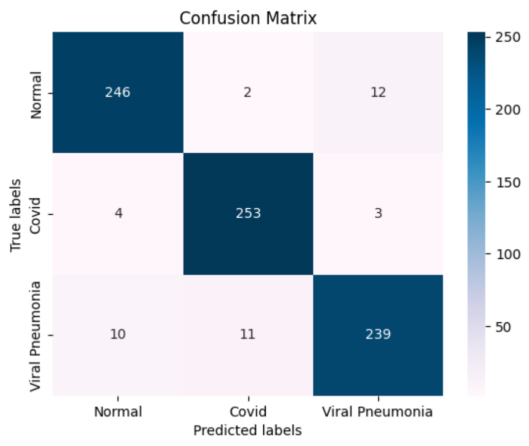


Figure 5.60: We show in this diagram the confusion matrix of the Auto Encoder model algorithm's predictions of Dataset1's test data based on Augmented (Brightness and Contrast) Dataset2's data.

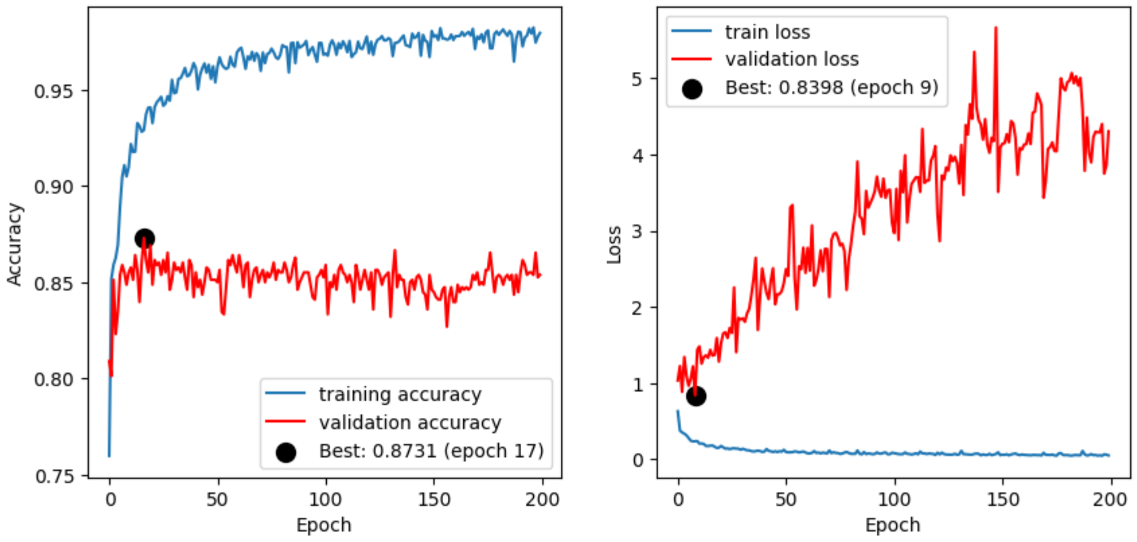


Figure 5.61: This is the history of loss and accuracy for each epoch, including the best accuracy and loss epoch for the Auto Encoder model on Augmented (Brightness and Contrast) Dataset2.

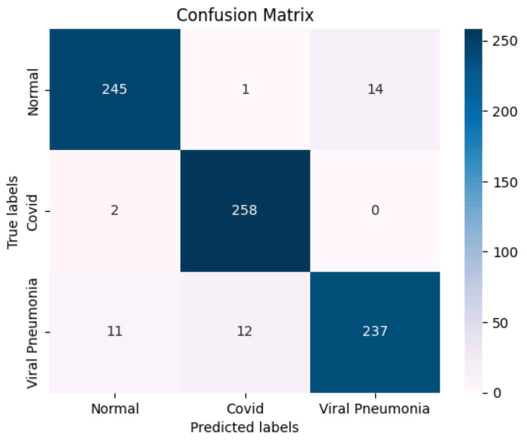


Figure 5.62: In this diagram, we show the confusion matrix of the Auto Encoder model algorithm's predictions of Augmented (Brightness, Contrast, and Flipping) Dataset 2 test data.

