A Comparative Study of Lung Cancer Prediction Using Deep Learning

by

Aka Mohammad Mugdho
16101249
Md. Jawad Hossain Bhuiyan
16301187
Tawsif Mustasin Rafin
18301102
Adib Muhammad Amit
21241062

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2022

Declaration

It is hereby declared that

1. The thesis that we submitted was our own original work done while pursuing a degree at BRAC University.
2. The thesis must not include any content that has been published before or that was authored by a third party, unless it is properly referenced with complete and correct referencing.
3. The thesis must not include any content that has been approved or submitted for consideration for any other degree or certificate at a university or other institution.
4. We have given credit to all major sources of help.

Student's Full Name & Signature:

[illegible]

Aka Mohammad Mugdho

16101249

Jawad H Bhuiyan

Md. Jawad Hossain Bhuiyan

16301187

Tawsif Rafin

Tawsif Mustasin Rafin

18301102

Amit

Adib Muhammad Amit

21241062

Approval

The thesis/project titled "A Comparative Study of Lung Cancer Prediction Using Deep Learning" submitted by

1. Aka Mohammad Mugdho (16101249)
2. Md. Jawad Hossain Bhuiyan (16301187)
3. Tawsif Mustasin Rafin (18301102)
4. Adib Muhammad Amit (21241062)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 20, 2022.

Examining Committee:

Supervisor:
(Member)

[illegible]

Dr. Amitabha Chakrabarty

Associate Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Annajiat Alim Rasel

Digitally signed by Annajiat Alim Rasel
DN: cn=Annajiat Alim Rasel, o=Brac University, ou=CSE Department, email=annajiat@bracu.ac.bd, c=BD
Date: 2022.09.18 08:05:23 +06'00'

Annajiat Alim Rasel

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

____

Sadia Hamid Kazi

Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

At the point when cells in the body develop out of control, this is alluded to as cancerous development. Lung cancer is the term used to depict cancer that starts in the lungs. At first in the field, classifier-based approaches are joined with various division calculations to utilize picture acknowledgment to recognize lung cancer nodules. This study found that CT scan images are more reasonable for delivering improved results than other imaging modalities. The use of the images is a piece of chiefly inspecting the CT scanned images that are viewed as informational collections for patients affected by lung cancer. The suggestion of our paper exclusively centers around the execution of concentrating on the calculation's accuracy in diagnosing lung cancer. Thus, the primary plan of our examination is to utilize examined calculations to conclude which strategy is the most efficient method for detecting lung cancer initially. After training the model we found that Over all accuracy of Resnet-18 is 99.54%, the Overall accuracy of Vgg-19 is 96.35%, The overall accuracy of MobileNet V2 is 98.17%, Dense Net161 is 99.09% and Inception V3 is 98.17%. So we can see that ResNet18 perform better than other train model

Keywords: Hog Feature Extraction, Lung Cancer, Deep learning, ResNet18, DeneNet161, MobileNetV2, ShuffleNet, InceptionV3, VGG19.

Dedication

We would like to dedicate this paper to our beloved parents who helped us get to this position.

Acknowledgement

First and foremost, thanks to the Great Almighty, we were able to complete our thesis without encountering any significant obstacles. Secondly, we appreciate the guidance and support provided by our co-supervisor, Mr. Annajiat Alim Rasel, and our supervisor, Dr. Amitabha Chakrabarty, during this project. Last but not least, a particular thank you to our parents, who's constant encouragement and support were essential to the accomplishment of our thesis.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Initial Thoughts

Lung Cancer is a particular kind of cancer that influences the entire respiratory framework and is one of the main sources of cancer-related deaths around the world. Any kind of cancer can be terminal on the off chance that not treated accurately and immediately. Lung Cancer starts in the lungs and spreads from that point to the lymph hubs and afterward all through the whole human body. Lung cancer can be isolated into three sorts and they are adenocarcinoma, squamous cell carcinoma, and enormous cell carcinoma. Each type requests an alternate treatment. On the off chance that we are rigorously discussing numbers as far as how serious lung cancer is then we need to think about these measurements: an expected 236740 individuals will be determined to have lung cancer in the U.S. in 2022. The death pace of lung cancer patients is multiple times more than that of prostate cancer as well as multiple times more than breast cancer. Lung Cancer kills men multiple times more than prostate cancer and it kills off ladies multiple times more than breast cancer [11]. According to the survey, more than 200000 people were infected with lung cancer worldwide in 2021, with the survival rate being 50% [29]. Because lung cancer is a growing public health concern, there have been an excessive number of ways developed in the past couple of years to diagnose lung cancer, the majority of which utilize CT scan images, and some others make use of X-ray images.
Roughly, north of a hundred thousand individuals lose their lives to lung cancer in the U.S. what's more, it is as yet expanding. The main driver of lung cancer is guessed to be smoking, recycled smoking, asbestos, radiation, air contamination, motor exhaust, different harmful metals like cadmium, and so on. The specialists pertinent to the clinical field conjoined with software engineering have prevailed with regard to figuring out how to treat this infirmity.

In this research, we have approached deep learning prediction-based models that can predict with accuracy whether a person is affected by lung cancer or not. Deep learning is basically a machine learning technique that is utilized to teach or make computers learn objectives that come naturally to human beings. The idea of Deep Learning was conceived by Geoffrey Hinton, a British-Canadian psychologist who introduced the idea of Artificial Neural Networks, and from then on the evolution of Deep Learning is staggering. The application of Deep Learning is immeasur-

able and scientists have barely scratched the surface of its potential. Some of the examples of the utilization or application of Deep Learning are fraud detection, computer vision, vocal AI, natural language processing, data refining, autonomous vehicles, supercomputers, healthcare, emotional intelligence, investment modeling, e-commerce and etc.
In this research, we utilized deep learning because common deep learning techniques have been used to make a prediction-based model do exactly what we are trying to do but deep learning models or techniques have been scarcely used to make a prediction-based model. It has been done before but our aim is to make multiple models using deep learning and compare and contrast between them which has the highest accuracy and which one takes the least amount of time to bear results. Furthermore, we decided to utilize deep learning because the architectures are easier to understand and there are fewer chances of overfitting and there is a high chance of getting a high accuracy due to its compound structure.

Our research has been organized to accommodate the concepts, the ideas behind the research, the algorithms, preprocessing, data acquisition, methodology and etc. It is done so as to reflect the main goal which is to compare and contrast the multiple models and show the multiple models we implemented, the accuracy, F1 score, and other desired results, and ultimately determine the best model to predict lung cancer in patients.

1.2 Aims and Objectives

Our research aim and objectives are as follows:

1. To compare and contrast trained models and determine which is the best model to predict the incidence of lung cancer.
2. Delve deeper into the possibilities of deep learning in predicting lung cancer
3. Produce more accurate predictions with the help of the best-trained models'
4. To provide a review of the best model along with its accuracy.
5. Utilize and understand Deep Learning Models.
6. Display the potential of Deep Learning in the field of prediction-based algorithms.
7. Portray the comparisons between the models that we used in our research.
8. Determine the best algorithm used amongst the paper sourced and utilized.
9. Determine the best architecture.
10. Display the statistics that entail the comparison and contrast amongst the papers as well as the algorithms chosen to be implemented.

1.3 Research Problems

The difficulty that we will need to deal with during our research is the absence of solid data sets. Numerous data sets have stretches or huge numbers of data in the middle between records. So despite the fact that there are a large number of data sets, few out of every odd one of them will be as dependable and precise for our utilization. Not exclusively were the picture quality poor yet in addition exceptionally boisterous. So we needed to cure that. Another issue we confronted in regards to the research is that the data set that we used/carried out appeared to override the naming of the records. To address this issue, we needed to change over the records to a number of documents. If not, the code might have been executed in a lot more straightforward manner and would have run a lot smoother. One of the key issues was picking the right engineer. As we probably are aware deep learning in expectation-based models is fairly new and we were battling a piece to pick the right engineering that accommodates our schematics. Furthermore, our data set was not mixing with a portion of the structures thus when we were picking the models that we wanted, we must be mindful. Besides, it was a piece challenging to source pertinent papers that relate to our subject and particularly deep learning papers were exhausting to gather. Despite the fact that there were many papers that elaborate on AI, there were very few papers that elaborate on deep learning. We additionally needed to actually look at the legitimacy of the papers because of the commonness of ruthless papers in the ebb and flow research field. Besides, since our paper is a review paper, we needed to look for an ever-increasing number of papers that we are hoping to investigate and afterward execute every one of the calculations referenced inside those research. Among those calculations, we needed to figure out which one was the most effective and the most dependable which was not a simple errand using any and all means. We additionally were deficient with regards to the ability to run the codes vital since only one of our individuals had the GPU sufficiently strong to run the codes. Accordingly, this research will actually want to figure out which model is the most productive and exact in anticipating the occurrence of cellular breakdown in the lungs inside patients.

Chapter 2

Literature Review and Related Works

We know for a fact that lung cancer is a rising problem and that more and more individuals are becoming impacted by it every day, which has resulted in a large number of study papers being published on the subject. Some identify cancer in its early stages, while others save lives. And we have selected a few of these study articles to determine which strategy is the most effective for identifying lung cancer in the first place.

In the reports that we researched [11] and [4], we found that these papers aim to shed light on how to detect/predict the occurrence of lung cancer in patients with higher accuracy. The main aim was to examine the use of classification algorithms such as Support Vector Machine (SVM), Naïve Bayes, Decision tree, and logistic regression. For logistic regression, the accuracy was 66.7%, for Naïve Bayes, it was 87.87%, for the decision tree, it was 90%, and for Support Vector Machine(SVM), it was 99.2%. The data set also reflects the amount of data scoured from the data repositories. This study employed a proposed model that analyzed 1359 out of 1449 CT scans and had an accuracy of 82%. In [11] it highlighted machine learning algorithms for the sole purpose of predicting lung cancer and doubles down by using the algorithm kernels, i.e., Gaussian, Radial Base Function (RBF), and Polynomial. The outcome divulged that the proposed multi-modal features would indeed assist in detecting and distinguishing lung cancers. In [24], it is observed that the researchers delved into the possibilities of supervised learning algorithms. They aimed to utilize machine learning algorithms such as Long Short-term Memory(LSTM), Mean Squared Error(MSE), and BP on their data set to predict the incidence of lung cancer. In contrast, the paper [26] carries out the Support Vector Machine (SVM), K Nearest Neighbor (KNN), and Convolutional Neural Network (CNN) which are the key topics covered in this study (CNN). Here, the Support Vector Machine(SVM) received the highest percentage with 95.56%, followed by CNN with 92.11% and K Nearest Neighbour(KNN)with 88.40% [14].

However, compared to reports [11], it is a bit less. Though the Unique Client Identifier(UCI) database is used in the researched papers [26] and [5], this paper used CT scanned pictures as its data set as it contains less noise than X-rays and MRI images. This paper implements grayscale conversion, noise reduction, and binarization techniques to generate an image in a suitable format, unlike articles [1] focusing on Histogram Equalization. The Histogram Equalization is used for image preprocessing and feature extraction, as well as a neural network classifier to determine a patient's state. Based on the outcomes of this study, a computer-aided diagnostic (CAD) system for early identification of lung cancer will be developed, boosting patient survival prospects. This evaluation is based on testing different classifiers on a data set. As a result, the primary purpose is to show that the kennel technique delivers more precise results. Nonetheless, the [33] paper authors wanted to efficiently manage large amounts of data and high-dimensional data by using Apache Spark as its architecture. Support Vector Machine outperforms other algorithms in terms of accuracy (86%), which is comparatively less than both results in paper [11] and [26] and the papers that we have gone through, and also to optimize the architecture and process data T-BSVM and WTA-SVM are proposed.

The paper [13] mostly comprises ANN to diagnose lung cancer, and this paper [9] uses the method of computer-aided lung image categorization. In the article [13], the aspects of image acquisition and processing include preprocessing and segmentation. Anxiety, Chronic Disease, Fatigue, Allergy, Wheezing, Coughing, Shortness of Breath, Swallowing Difficulty, and Chest Pain was utilized for training the ANN. Thus, when 80% of the sample data is used for training and 20% for validation, the ANN model can detect the absence of lung cancer with 96.67% accuracy. Whereas in the researched papers, uncovering and categorizing lung cancer using MRI images using the k-nearest neighbor (KNN) algorithm where gives 86.25% accuracy overall, which is less than the 88.40% found in the researched papers. Moreover, imagery of the lung was binarized to detect typical abnormalities. The tumor size correctly identified the stage. The method assesses roughly 80 CT scans to detect lung cancer, 40 of which are malignant accurately. In addition, the proposed approach can detect cancer nodules near the lung's edges along with discussing Deep convolutional neural networks(CNN) in image processing. Again in the researched papers, The researchers are unique from other documents using the NIHI NCI(National Institute of Cancer) respiratory organ image information syndicate Learning and Instructional Development Centre(LIDC) data. There were 888 CT scans in this data set, each with coordinates and ground truth labels. The photos of lung cancer are used as observational symbols, believing the various tumors represented different stages of their bodies. The model is accurate 89.3% of the time and recalls 72.2%. This method uses a cluster of CNNs to improve categorization. We also found that the WS-GDL technique can identify important and critical aspects while also adaptive analyzing the sickness from paper [31]. The WS-GDL strategy for a cellular breakdown in lung infection is proof of the lung infection.

We proposed robust RICA, and sparse channel-based autoencoders and compared the results to the conventional surface, morphological, and entropy-based elements. For a cellular breakdown in the lungs recognition, SVM, RBF with TA (97.17%) outperformed SVM Gaussian with TA (96.67%) and DT with TA (92.88%). The area under the collector working bend was logged. AUC is valued between 0 and 1, as shown in [24,25,35,124]. Lung cellular breakdown is the leading cause of death globally.

From the previously stated statistics, we can infer that in any lung cancer or any other type cancer prediction based work/project, Deep Learning algorithms have a much better chance of getting a better accuracy and overall performance measures than Machine Learning algorithms. An overview of the algorithms used in related works have been tabulated below:

Reference No.	Algorithm/ Model	Data set name	Data set size	Accuracy
[11]	SVM Polynomial	Lung Cancer Alliance	954 images	99.58%
[15]	SVM	Lung Cancer Data Set	1000 CT scans	99.2%
[21]	grt123	TCIA	1449	82%
[24]	SVR	Cancer incidence in five continents	10 countries, 42 years	83%
[25]	SVM	Lung Cancer Data set	32 observations	95.56%
[5]	SVM	ELCAP Public Lung Image DB	200 lung images	80%
[19]	ResNet50	HAM10000	10015 images	90%
[9]	SVM	AIMS Kochi	200 DICOM lung CT images	86.25%
[12]	CNN	LIDC	1623 images	93%
[8]	CNN	ImageNet	1400 images	81.97%
[31]	WS-GDL	Thoracic Surgery Data Set	1000 images	86%
[17]	DCNN	HAM10000	10015 images	99%
[22]	CNN	ISIC-Kaggle	25,780 images	79.45%
[27]	DCNN	HAM10000	10015 images	93.16%
[29]	ANN	HAM10000	10015 images	97.51%

Table 2.1: Summary of the classifiers.

Chapter 3

Methodology

We will describe the methods for this thesis work in this chapter. The approach began with selecting an appropriate dataset and its preparation using pre-processing methods. The workflow consists of proposing a model and evaluating its performance against six transfer learning models that have already been trained, including Inception v3, ShuffleNet, Vgg19, MobileNet_v2, Densenet161, and ResNet 18. The approach is completed by evaluating how well the models performed on the dataset. The following stages are listed in order according to the methodology: Conventional Expansion with CNN Classifier. Collecting the dataset, pre-processing the dataset, pre-training CNN models, proposing an appropriate CNN model, and assessing the performance of the CNN model are the first four steps.

3.1 Work Flow of the methodology

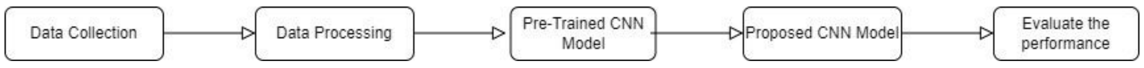


Figure 3.1: Workflow

3.2 Dataset observation and preprocessing

Collection and description of data sets

For the sake of our research, we have utilized the lung cancer data-set from The Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases which was collected in the above-mentioned specialist hospitals over a period of three months in fall 2019. The idea was to take the images, 1100 in total provided in the dataset, as reference for the model we will train to recognize and distinguish between malignant, benign, and normal and healthy lung CT scan images.

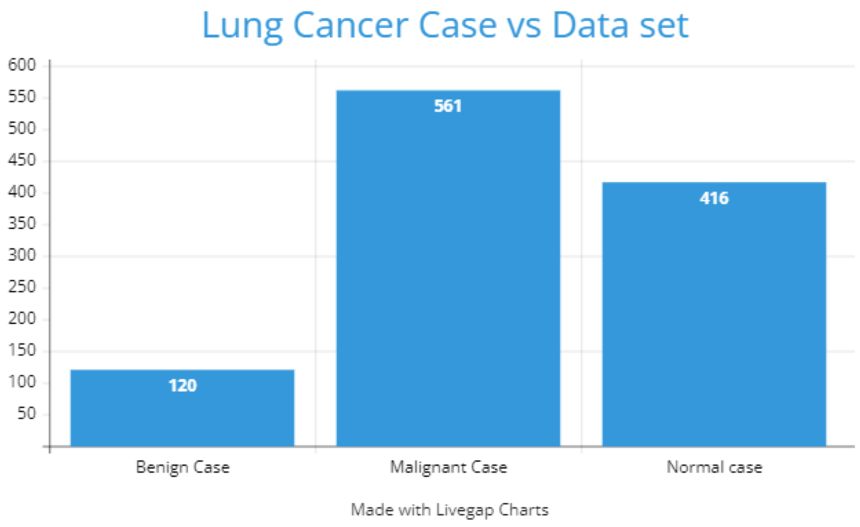


Figure 3.2: Data distribution table

Type of lung cancer

- **Benign Case:** Cancer that remains in its primary region without spreading to other parts of the body is considered benign. They don't spread to distant parts of the body or to nearby buildings. In general, benign instances of benign cancer will grow gradually and have certain limits.

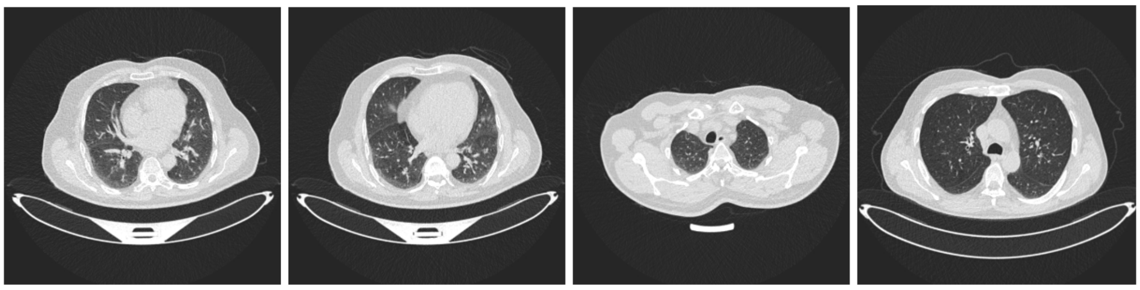


Figure 3.3: Benign cases [18]

- **Malignant case:** Cells in malignant malignancies grow and spread widely, both locally and to distant locations. Cancer is malignant cancer (i.e. they attack different locales). Through the lymphatic or vascular systems, they spread to too distant locations. We refer to this spreading as metastasis. Metastasis may occur everywhere in the body, however, it is often seen in the liver, lungs, brain, and bone.

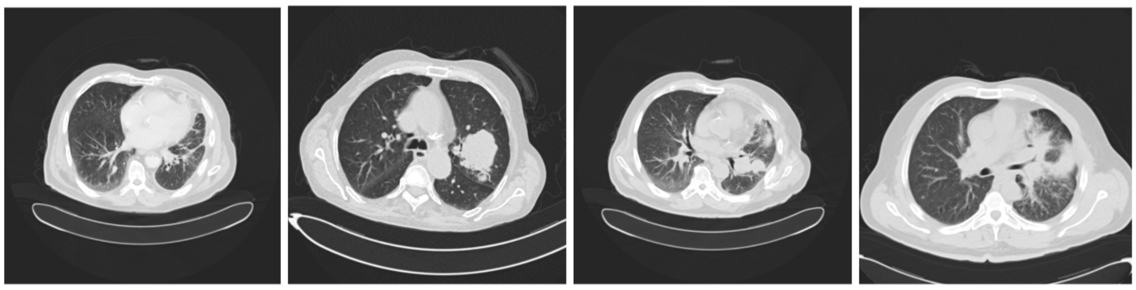


Figure 3.4: Malignant cases[18]

- **Normal case:** The lung that is healthy in shape and size. That is not infected with any kind of cancer. This data is given for reference.

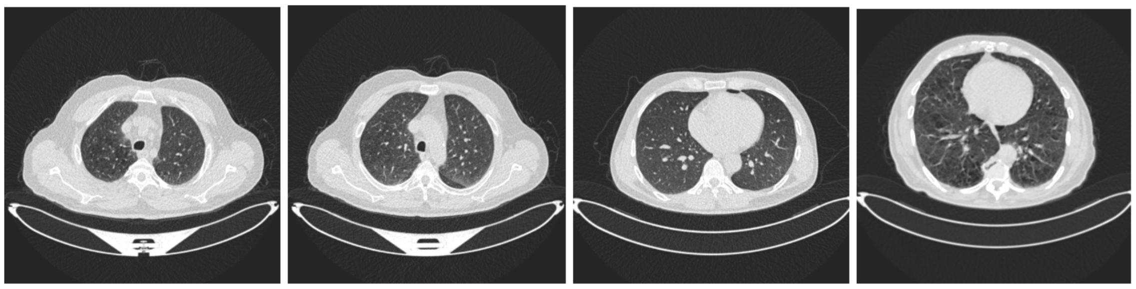


Figure 3.5: Normal cases[18]

3.3 Data Prepossessing

Any CNN model must go through data prepossessing in order to protect the integrity of the data. Data preparation in deep learning refers to the procedure of preparing raw data for the development and training of deep learning models by cleaning and organizing it. Data prepossessing, to put it simply, is a data mining technique used in deep learning that converts raw data into a format that is readable and understandable. Data prepossessing is needed in order to extract useful and distinguishing characteristics from the photos. The first step in developing any intelligent system is data preparation. The real-world data is often unreliable, inconsistent, and noisy. Additionally, a CNN model needs all input photos to be the same size, yet a data collection may include images of various sizes from many sources. Data prepossessing will solve these issues and make the data ready for a model. The data prepossessing steps are:

1. Acquire the data set
2. Identify and handle missing values
3. Split the data set into training and test set
4. Feature scaling.

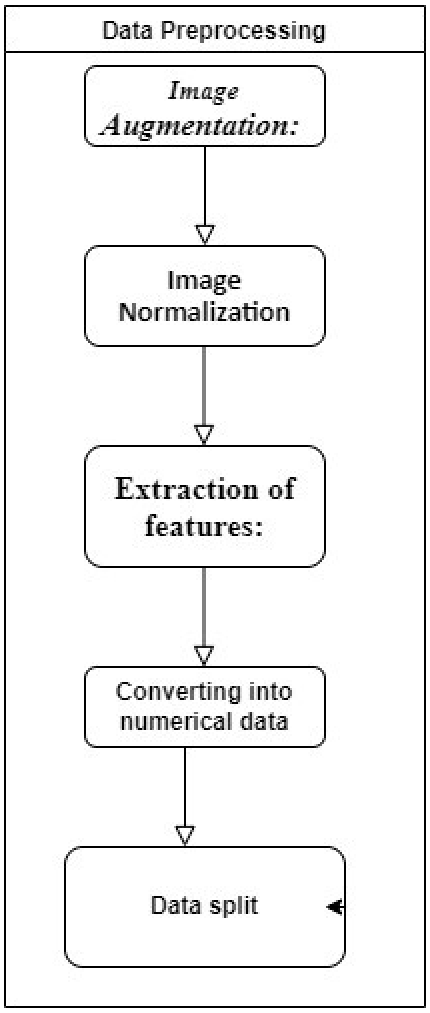


Figure 3.6: steps of data preprocessing

In the case of coding for implementing this work, we have applied python programming language and utilized the available open source libraries to develop classifier models. The various stages of implementation are the two points given below:

- **Importing and Splitting Data:** The images stored on memory are imported and read using the "OS". A total of 1100 images from the three classes of the data set are read and sorted into a list. The images are split into training, validation, and testing data. The train-validation-test ratio of the data is 70-10-20 percent, respectively, resulting in 788 images in the training set, 88 images for the validation set, and 221 images in the test set. Train, validation, and test sets are further split into batches of 32 images to feed the CNN.