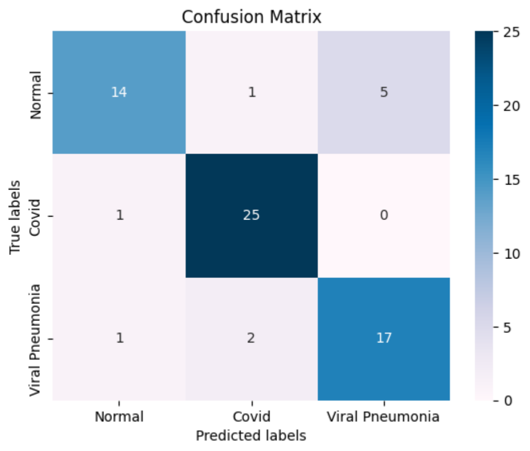


Figure 5.63: We show in this diagram the confusion matrix of the Auto Encoder model algorithm's predictions of Dataset1's test data based on Augmented (Brightness, Contrast, and Flipping) Dataset2's data.

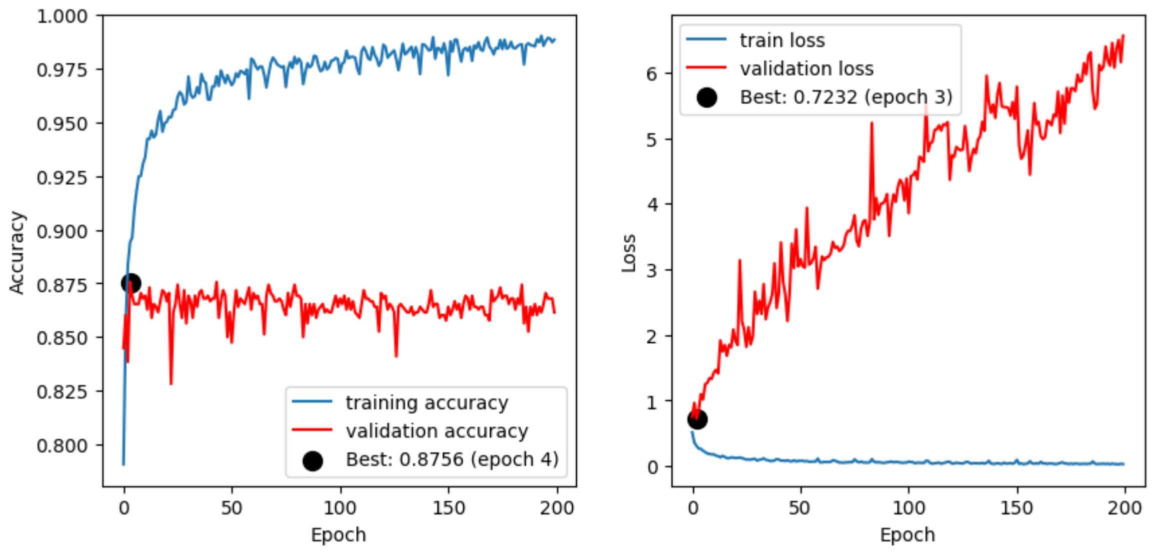


Figure 5.64: This is the history of loss and accuracy for each epoch, including the best accuracy and loss epoch for the Auto Encoder model on Augmented (Brightness, Contrast, and Flipping) Dataset2.