- **Generating Labels:** In order to enable the classifier model to distinguish among malignant, benign, and normal classes, the semantic labels are converted to numeric labels, which is referred to as 'one hot encoding'. One hot encoding is a method used to label data set that have multiple categories, where a label is transformed into a binary matrix where 0 means that the particular data is not in a category and 1 means that it is.

3.3.1 Image Augmentation

Image augmentation has been performed as a data prepossessing step in this project. Image augmentation is a process of modifying existing images in order to generate additional data for the model training process. In other words, it is the technique of artificially increasing the data set available for deep learning model training. A deep CNN network usually requires many input samples to learn robust classification. Practically, the performance of a deep learning model increases with the amount of data supplied to the model. But data acquisition is a costly process since each data need to be manually verified and annotated. Because any error inthe data annotation or label will result in erroneous classification performance. In image augmentation, each image in the training set is slightly changed. All images in the data set are resized to the dimension of $299X299$ on both the training and test set. Then the augmentation applied in our model are

- Random rotation,
- Horizontal Flip,
- Cropping,
- Normalization

By applying these transformations to each image, we created a large number of training samples from the original data set containing only 1100 images. The variation introduced due to image augmentation enhances the performance of the model and makes our model suitable for real-world deployment. Because real-world images would contain various noise.

3.3.2 Image Normalization

Image normalization, a method for database architecture, decreases data duplication, ensures data uprightness, and gets rid of undesirable elements including insertion, update, and deletion anomalies. Z-score normalization, min-max normalization, and decimal scaling normalization are just a few of the numerous normalizing techniques that are accessible. By multiplying the image pixel value by 255, the data-set was normalized. The grayscale value of a picture is 255. After normalizing, all of the pixel values were between (0,1). Normally, scaled data is needed for neural networks, however, the photos here have RGB values from 0 to 255. So, using mean values of [0.485, 0.456, 0.406] for each of the three channels and standard deviation values of [0.229, 0.224, 0.225], the pictures are standardized from 0-255 to 0-1 as a result. The classifier network was then fed with each picture as a tensor. [34]

3.3.3 Extraction of features

Finding features is important. The inputs that result in the output are called features. The outcome of categorization is referred to as a label. It simplifies the data-set so that it may be processed and analyzed further. We also retrieved the features in this study, which made the data-set easier to handle. The data-set includes a lot of characteristics at the start. The model may run slowly as a result of the many features. We lowered the number of features by using feature extraction. Based on how well the attributes correlated with the actual value, we chose them. Only traits with a strong link to reality were retained. Our model became considerably quicker and more effective as a result. The image's gradient is computed. Combining the image's magnitude and angle yields the gradient. Gx and Gy are initially computed for each pixel in a block of $3x3$ pixels. The following formulas are used to compute Gx and Gy first for each pixel value.

$$G_x(r,c) = I(r,c+1) - I(r,c-1) \tag{3.1}$$

$$G_y(r,c) = I(r-1,c) - I(r+1,c) \tag{3.2}$$

where rows and columns are denoted by r and c, respectively.
Each pixel's magnitude and angle are computed using the following formulas once Gx has been determined:

$$Magnitude = \sqrt{G_x^2 + G_y^2} \tag{3.3}$$

$$Angle = |tan^{-1}(G_x/G_y)| \tag{3.4}$$

where rows and columns are denoted by r and c, respectively.

The following formulas are used to determine the magnitude and angle of each pixel once Gx are determined.

3.3.4 Converting into numerical data

The cancer column ground truth is not encoded. We have encoded cancer for the benefit of our model, and the encoded labels were put into a brand-new column called "label." Our labels have been changed into a numerical representation using the LabelEncoder function. The LabelEncoder function is included in the Python standard library. ML works effectively when a dataset has many scales for each feature.

3.3.5 Data Split

For train test validation, we have split the data set. The data was imbalanced that's why we did the augmentation. It was split into 70 percent,20 percent, and 10 percent. Validation data is exclusively used for hyperparameter adjustment. The model will be able to train on a significant amount of pictures since the training dataset is the largest, which will make the model stronger.
Again, we trained our dataset first using the training dataset. After completing the training phase, we adjusted each algorithm's hyperparameters on the validation dataset. Finally, we applied the model to the testing dataset to evaluate the performance of the models.

Chapter 4

Pre-trained Transfer Learning Model

4.1 Pretrained Transfer Learning Model

We utilized different pre-prepared CNN models in our research utilizing transfer learning. They are Resnet18, VGG19, InceptionV3 and Densenet13, Shufflenet, and Mobilenet_v2. The benefit of utilizing transfer learning is that it is pre-trained on the Imagenet dataset of 1100 images. Running in the deep learning model is insufficient. By utilizing pre-prepared models, we can increase the viability of a model. These models are various designs of CNN and are prepared on Imagenet. We will describe the models exhaustively in the paragraph below:

4.1.1 ResNet 18

ResNet-18 is the name of a convolutions neural network with 18 layers. A pre-trained version of the network that has been trained on more than a million photographs is included in the ImageNet database. The pre-trained network can classify images using 1000 distinct item categories. The ResNet-18 model developed by He et al. [35] is based on a residual learning framework that increases the efficiency of deep network training.[20], ResNet models' residual blocks, in contrast to monotonically progressive convolutions' initial unreferenced mapping, make it simpler to optimize the whole network, improving model accuracy. These residuals or "skip connections" are used for identity mapping, which neither adds extra factors nor complicates the process. The architecture of the ResNet-18 model is shown.[3]

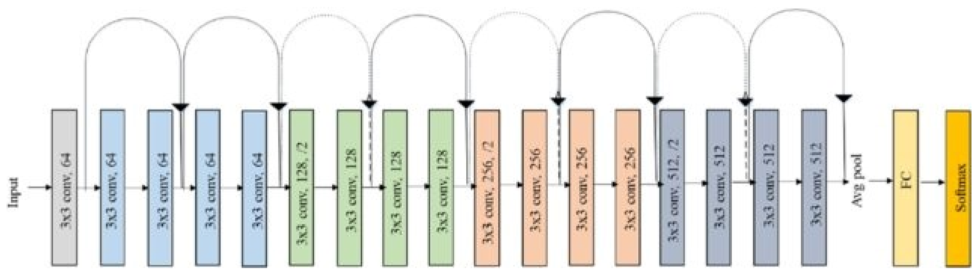


Figure 4.1: ResNet18 Architecture Mode [20]

The modular element of the generalized residual network architecture is a generalized residual block, which has parallel states for a residual stream, r, that has identity shortcut connections and is structurally similar to a residual block from the original ResNet with a single convolutional layer (parameters Wl,r), and a transient stream, t, that is a typical convolutional layer. (Wl,t→t). Additionally, each block transmits two additional sets of convolutional filters (Wl,rt, Wl,tr).

$$rl + 1 = (conv(rl, Wl, r{\to}r) + conv(tl, Wl, t{\to}r) + shortcut(rl)) \tag{4.1}$$

$$tl + 1 = (conv(rl, Wl, r{\to}r) + conv(tl, Wl, t{\to}r) \tag{4.2}$$

Same-stream and cross-stream activations are added along with the shortcut connection for the residual stream in order to generate the block's output states before batch normalization and ReLU nonlinearities are used (jointly) (Equation 1). (2015) Ioffe and Szegedy. Information from previous stages may be discarded thanks to the transient stream t's ability to handle data from either stream nonlinearly without shortcut links. With shortcut connections between each processing unit, the residual stream r operates similarly to the ResNet's original structure (He et al., 2015b). ResNet effectively combats the problem of extensive network degeneration by connecting the initial input to all layer outputs right away. With the addition of a residual connection to the inception module, Inception-ResNet enhances the network's ability to articulate patterns while also considerably accelerating training. Resnet identity mapping deepens the network, reroutes information flow across it, and rebuilds the learning process. In addition to accelerating network convergence and effectively addressing issues like gradient vanishing and network degradation, this improves the model's ability to represent data. The residual neural network consists of several overlapping residual block structures, whilst surrounding convolutional layers are connected by shortcuts to form residual blocks. The system of the leftover block is shown.

H is used to indicate input. The I output is represented by Hi+1, the weights by Wi, and the residual mapping by F. The remaining block mapping is so shown as follows: When the input dimension Hi and the output dimension Hi+1 don't agree, the linear projection is employed to align the dimensions.

The residual mapping is easier to learn experimentally via practice than the original mapping. As a consequence, the ResNet uses the middle stacked layers to learn the residual mapping. The residual mapping F is more sensitive to output variation and the parameter adjustment range is comparably wider, which speeds up learning and improves network optimization performance.

4.1.2 Inception v3

CNN network development underwent a critical turning point with the introduction of the Inception network. The bulk of well-known CNN models that existed before the model's development piled convolutional layers deeper and deeper to provide better performance. On the other hand, the Inception network proved difficult (laboriously engineered). To improve performance in terms of accuracy and speed, it used a number of tactics. Its continuing development showed the effects of several network iterations.

Inception v3 refers to the third version of Google's Inception CNN model. In comparison to older versions, it is more accurate and less costly to calculate. It is a 48-layer pre-trained convolutional neural network model, which has a lower error rate than prior models. On the ImageNet dataset, a trained version of the network exhibits 78.1% accuracy. The more than a million pictures in the ImageNet dataset are split into training datasets (1,281,167 images) and evaluation datasets ($50,000$ images). The 1000 item categories that this inception model can identify photos into include a mouse, keyboard, pen, flower, and various animals. The network has preserved in-depth feature representations for a range of pictures as a consequence. The key modifications applied to the Inception V3 model include the inclusion of auxiliary classifiers, factorization into smaller convolutions, spatial factorization into asymmetric convolutions, and effective grid size reduction. The model gathers general characteristics from the input photos in the first segment, and in the second, it classifies the images based on those attributes.[2] The input and output sizes of this model are $299 * 299 * 3$ and $8 * 8 * 2048$, respectively.

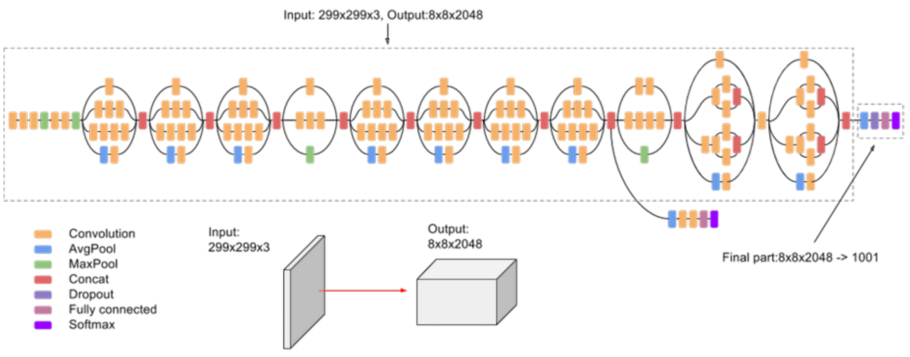


Figure 4.2: Inception_V3 Architecture [34]

4.1.3 VGG19

VGG19 is a complete CNN with layers that have already been trained and a strong understanding of visual form, color, and structure. A deep neural network called VGG19 was trained using a variety of classification tasks on millions of photos. A 19-layer deep convolutional neural network makes up the VGG-19. It is a modification of the VGG model and consists of 16 convolutional layers, 5 max pool layers, 3 fully connected layers, and 1 softmax layer. The architect of CNN controls the size and quantity of convolutional and fully linked layers. For instance, the VGG19 architecture received a fixed size $(224 * 224)$ RGB picture as input, indicating that the matrix was of form $(224 * 224 * 3)$. As a pre-processing step, just the average RGB value of each pixel throughout the whole training set was deleted. To maintain the spatial resolution of the picture, spatial padding was used. Stride 2 was used to perform max pooling across a 2 by 2 pixel frame. Rectified linear unit (ReLu) was utilized to describe non-linearity to increase model classification and computing performance since older models employed tanh or sigmoid functions; this proved to be noticeably superior to them. Once again, this model was composed of three fully connected layers, the first two of which were 4096 pixels in size. The third layer, which included a softmax function, had 1000 channels and was used for the 1000-way ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) classification. Furthermore, it is simple to do so since VGG19's autonomous feature extraction makes it possible to find the characteristics that set each cancer type apart without having to spend time manually evaluating them. The VGG19 is a fantastic model, but it does have some flaws, such the fact that there were a lot of parameters that needed to be trained and that the network was fairly huge.

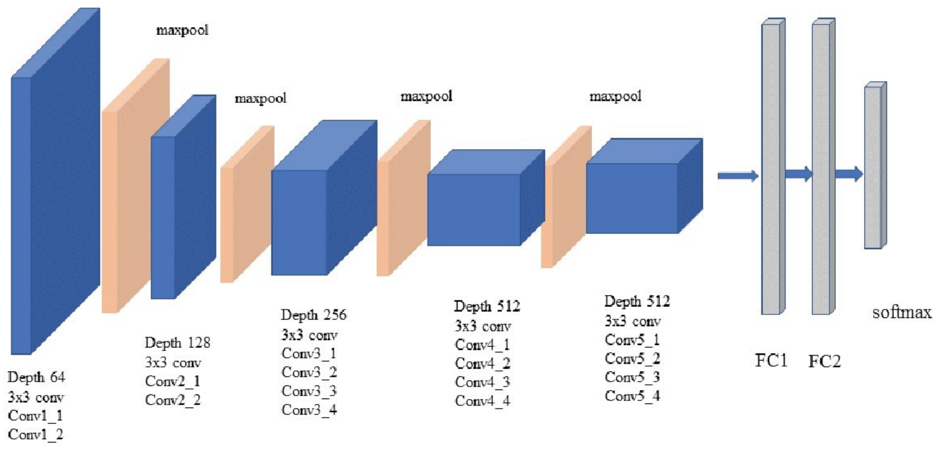


Figure 4.3: VGG19 Architecture [34]

4.1.4 MobileNetV2

This work makes use of both the CNN layers and the MobileNet V2 model to predict and classify the thoracic chest illnesses in the chest. MobileNet V2 is a similar notion that works well for simple mobile and on-board vision applications. Applications for deep learning include computer vision, robots, the Internet of Things (IoT), natural language processing (NLP), and the analysis of medical images.

The modified MobileNetV2 architecture consists of a set of hidden layers built on a bottleneck block that is still present. Each of these hidden layers has a depth-wise separable convolution, which significantly lowers the number of parameters and produces a lightweight neural network that is distinct from conventional convolution. A depth-wise convolution with a single filter is used in lieu of the usual convolution, and a pointwise convolution known as a depth-wise severable convolution is then applied. Three convolutional layers made up the majority of the bottleneck residual block. Figure 6 shows the last two layers, which were once a part of MobileNet's first generation and consist of a depth-wise convolution layer that filters the inputs and a 11 pointwise convolution layer. However, the objective of this one-layer system has changed..

MobileNet, one of the most popular deep learning structures for convenient gadgets, isn't just conservative yet additionally has high handling effectiveness. The key idea fundamental MobileNet is that the interaction is isolated into profundity-wise distinguishable 33 convolution channels, trailed by 11 convolution, as opposed to using conventional 33 convolution channels. With fewer tasks and boundaries, the new engineering achieves the equivalent separating and consolidating methodology as a conventional convolution. In MobileNetV1, the point-wise convolution had to keep a similar number of channels or increment it. MobileNetV2's point-wise convolution, then again, diminishes the number of channels. As demonstrated in Table 3, this layer, which is presently known as the projection layer, turns information with a ton of aspects (channels) into a tensor.

The augmentation layer is MobileNetV2's most memorable novel component. A 11 convolution is utilized for the development layer. Preceding playing out the profundity wise convolution, it has the obligation of expanding the quantity of diverts in the image information. Not at all like the projection layer, this development layer generally has more result channels than input channels.

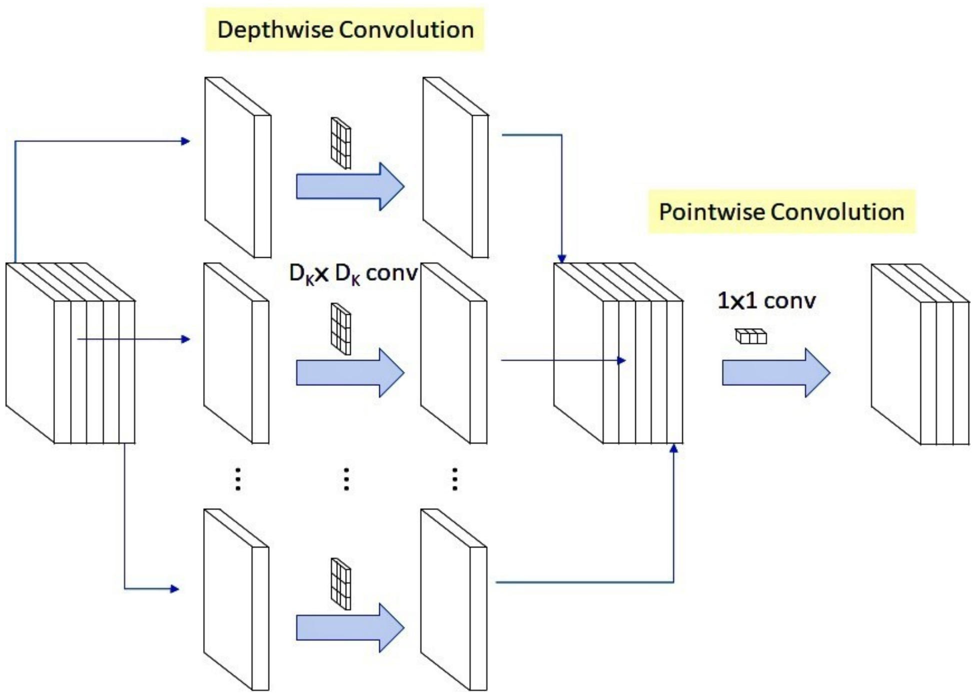


Figure 4.4: MobileNetV2 Architecture [32]

According to what was laid out in [24] the channel-wise DK×DK spatial convolution is what we refer to as the depth wise convolution. Let us consider, we have five channels, as shown in the figure above and we will have five DK×DK spatial convolutions. Meanwhile, the 1×1 convolution is known as the pointwise convolution which alters the dimension.

4.1.5 ShuffleNet

It was developed by Megvii Inc. (Face++), which encourages the CNN to attain great computational efficiency in addition to high accuracy. To achieve computational efficiency, it combines two elements: channel shuffling and pointwise group convolution. The feature map from the prior group is jumbled in the shuffle channel before being sent to the subsequent group's convolution layer. They also provide the ShuffleNet Unit, which implements pointwise group convolution before to the shuffle process[6]. Pointwise group convolution is applied twice, and 3x3 average pooling is added to enhance the channel's dimension while keeping the calculation cost reasonable.

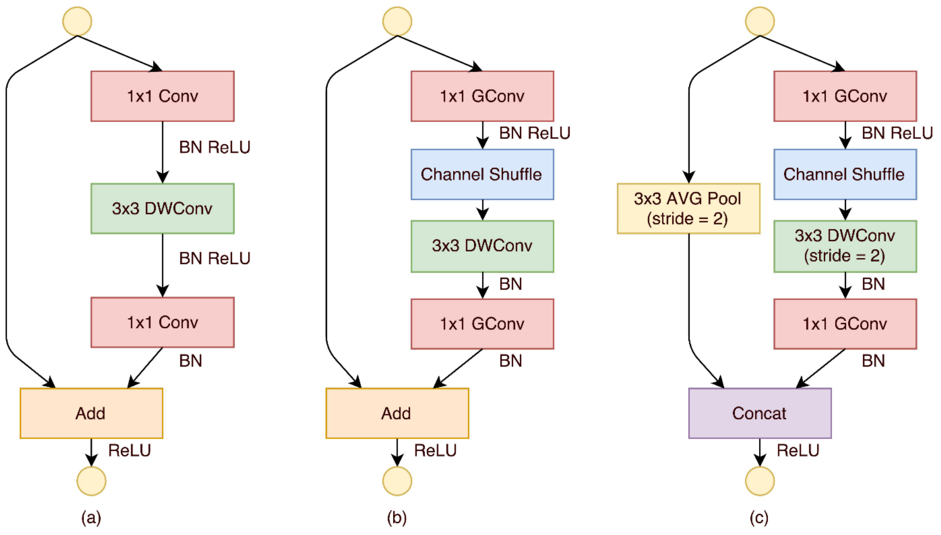


Figure 4.5: Shufflenet Architecture [6]

4.1.6 Densenet161

Consider a convolutional network being applied to a single picture, x0. The network consists of L layers, where'indexes the layer and each layer performs a non-linear transformation H'(). Batch Normalization (BN) rectified linear units (ReLU) pooling [19], or convolution are examples of procedures that may be combined to form H'() (Conv). The result of the "th layer" is designated as creating a network that uses fewer parameters to attain great accuracy. With this design, a new technique is introduced in which each layer gets a feature map input from all of the preceding levels.

This makes it possible for the subsequent layer to have information from each layer that came before it. Limiting the number of channels in DenseNet results in a smaller, more streamlined network. Apart from that, it systematically uses memory and processing expense."x." Huang et al.'s DenseNet designs provide a rich feature representation while being computationally effective. The fundamental explanation is because, as shown in Fig 4.6, the feature maps in each layer of the DenseNet model are concatenated with those from all the previous levels. The convolutional layers may accommodate fewer channels, which reduces the number of trainable parameters and makes the model more computationally efficient.[16] Additionally, the feature representation is improved by concatenating the feature maps from the preceding layers with the current layer.

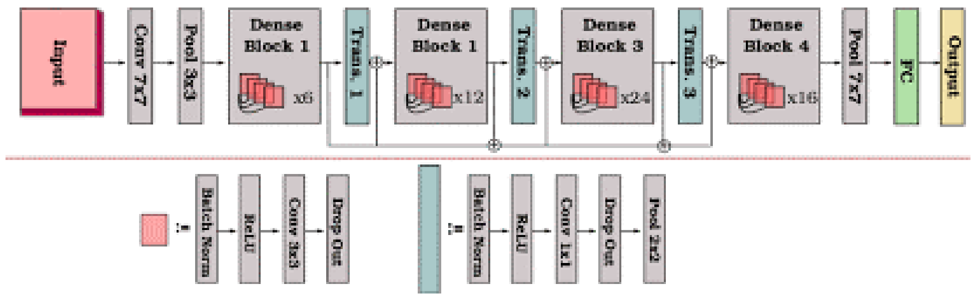


Figure 4.6: DenseNet161 Architecture [23]

ShuffleNet: An Extremely Efficient CNN for Mobile Devices

Chapter 5

Proposed Model

5.1 Proposed Model

The proposed technique for classifying lung cancer intends to precisely and quickly categorize lung cancer so that medical practitioners may begin treatment as soon as possible. The model must be developed in a manner that enables image data entry, systematic analysis of the input data, and prediction output. Here it depicts the model design from a high-level viewpoint. In this chapter, we developed a training scheme based on the efficient data augmentation technique to improve a CNN classifier's performance by producing new training samples from the old ones while maintaining the original class labels. Although data augmentation has often been employed as a regularizer, the main challenge in this situation is to properly expand the dataset size in order to overcome overfitting and generalization difficulties. Conventional augmentation is utilized as a benchmark way to enhance CNN's performance and compare it to other strategies. By applying a simple geometric correction to the original data, it may be used with small sample size. This transformation encompasses shearing, flipping, scaling, rotations, and translations. In addition, we suggested the CNN model architecture for the categorization of melanomas. By giving a slightly modified version of the input data, we hoped to improve the CNN model's generalizability so that our model could learn robust features. The two components of the suggested method are augmentation and categorization. Using a number of conventional augmentation techniques, we were able to double the training data for a specific dataset by four. The proposed CNN model was then trained using the expanded dataset. When compared to the CNN model trained without augmentation, the experimental findings demonstrate that the suggested approach can not only address overfitting and generalization problems but also enhance CNN accuracy. The strategy we suggested is summarized in the chart below [30].