Chapter 6

Discussion

Recently, machine learning and deep learning models have attracted significant attention for the detection of COVID-19 using chest X-rays. The results from various studies suggest that these models have the potential to detect COVID-19 accurately with high sensitivity and specificity, making them useful for diagnosing and treating COVID-19. One of the advantages of using these models is that they can analyze different data with large amounts. They can identify patterns that may be difficult for human experts to detect. This allows for the rapid and accurate identification of COVID-19 cases, which is particularly important in situations where access to testing facilities is limited or where there is a need for rapid diagnosis. In the case of COVID-19 detection, using chest X-ray images, deep learning, and machine learning models also presented several challenges in this study. We encountered some challenges when testing a trained model on another dataset set from a different source due to the fact that we were using different sources of data sets. Here are some challenges mentioned. Since we used two different datasets from two sources, we had some issues when training on one dataset and testing on another from a different source. In testing the model on another dataset, the sensitivity decremented too much for Normal Chest X-ray images, and they were misclassified as COVID-19 images. Several ways can solve this problem, including data augmentation, fine-tuning, regularization techniques, ensemble models, and hyperparameter tuning. As a result of using these methods, we were able to improve performance in some cases. Data regularization with dropout, for instance, could improve performance, but the data augmentation did not work for Dataset 1, which was a small dataset. A second challenge was overfitting, which we had, especially in Dataset 1, since we knew that overfitting was bound to occur in such a small dataset and also that data augmentation did not help us much with this problem for dataset 1. However, in Dataset 2, we were able to train models and prevent overfitting for more epochs since more images were available for training. However, overfitting was not completely eliminated in Dataset 2. Thirdly, standardized image acquisition protocols, different formats of images in datasets, and different annotation protocols can result in significant differences in the quality and consistency of the data used to train the models. If the dataset used for training is too small or biased, the models may also be prone to overfitting. We had this problem with Dataset 1. There was also a problem with image dimension when training. We tried different image dimensions to train, but the image with (50,50,3) was able to train the model better and result in better results, while the image with (100,100,3) could not. Additionally, this information was about dataset one, which was small. However, when we wanted to apply this dimension (100,100,3) to datasets, we also encountered problems with the GPU memory and image count. Thus, it was evident that the best image dimension for training the model when using Dataset 2 was (50,50,3) as well. Another challenge is the lack of interpretability of the models. Although these models may be highly accurate, it may be difficult to understand how they arrived at a particular diagnosis. As a result, determining why a model produced an incorrect prediction can be particularly difficult if the model produced a false positive or false negative result. Despite these challenges, using deep learning and machine learning models for COVID-19 detection using chest X-ray images holds promise for improving the accuracy and efficiency of COVID-19 diagnosis. Further research is needed to address the challenges associated with these models and to validate their accuracy and effectiveness in clinical settings. Additionally, efforts should be made to develop standardized protocols for image acquisition and annotation and improve the interpretability of the models. Overall, applying these models represents an exciting opportunity for advancing our ability to diagnose and manage COVID-19.

Chapter 7
Conclusion

The research work presented some research methods using traditional Machine Learning and Deep Learning algorithms, along with data analysis methods, to detect COVID-19 by examining chest X-ray images. The models used are able to detect COVID-19 with a high degree of accuracy, and we found that by using deep learning models, they could detect COVID-19 more accurately than using traditional data analysis methods.

7.1 Conclusions

Our master's thesis examined the use of Deep Learning models and Machine Learning methods to detect COVID-19 in chest X-ray images. Public health has been affected by COVID-19, and we need accurate and rapid diagnostic tools. COVID-19 can be diagnosed using chest X-ray imaging, a widely available and cost-effective diagnostic modality. Using chest X-ray images, this thesis aims to address the problem of the COVID-19 pandemic by detecting it. To classify chest X-ray images as positive or negative for COVID-19, we applied several deep learning models, Convolutional Neural Networks (CNNs), and transfer learning techniques. Furthermore, we compared the performance of our models with that of radiologists using a dataset of chest X-ray images from COVID-19 patients. According to our results, deep learning models could detect COVID-19 with high accuracy, with some models outperforming radiologists regarding sensitivity and specificity. The use of transfer learning (VGG19) approaches and pre-trained models to classify COVID-19 chest X-ray images showed promising results. However, our experiments also highlight the importance of dataset quality and size, as well as the interpretability and generalizability of models. According to our results, the performance of the models varied depending on the dataset used, and further research is needed to ensure the reliability and generalizability of the deep learning models in clinical practice. In addition, we realized that if we use different datasets for training and testing, some bias may occur, but this can be addressed by utilizing a larger dataset. However, data augmentation by Brightness, Contrast, and Flipping for small datasets did not work. It could not accurately predict the test data from different sources, and it misclassified normal images as COVID-19 images, or sometimes they were all classified as Viral Pneumonia images. In summary, our findings suggest that deep learning models may assist in diagnosing COVID-19 using chest X-ray images, but further research and validation will be necessary to ensure their clinical validity and applicability.