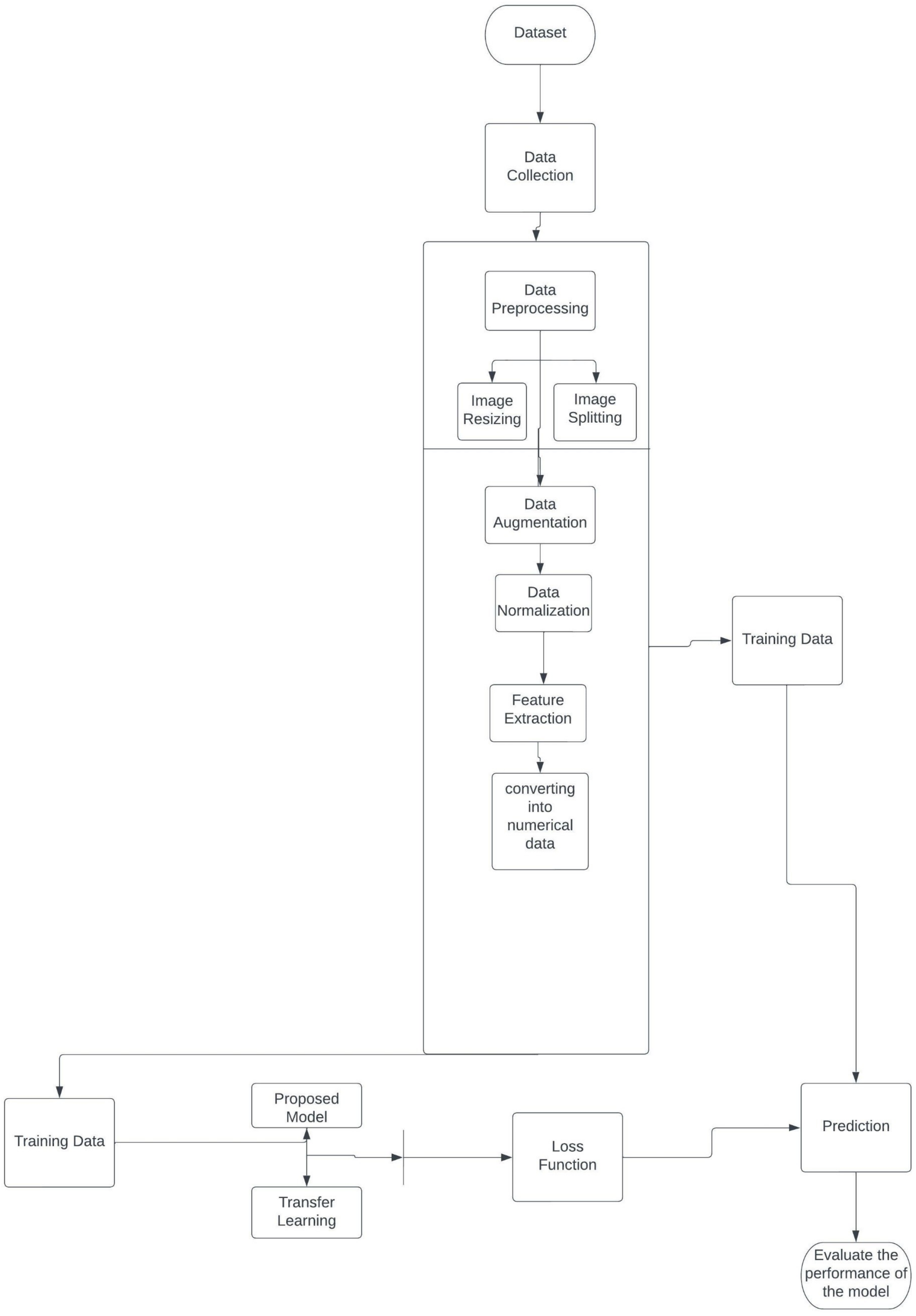


Figure 5.1: Work flow of the Proposed model

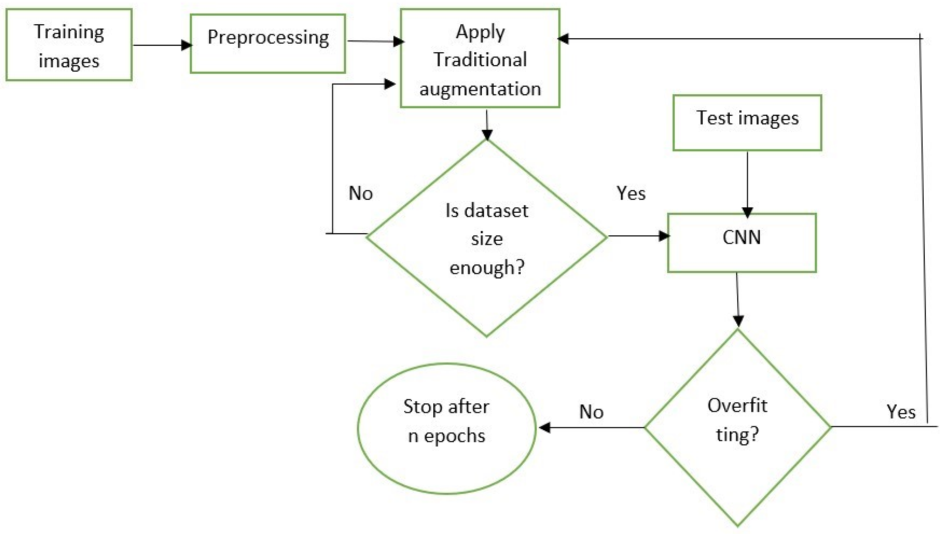


Figure 5.2: Flowchart of Proposed model

5.1.1 Traditional Augmentation

Affine transformations, including translations, rotations, rescaling, shearing, and flips on the original picture, were included in classic augmentation approaches. When applied to the original image, these transformations subtly alter its look without changing its class designation. Due to several limitations, such as overfitting concerns or an effort to improve the model's generalizability, the changes were limited to tiny quantities. Using affine transformation, we created 4000 additional training photos for each class. We then rotated our input images between $[-30 : +30]$ and applied 10% horizontal and vertical shifts, respectively. The rescaling was then accomplished by uniformly distributing values when zooming in and out [1- zoom range, 1+zoom range]. Our horizontal and vertical picture flipping had no impact on the semantic significance of the input photos. On test photos, the conventional enhancement was not used.

5.1.2 Prologue to CNN and its parts

Convolutional neural networks are among the most frequently used deep learning techniques in computer vision. The initial layer of neural networks used in the construction of modern deep learning algorithms extracts the basic information of an image, such as its boundaries, color, and so on. The output of one layer is then used as the input for the layers that follow. Thus, adding additional layers will make learning more difficult. Compared to traditional machine learning methods, deep learning conducts automatic feature extraction and classification. There are various components to a CNN. The first component is a matrix of pixel values from an input image. In our work, RGB (Red, Green, and Blue) visuals have been used. The color plane that separates the RGB image is made up of the colors red, green, and blue. Its three dimensions of it are height, breadth, and channel count. CNNs convert images into the appropriate format to streamline processing while maintaining vital and relevant features [10]. The second component of the CNN architecture is convolution. High-level features are extracted via convolution using many layers, with the lower layers collecting low-level information and the upper levels extracting high-level features. The relative weights of individual characteristics may also be altered by convolution using biases and updated weights. A convolutional layer, also known as a kernel, is composed of many filters and is responsible for carrying out convolution operations in the top layer of the network. As shown in the images, we place a kernel of size K over an area P of an image to perform matrix multiplication and set the filter's size. The stride parameters are configured to allow the filter to travel the necessary distance before changing directions. The filter moves from left to right until the whole image has been seen, then from top to bottom. In this instance, the depth of the kernel is equivalent to the depth of an input picture. After matrix multiplication with filter "K" and image pixel "I," results are summed with bias.[4]

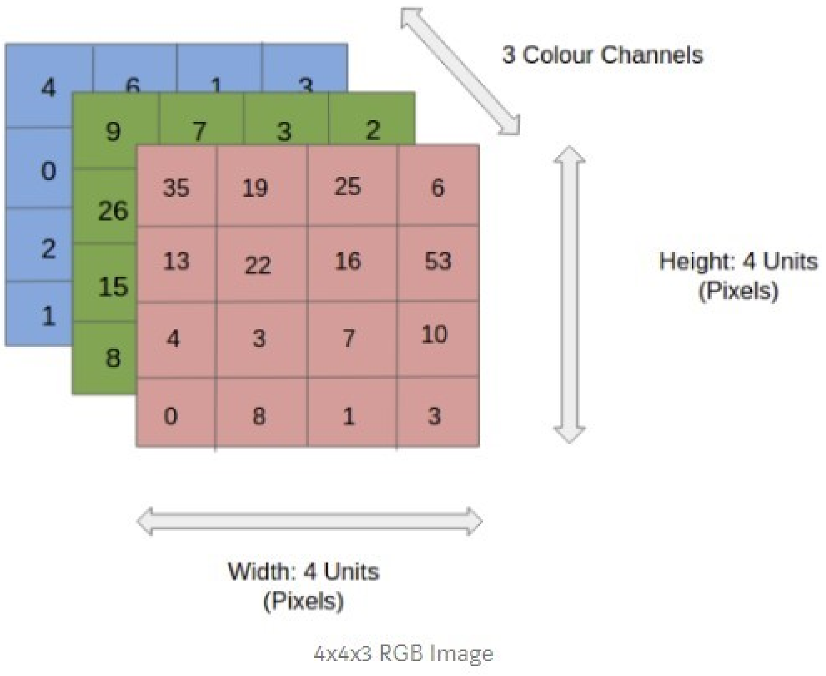


Figure 5.3: RGB image representation [7]

Pooling is in the third position. The pooling is in charge of dimensionality reduction by using a kernel similar to the convolutional layer, reducing the amount of computer resources required to analyze the input and extract the rotational and position invariant dominant features. In this thesis, max pooling has been used for two separate tasks: first, it performs de-noising by reducing dimensionality and suppressing noise, as demonstrated in figures 3–4. It also outputs the highest pixel intensity value for each time the filter covers P pixels of the image.

The convolutional layer and the max-pooling layer are layered together to create the i-th layer of CNN. An activation function makes up the fourth element. Back-propagation is used in neural networks to alter the weights (w), and each neuron's output is the weighted sum of its input (x) plus bias (b).

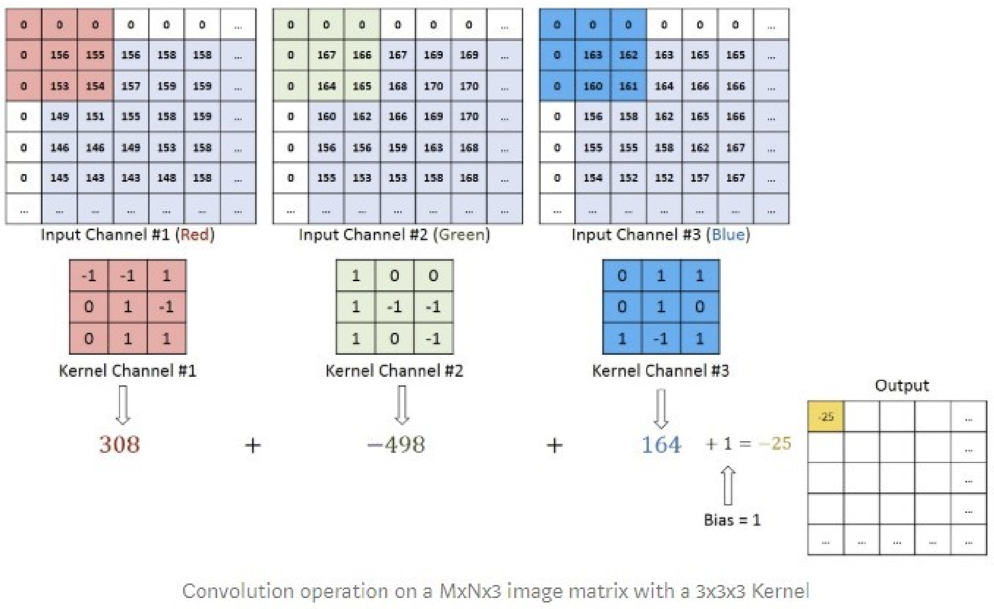


Figure 5.4: Convolution operation on a MxNx3 image matrix with a 3x3x3 Kernel [7]

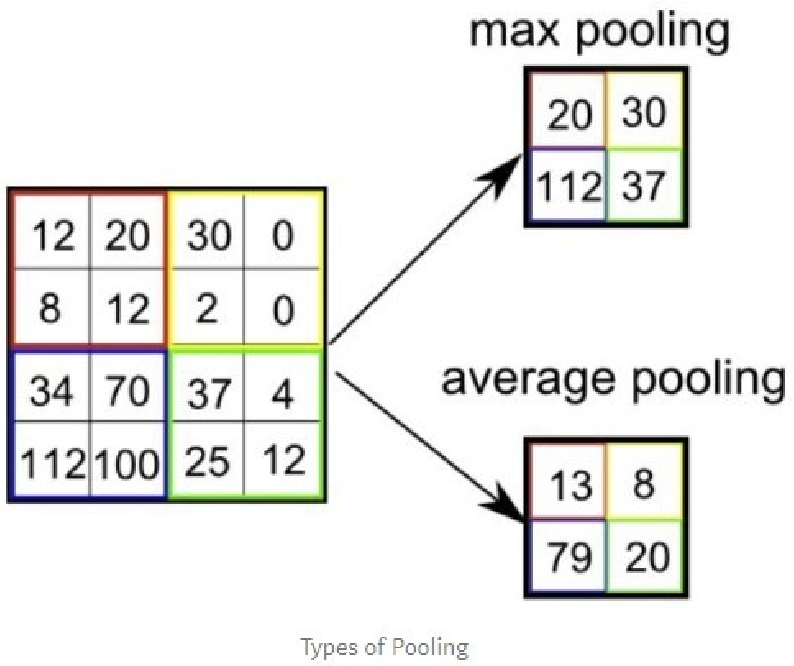


Figure 5.5: Pooling [7]

A function known as the activation function is applied to each neuron's output. ReLU activation function, which is non-linear and has the formula A(x) = max, has been used in this thesis (0,x). The ReLU function returns x when x is positive and 0 in all other situations.