Bibliography

[1] Covid-19 Image Dataset — kaggle.com. https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset. [Accessed 13-Mar-2023].

[2] COVID-19 Radiography Database — kaggle.com. https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database. [Accessed 13-Mar-2023].

[3] — covid19.who.int. https://covid19.who.int/table. [Accessed 04-Nov-2022].

[4] Classification of COVID-19 from chest x-ray images using deep features and correlation coefficient — ncbi.nlm.nih.gov. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8958819/. [Accessed 12-Nov-2022].

[5] Geoff Currie, K Elizabeth Hawk, Eric Rohren, Alanna Vial, and Ran Klein. Machine learning and deep learning in medical imaging: intelligent imaging. *Journal of medical imaging and radiation sciences*, 50(4):477–487, 2019.

[6] Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.

[7] Yunqian Ma and Guodong Guo. *Support vector machines applications*, volume 649. Springer, 2014.

[8] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42:1–13, 2018.

[9] Monika Bansal, Munish Kumar, Monika Sachdeva, and Ajay Mittal. Transfer learning for image classification using vgg19: Caltech-101 image data set. *Journal of ambient intelligence and humanized computing*, pages 1–12, 2021.

[10] A Convolutional Autoencoder Topology for Classification in High-Dimensional Noisy Image Datasets — ncbi.nlm.nih.gov. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8622369/. [Accessed 02-May-2023].

[11] Malathy Jawahar, J Prassanna, Vinayakumar Ravi, L Jani Anbarasi, S Graceline Jasmine, R Manikandan, Ramesh Sekaran, and Suthendran Kannan. Computer-aided diagnosis of covid-19 from chest x-ray images using histogram-oriented gra-

dient features and random forest classifier. *Multimedia Tools and Applications*, 81(28):40451–40468, 2022.

[12] Histogram of Oriented Gradients explained using OpenCV — learnopencv.com. https://learnopencv.com/histogram-of-oriented-gradients/. [Accessed 04-May-2023].

[13] Co-occurrence matrix - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Co-occurrence_matrix. [Accessed 04-May-2023].

[14] k-nearest neighbors algorithm - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. [Accessed 04-May-2023].

[15] César Ortiz-Toro, Angel García-Pedrero, Mario Lillo-Saavedra, and Consuelo Gonzalo-Martín. Automatic detection of pneumonia in chest x-ray images using textural features. *Computers in Biology and Medicine*, 145:105466, 2022.

[16] sklearn.ensemble.RandomForestClassifier — scikit-learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed 04-May-2023].

[17] 1.4. Support Vector Machines — scikit-learn.org. https://scikit-learn.org/stable/modules/svm.html. [Accessed 14-May-2023].

[18] Abhishek Dixit, Ashish Mani, and Rohit Bansal. Cov2-detect-net: Design of covid-19 prediction model based on hybrid de-pso with svm using chest x-ray images. *Information sciences*, 571:676–692, 2021.

[19] sklearn.cluster.KMeans — scikit-learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. [Accessed 04-May-2023].

[20] Ali R Yildiz. A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations. *Applied Soft Computing*, 13(3):1561–1566, 2013.

[21] Particle swarm optimization - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Particle_swarm_optimization. [Accessed 04-May-2023].

[22] Jamal N Hasoon, Ali Hussein Fadel, Rasha Subhi Hameed, Salama A Mostafa, Bashar Ahmed Khalaf, Mazin Abed Mohammed, and Jan Nedoma. Covid-19

anomaly detection and classification method based on supervised machine learning of chest x-ray images. *Results in Physics*, 31:105045, 2021.

[23] Local Binary Patterns - Scholarpedia — scholarpedia.org. http://www.scholarpedia.org/article/Local_Binary_Patterns. [Accessed 04-May-2023].

[24] Mahotas - Haralick features - GeeksforGeeks — geeksforgeeks.org. https://www.geeksforgeeks.org/mahotas-haralick-features/. [Accessed 04-May-2023].

[25] AD Ningtyas, EB Nababan, and Syahril Efendi. Performance analysis of local binary pattern and k-nearest neighbor on image classification of fingers leaves. *International Journal of Nonlinear Analysis and Applications*, 13(1):1701–1708, 2022.

[26] Kelvin Lee, Che Yon Choo, Hui Qing See, Zhuan Jiang Tan, and Yunli Lee. Human detection using histogram of oriented gradients and human body ratio estimation. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 4, pages 18–22. IEEE, 2010.

[27] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

[28] Di Huang, Caifeng Shan, Mohsen Ardabilian, Yunhong Wang, and Liming Chen. Local binary patterns and its application to facial image analysis: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):765–781, 2011.

[29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[30] Shutao Li, James Kwok, Hailong Zhu, and Yaonan Wang. Texture classification using support vector machines. *Pattern Recognition*, 36:2883–2893, 12 2003.

[31] Mohamed Loey, Shaker El-Sappagh, and Seyedali Mirjalili. Bayesian-based optimized deep learning model to detect covid-19 patients using chest x-ray image data. *Computers in Biology and Medicine*, 142:105213, 2022.

[32] Bayesian optimization - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Bayesian_optimization. [Accessed 07-May-2023].

[33] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[34] Govardhan Jain, Deepti Mittal, Daksh Thakur, and Madhup K Mittal. A deep learning approach to detect covid-19 coronavirus with x-ray images. *Biocybernetics and biomedical engineering*, 40(4):1391–1405, 2020.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[36] Bhawna Nigam, Ayan Nigam, Rahul Jain, Shubham Dodia, Nidhi Arora, and B Annappa. Covid-19: Automatic detection from x-ray images by utilizing deep learning methods. *Expert Systems with Applications*, 176:114883, 2021.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks (2016). *arXiv preprint arXiv:1608.06993*, 7, 2016.

[39] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[40] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[41] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[42] Harsh Panwar, PK Gupta, Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, Prakhar Bhardwaj, and Vaishnavi Singh. A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images. *Chaos, Solitons & Fractals*, 140:110190, 2020.

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[44] Shervin Minaee, Rahele Kafieh, Milan Sonka, Shakib Yazdani, and Ghazaleh Jamalipour Soufi. Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning. *Medical image analysis*, 65:101794, 2020.

[45] Hai Ye, Qingyu Tan, Ruidan He, Juntao Li, Hwee Tou Ng, and Lidong Bing. Feature adaptation of pre-trained language models across languages and domains with robust self-training. *arXiv preprint arXiv:2009.11538*, 2020.

[46] Emtiaz Hussain, Mahmudul Hasan, Md Anisur Rahman, Ickjai Lee, Tasmi Tamanna, and Mohammad Zavid Parvez. Corodet: A deep learning based classification for covid-19 detection using chest x-ray images. *Chaos, Solitons & Fractals*, 142:110495, 2021.