Batch normalization, the fifth element, normalizes the output from the preceding activation layer. It divides the result by the batch standard deviation after deducting the batch means. It makes a neural network more stable. Dropout, a regularization method used to remove the data from unneeded nodes in order to reduce the overfitting layer, is another element of CNN. Except for the output layer, it may be applied to any layer. The likelihood that a layer's outputs will be discarded is referred to as the dropout value. Fully Connected (FC) layers are the second fundamental part of CNN that learns the non-linear combination of high-level information represented by the outputs of convolutional layers after several layers of convolution and max-pooling. All of the activation mechanisms used across the network are connected through this layer. Images are transformed into a helpful form before being flattened into a vector and fed into a feed-forward neural network. Backpropagation is used in feed-forward, and it is applied after each iteration. The CNN model can recognize and differentiate between high-level and low-level characteristics in pictures and classify them using the Softmax classifier after training for several epochs.

For classification, the Softmax classifier employs the cross-entropy loss function and produces likely classes based on the probability distribution. In section 3.3.1, we talk about loss and optimization functions.

Convolutional layer: The convolutional layer filters the pictures using a variety of kernels before sending the results to the next step. It is called the Python Conv2d function. Each kernel produces a different picture. The dot product is extracted from the input picture by sliding the kernel over it. This convolutional layer's primary goal is to extract the image's feature map. The species feature map shown above gives us the feature map. This feature map is afterward supplied to the additional layer[12].

Max Pooling layer: A pooling layer often inherits properties from a convolutional layer. The size of feature maps that are extracted from a convolutional layer might be quite enormous, which raises the cost of computing. Thus, it slows down the process. The feature map's size is reduced using the pooling layer, making the process quicker and less expensive to compute. Various pooling techniques exist, depending on the model.

Dropout layer. This is used to reject a few randomly selected neurons. Accordingly, any weight changes are not sent to the neuron on the return journey, and their effects on the activity of downstream neurons are temporarily eliminated on the forward trip. Dropout is just used to train the model; it is not used to judge the model's aptitude. 2014: A Simple Method for Avoiding Overfitting in Neural Networks. The capacity of the network is limited or lightened during training because the outputs of a layer subject to dropout are randomly subsampled. As a consequence, a bigger network, such as more nodes, may be required when using dropout. Dropout is a further tactic for minimizing the network's overfitting of the training data. After convolutional layers and pooling layers, dropout may be applied. After the pooling layers, dropout is usually used, however, this is simply a recommendation. The accuracy steadily increases as the loss lowers after the dropout rate falls below a certain threshold. The model no longer fits well when dropout rates go over a certain degree.

5.4 Flatten layer: The last layer of CNN is the flattened and dense layer. Although the dense layer only needs a single-dimensional form as input, the convolutional layer produces multidimensional shapes as its output. Since cubic or rectangular forms cannot be used as direct inputs to the next layer.[7] So we must employ the Flatten layer between convolutional layers and dense layers to reduce the multidimensional matrix to a single-dimensional matrix.

Dense layer: This layer of a neural network is densely linked, which means that every neuron in each layer is connected to every neuron in the layer above it. Before moving on to the dense layer, the picture will travel through each convolutional layer and pooling layer. We will flatten the multidimensional output into a one-dimensional output and then send it as input to the dense layer. All of the neurons in this layer of neurons will get input from the neurons in the layer below. The convolutional layer's output is used by the dense layer to classify the picture. In a neural network, each layer is made up of neurons that weigh the input before passing it to a nonlinear function called an "activation function," the result of which is regarded as the neuron's output. The output of the picture is thought to be the last dense layer. The output layer furthermore contains a softmax activation feature. When there are two or more classes, this function is required. Each neuron belongs to a certain class. Each neuron will return the corresponding probability of the input picture for all classifications. The output will be the class with the highest probability.

5.1.3 The initiated CNN Architecture

Table 3.1 displays the suggested CNN architecture for the lung lesion categorization system. A picture's input size is 64x64x3 pixels. There are three primary convolutional blocks in the design, and the second and third blocks include many convolutional layers to further the neural network and avoid overfitting. Batch normalization was used after each activation, and dropout was used after each pooling. An initial architectural block consists of a convolutional layer with 32 3x3 filters. ReLU activation and Batch Normalization (BN) are the steps that come after it. After BN, Maxpooling2D is used to minimize the spatial dimension[28].

After pooling, a dropout of 0.25 is used to minimize overfitting. The second block consists of two convolution layers, each of which includes 64 3x3 filters stacked together and is followed by ReLU activation and BN. The second block's last BN is followed by the application of Maxpooling2D and dropout. These first two blocks teach the fundamental attributes of a picture, while the third block teaches the more profound and richer features. According to table 3.1, the third block stacks three convolutional layers, ReLU activation, BN, and final pooling. Following the third block's completion, a 256-node standard feedforward network is constructed utilizing completely linked layers. We execute BN once an ultimately linked layer has been formed, then we drop out by 0.5. The SoftMax classifier is introduced in the last block. An adaptive gradient (Adagrad) optimizer with a learning rate of 0.001 was used for the training. Binary cross-entropy was utilized as a cost function since the task was a two-class classification problem.

5.2 ReLU Activation Function

The activation function selects the first neuron to be activated. The most often used activation technique is relu. The non-linear activation function known as the Rectified Linear Unit, or RELU, is gaining popularity. Deep neural networks and multi-layer neural networks are used the most often. According to the ReLU equation, the output is the highest value between 0 and the input value. When the input is positive and zeroes when it is negative, the output is equal to the input. The vanishing gradient problem is handled by ReLU. As the number of layers increases, the ReLU function does not result in the vanishing gradient issue. A quick or substantial change occurs when there is a different attribute. All negative numbers have been converted to zero, therefore there are no negative values. Last but not least, since the ReLU function's derivative is 1 for positive input, compared to other activation functions, it may facilitate the training of deep neural networks more quickly.

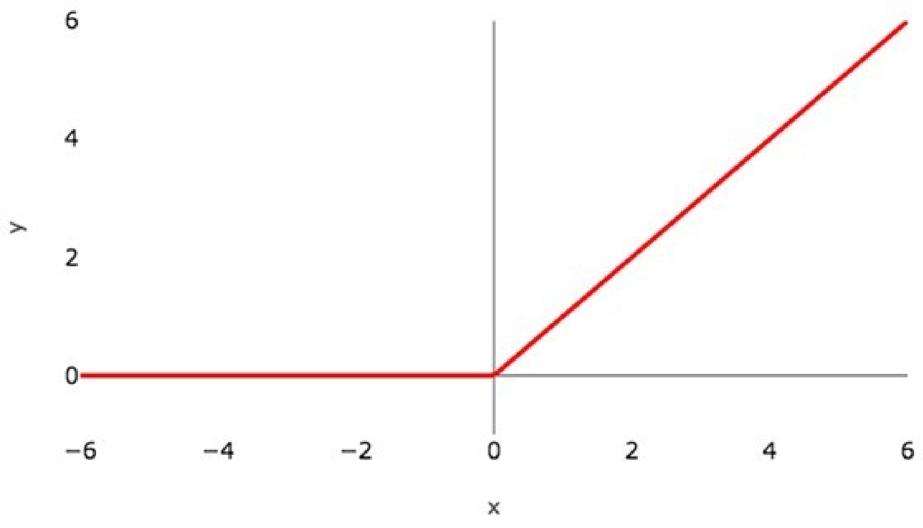


Figure 5.6: Graph of activation function ReLU [7]

5.2.1 Loss and the Function of Optimization

By contrasting the forecast with reality, the loss function—also referred to as the cost/error function—is repeatedly evaluated. The machine learning model aims to identify variables and weights that will minimize the cost function. For a given

weight and bias, the cost function in our model may be stated as follows: where x represents training instances, y is labeled (0=benign, 1=malignant), and p(y) is the projected probability that y is malignant.
The optimization function, which looks for the cost function's global minima, seeks to minimize the cost function. The widely used optimization method is known as gradient descent. Gradient descent enables a model to iteratively learn the gradient/direction toward minimizing errors by modifying its parameters and weights throughout the training process.

The model iterates until it achieves a minimum when modifying the parameters and weights has little to no impact on the loss. This causes the model to converge, as illustrated in Fig. 3-2, where J(w) represents the weight function and cost function C. Stochastic Gradient Descent (SGD), an optimization technique, has been applied in our model. Each training sample updates the parameter, and the label and is supplied by:

$$W_m = W_m - \alpha * \frac{dC(W, B}{dW_m} \tag{5.1}$$

$$B_k = B_k - \alpha * \frac{dC(W, B}{dB_k} \tag{5.2}$$

Here, the parameter is updated at a learning rate of 0.001 and is represented in equations 3.2 and 3.3. k stands for the total number of biases (B), and m stands for the total number of weights (W).

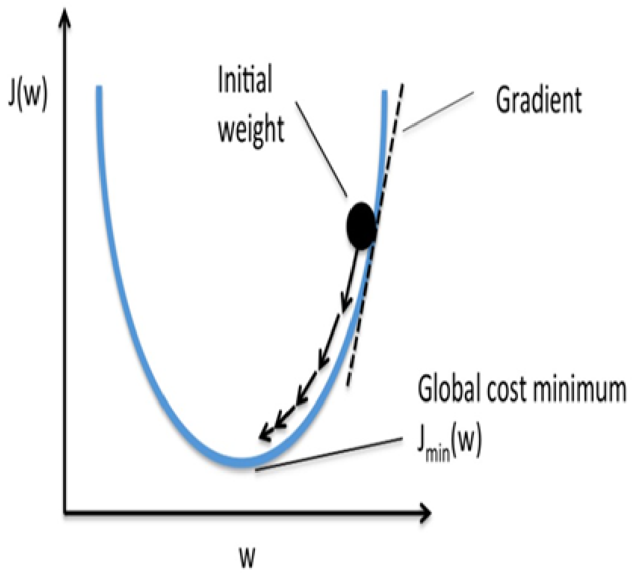


Figure 5.7: Convergence of CNN [7]

5.2.2 Execution of Traditional Augmentation

The CNN classifier was used in conjunction with traditional augmentation to categorize skin lesions into benign and malignant conditions. Here's more information:

- Load the training data with labels.
- Resize the training dataset using aspect-aware preprocessors to 64×64×3.
- Save photos in HDF5 (Hierarchical Data Format 5) after resizing them.
- Subtract the mean Red, Green, and Blue pixel intensities from the input before applying the mean preprocessor, which is used to normalize data.
- Use conventional enhancement to create four distinct photos from one.
- Incorporate the CNN model.
- To prevent overfitting, use a patch preprocessor to remove M N pixel patches from the picture during training.
- Use an adaptive gradient descent optimizer and binary cross entropy as a cost function to train the CNN model.
- To test photos, use a crop preprocessor.
- Predict the CNN model's output using test pictures.