[47] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[48] K Shankar, Eswaran Perumal, Prayag Tiwari, Mohammad Shorfuzzaman, and Deepak Gupta. Deep learning and evolutionary intelligence with fusion-based feature extraction for detection of covid-19 from chest x-ray images. *Multimedia Systems*, 28(4):1175–1187, 2022.

[49] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

[50] Mary M Galloway. Texture analysis using gray level run lengths. *Computer graphics and image processing*, 4(2):172–179, 1975.

[51] Seyedali Mirjalili, Amir H Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, 114:163–191, 2017.

[52] Stefanos Karakanis and Georgios Leontidis. Lightweight deep learning models for detecting covid-19 from chest x-ray images. *Computers in biology and medicine*, 130:104181, 2021.

[53] Khabir Uddin Ahamed, Manowarul Islam, Ashraf Uddin, Arnisha Akhter, Bikash Kumar Paul, Mohammad Abu Yousuf, Shahadat Uddin, Julian MW Quinn, and Mohammad Ali Moni. A deep learning approach using effective preprocessing techniques to detect covid-19 from chest ct-scan and x-ray images. *Computers in biology and medicine*, 139:105014, 2021.

[54] Sara Hosseinzadeh Kassania, Peyman Hosseinzadeh Kassanib, Michal J Wesolowskic, Kevin A Schneidera, and Ralph Detersa. Automatic detection of coronavirus

disease (covid-19) in x-ray and ct images: a machine learning based approach. *Biocybernetics and Biomedical Engineering*, 41(3):867–879, 2021.

[55] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[57] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[58] Chaimae Ouchicha, Ouafae Ammor, and Mohammed Meknassi. Cvdnet: A novel deep learning architecture for detection of coronavirus (covid-19) from chest x-ray images. *Chaos, Solitons & Fractals*, 140:110245, 2020.

[59] Eduardo Luz, Pedro Silva, Rodrigo Silva, Ludmila Silva, João Guimarães, Gustavo Miozzo, Gladston Moreira, and David Menotti. Towards an effective and efficient deep learning model for covid-19 patterns detection in x-ray images. *Research on Biomedical Engineering*, pages 1–14, 2021.

[60] COVID-Net/COVIDx.md at master · lindawangg/COVID-Net — github.com. https://github.com/lindawangg/COVID-Net/blob/master/docs/COVIDx.md. [Accessed 07-May-2023].

[61] Sarra Guefrechi, Marwa Ben Jabra, Adel Ammar, Anis Koubaa, and Habib Hamam. Deep learning based detection of covid-19 from chest x-ray images. *Multimedia tools and applications*, 80:31803–31820, 2021.

[62] Aras M Ismael and Abdulkadir Şengür. Deep learning approaches for covid-19 detection based on chest x-ray images. *Expert Systems with Applications*, 164:114054, 2021.

[63] Seyed Mohammad Jafar Jalali, Milad Ahmadian, Sajad Ahmadian, Rachid Hedjam, Abbas Khosravi, and Saeid Nahavandi. X-ray image based covid-19 detection using evolutionary deep learning approach. *Expert Systems with Applications*, 201:116942, 2022.

[64] Prottoy Saha, Muhammad Sheikh Sadi, and Md Milon Islam. Emcnet: Automated covid-19 diagnosis from x-ray images using convolutional neural network

and ensemble of machine learning classifiers. *Informatics in medicine unlocked*, 22:100505, 2021.

[65] 1.10. Decision Trees — scikit-learn.org. https://scikit-learn.org/stable/modules/tree.html#:~:text=Decision%20Trees%20(DTs)%20are%20a,inferred%20from%20the%20data%20features. [Accessed 14-May-2023].

[66] sklearn.ensemble.AdaBoostClassifier — scikit-learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html. [Accessed 14-May-2023].

[67] Murugan Hemalatha. A hybrid random forest deep learning classifier empowered edge cloud architecture for covid-19 and pneumonia detection. *Expert Systems with Applications*, 210:118227, 2022.