5.2.3 Adamax optimizer

The Adamax optimizer employs the infinity norm and is a variation of the Adam algorithm. In certain situations, Adamax performs better than Adam, especially in models with embedding. Adam updates individual weights by scaling their gradients inversely proportionate to an L2 norm of their most recent and previous gradients. It is possible to convert the L2 norm-based update rule into an Lp norm-based update rule. Such variants exhibit numerical instability with a more significant It has been shown that the ut value in Adam with 1 merges to a progressively steady value.

$$u_t = \beta_2^{\infty}.u_t + 1 + (\beta_2^{\infty}).|g_t|^{\infty} = max(\beta_2.u_{t-1}, |g_t|) \tag{5.3}$$

Chapter 6

Implementation and Discussion

In this chapter, we will be focusing on the experimental setup along with the performance metrics and the results following the discussion. Here in the discussion we also present the comparative analysis with other related works.

6.1 Experiment Setup

The experiments have been conducted on the Google Co-laboratory workspace. Google Co- the laboratory is a cloud computing workspace that enables anyone to write and execute programs written in python which also provides GPU. So it is an ideal environment for developing and debugging CNN Models. We have used the PyTorch library for developing our classifier model. Each experiment runs in co-lab GPU for 50 epochs, batch size of 16 images, learning rate = 0.0001. We have utilized cross-entropy loss for measuring the classification error at each iteration and Adam optimizer for gradient updates of the model.
Execution Assessment Frameworks of CNN Models

6.2 Performance Metrics

For our model, we utilized several CNN architectures. They don't all perform at the same level. Several models outperformed others. Accuracy provides an estimate of these designs' performance. In this chapter, we first discussed the performance measures before disclosing how each CNN design performed according to those metrics.

6.2.1 Accuracy

A classification machine learning model's accuracy is measured as a performance metric. By examining the parameters, one may gauge which machine learning model is the most accurate at predicting a label.

$$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}} \tag{6.1}$$

6.2.2 Precision

Precision demonstrates the genuine upsides partitioned by the total number of positive expectations. It estimates how well a model characterizes an example as confident. At the point when the model produces numerous positive incorrect characterizations or few precise positive arrangements, the denominator increments, and the preciseness diminishes. The accuracy is high when the model makes a critical level of positive characterizations and less inaccurate positive classifications.

$$Precision = \frac{True_{positive}}{True_{positive} + False_{negative}} \tag{6.2}$$

6.2.3 Recall

Recall decides the model's capacity to characterize positive pieces of information. The higher the review, the more noteworthy number of positive examples being characterized. Negative characterizations don't influence a review.

$$Recall = \frac{True_{positive}}{True_{positive} + False_{negative}} \tag{6.3}$$

6.3 Result

From the results, we can observe that most of the classification performance achieves above 90% test accuracy. The following figures have been acquired for each model i.e. ResNet18, VGG19, Shufflenet, Inception_V3, Densenet161, and MobileNet_V2 through implementation. The figures include confusion matrices, accuracy and loss curves, and affected lung images after the implementation of each model.

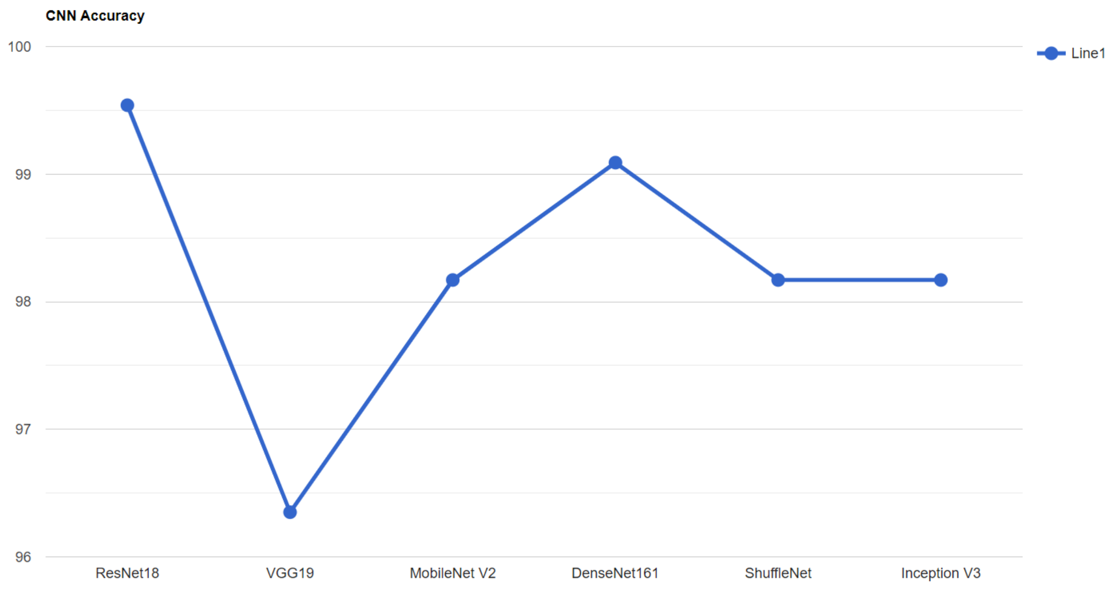


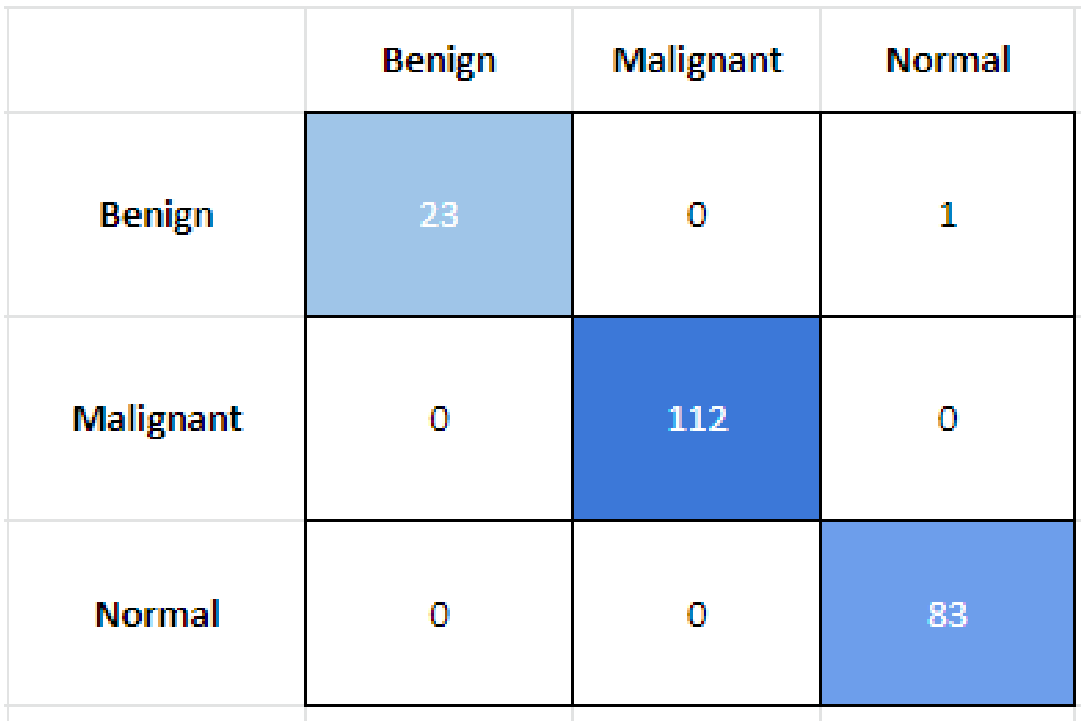
Figure 6.1: The Overall accuracy line graph of the CNN classifiers

6.3.1 ResNet18

We can observe that the ResNet18 outperforms all other proposed classifiers. The accuracy for ResNet18 is 99.54% which is higher than other proposed classifier accuracy. The overall accuracy table is given below:

	Accuracy	**Precision**	**Sensitivity**	**F1_score**	**Specificity**
Benign Cases	99.54	100	95.83	97.87	100
Malignant Cases	100	100	100	100	100
Normal Cases	99.54	98.81	100	99.4	99.26
Weighted Average	99.78	99.55	99.54	99.54	99.72
Macro Avg	99.69	99.6	98.61	99.09	99.9
Overall Accuracy	99.54				

Table 6.1: The performance evaluation for ResNet18



	Benign	Malignant	Normal
Benign	23	0	1
Malignant	0	112	0
Normal	0	0	83

Figure 6.2: The confusion matrix of ResNet18

We can observe from figures 6.3 and 6.4 representing the ResNet training and validation accuracy and loss graph, that the curve of the accuracy graph is fluctuating during the first 10 epochs, and also two small fluctuation bursts take place between the 20-24 epochs and 34 - 38 epochs. Meanwhile, the rest of the curve remains converging. However, in the case of the loss graph, the curve is more or less converging except it fluctuates from 35 - 40 epochs. On the contrary, it gives us a higher overall accuracy.