[68] Yu Ding, Wan Zhang, Xingqiang Zhao, Liwen Zhang, and Fei Yan. A hybrid random forest method fusing wavelet transform and variable importance for the quantitative analysis of k in potassic salt ore using laser-induced breakdown spectroscopy. *Journal of Analytical Atomic Spectrometry*, 35(6):1131–1138, 2020.

[69] YR Fan, Gordon H Huang, YP Li, XQ Wang, Zhong Li, and Lei Jin. Development of pca-based cluster quantile regression (pca-cqr) framework for streamflow prediction: Application to the xiangxi river watershed, china. *Applied Soft Computing*, 51:280–293, 2017.

[70] Mesut Toğaçar, Burhan Ergen, and Zafer Cömert. Covid-19 detection using deep learning models to exploit social mimic optimization and structured chest x-ray images using fuzzy color and stacking approaches. *Computers in biology and medicine*, 121:103805, 2020.

[71] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[72] Saddam Hussain Khan, Anabia Sohail, Asifullah Khan, Mehdi Hassan, Yeon Soo Lee, Jamshed Alam, Abdul Basit, and Saima Zubair. Covid-19 detection in chest x-ray images using deep boosted hybrid learning. *Computers in Biology and Medicine*, 137:104816, 2021.

[73] Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. A hybrid deep learning architecture for sentiment analysis. In *Proceedings of COLING 2016,*

the 26th International Conference on Computational Linguistics: Technical Papers, pages 482–493, 2016.

[74] Rui Yan, Fei Ren, Zihao Wang, Lihua Wang, Tong Zhang, Yudong Liu, Xiaosong Rao, Chunhou Zheng, and Fa Zhang. Breast cancer histopathological image classification using a hybrid deep neural network. *Methods*, 173:52–60, 2020.

[75] Jawad Rasheed, Alaa Ali Hameed, Chawki Djeddi, Akhtar Jamil, and Fadi Al-Turjman. A machine learning-based framework for diagnosis of covid-19 from chest x-ray images. *Interdisciplinary Sciences: Computational Life Sciences*, 13:103–117, 2021.

[76] Nahida Habib, Md Mahmodul Hasan, Md Mahfuz Reza, and Mohammad Motiur Rahman. Ensemble of chexnet and vgg-19 feature extractor with random forest classifier for pediatric pneumonia detection. *SN Computer Science*, 1:1–9, 2020.

[77] CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning — arxiv.org. https://arxiv.org/abs/1711.05225. [Accessed 07-May-2023].

[78] Danial Sharifrazi, Roohallah Alizadehsani, Mohamad Roshanzamir, Javad Hassannataj Joloudari, Afshin Shoeibi, Mahboobeh Jafari, Sadiq Hussain, Zahra Alizadeh Sani, Fereshteh Hasanzadeh, Fahime Khozeimeh, et al. Fusion of convolution neural network, support vector machine and sobel filter for accurate detection of covid-19 patients using x-ray images. *Biomedical Signal Processing and Control*, 68:102622, 2021.

[79] Random forest - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Random_forest#Properties. [Accessed 20-Feb-2023].

[80] Support vector machine - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Support_vector_machine. [Accessed 08-Mar-2023].

[81] Convolutional neural network - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Convolutional_neural_network. [Accessed 08-Mar-2023].

[82] Mike. Why is the VGG Network Commonly Used? — nnart.org. https://nnart.org/why-is-vgg-commonly-used/. [Accessed 14-Nov-2022].

[83] Dr. Info Sec. VGG-19 Convolutional Neural Network - All about Machine Learning — blog.techcraft.org. https://blog.techcraft.org/vgg-19-convolutional-neural-network/. [Accessed 29-Nov-2022].

[84] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. Deep clustering with convolutional autoencoders. In Derong Liu, Shengli Xie, Yuanqing Li, Dongbin Zhao, and El-Sayed M. El-Alfy, editors, *Neural Information Processing*, pages 373–382, Cham, 2017. Springer International Publishing.