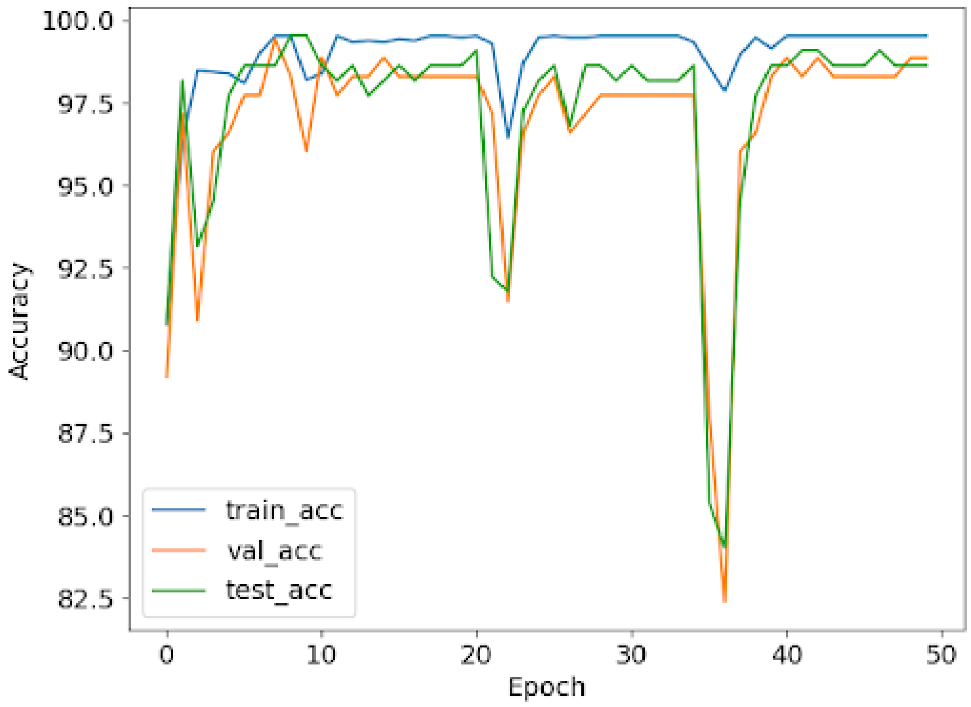


Figure 6.3: The training, validation, and test accuracy graph of ResNet18

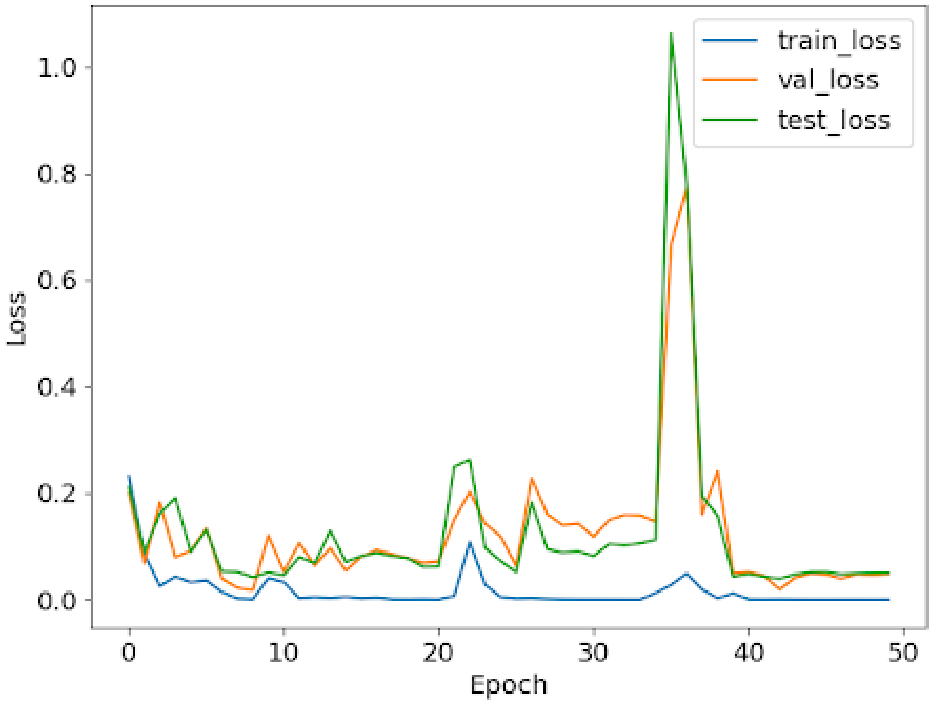


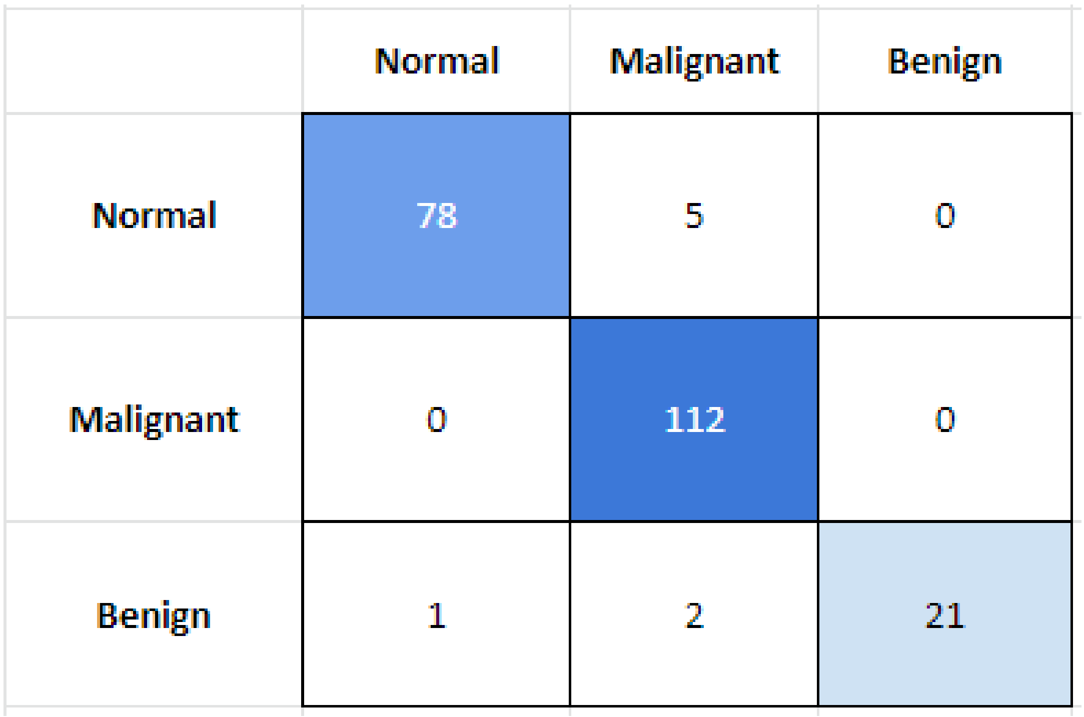
Figure 6.4: The training, validation, and test loss graph of ResNet18

6.3.2 VGG19

The performance of VGG19 is a little bit mediocre where it is giving us an accuracy of 96.35%. One probable reason is that VGG is prone to vanishing gradient problems. The overall accuracy is provided in the following table:

	Accuracy	**Precision**	**Sensitivity**	**F1_score**	**Specificity**
Normal Cases	97.26	98.73	93.98	96.3	99.26
Malignant Cases	96.8	94.12	100	96.97	93.46
Benign Cases	98.63	100	87.5	93.33	100
Weighted Average	97.17	96.51	96.35	96.32	96.37
Macro Avg	97.56	97.61	93.82	95.53	97.57
Overall Accuracy	96.35				

Table 6.2: The performance evaluation for VGG19



	Normal	Malignant	Benign
Normal	78	5	0
Malignant	0	112	0
Benign	1	2	21

Figure 6.5: The confusion matrix of VGG19

Again for VGG19, we can observe from figures 6.6 and 6.7 that the curve of the accuracy graph is fluctuating throughout all epochs which refers to it as overfitting. On the other hand, though the loss graph is fluctuating all the way, it is decreasing. As a result, it is not underfitting in contrast to its mediocre accuracy.
Again for VGG19, we can observe from figures 6.6 and 6.7 that the curve of the accuracy graph is fluctuating throughout all epochs which refers to it as overfitting. On the other hand, though the loss graph is fluctuating all the way, it is decreasing. As a result, it is not underfitting in contrast to its mediocre accuracy.

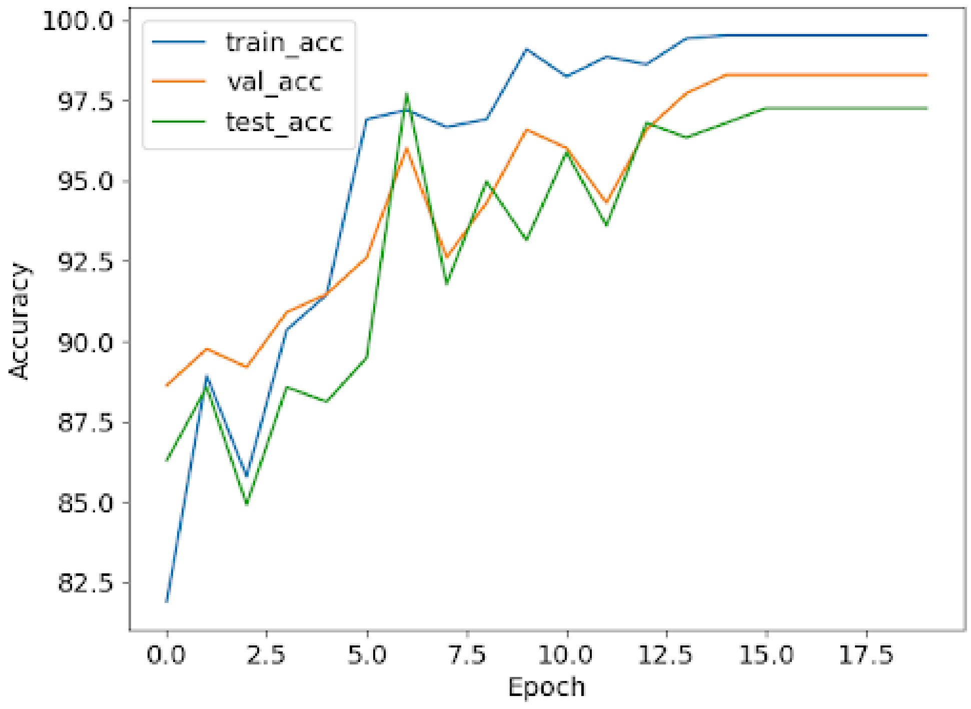


Figure 6.6: The training, validation, and test accuracy graph of VGG19

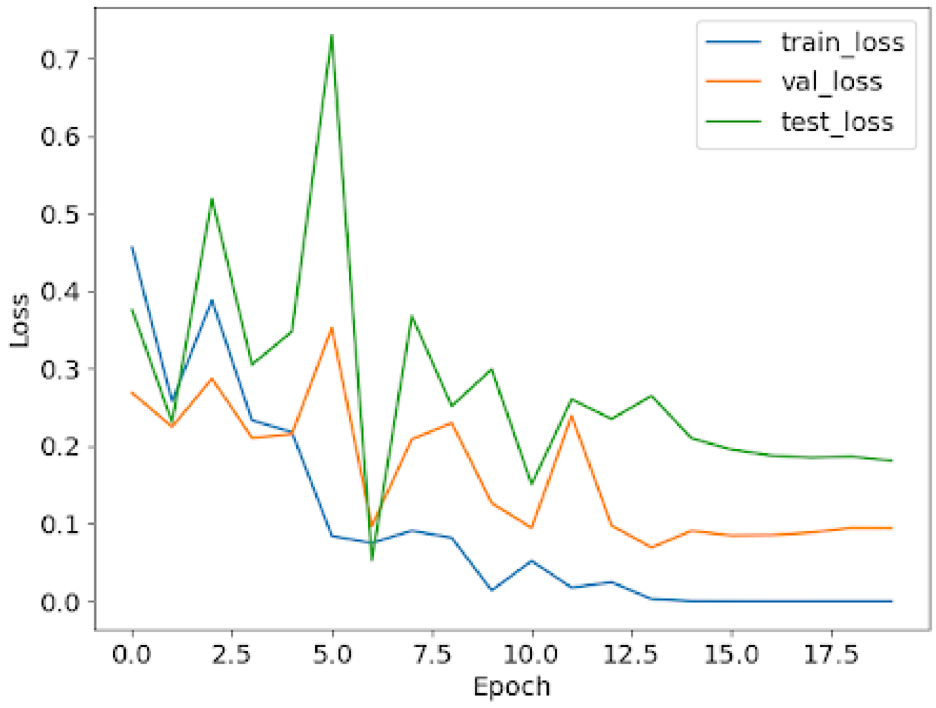


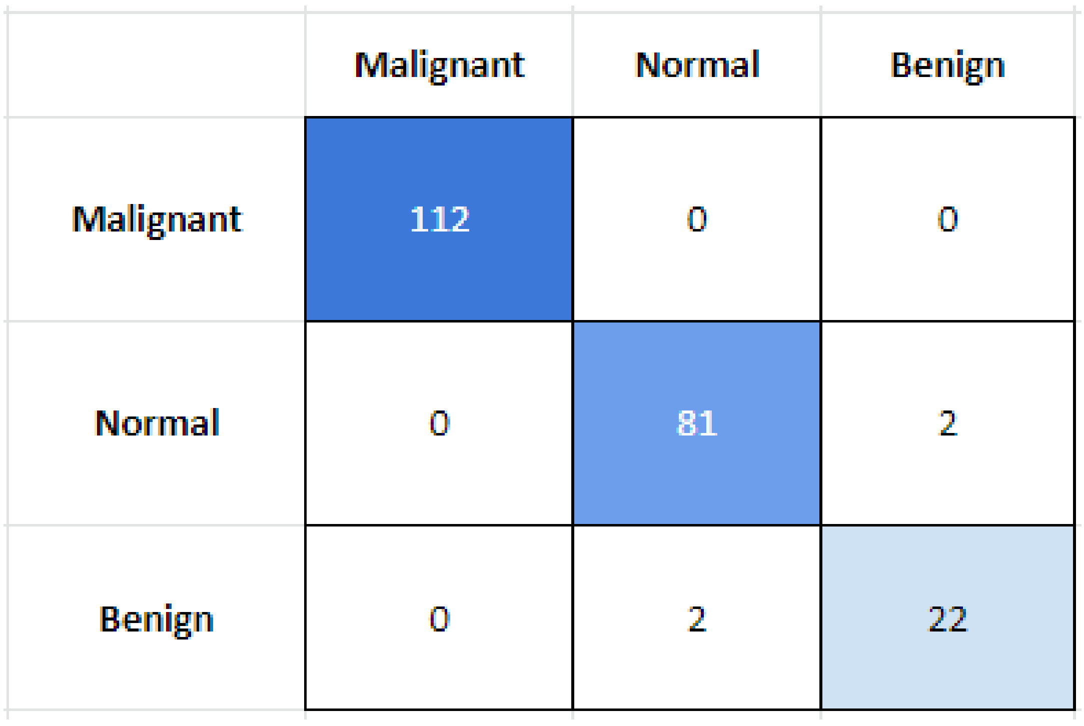
Figure 6.7: The training, validation, and test loss graph of VGG19

6.3.3 MobileNet_V2

The MobileNet_V2 model transpires with an accuracy of 98.17%. Nevertheless, it has fewer parameters than the other model, so it is faster to train and takes less computational resources and achieves good classification performance. So a faster low latency model can be developed by utilizing MobileNet. The accuracy table is given below.

	Accuracy	**Precision**	**Sensitivity**	**F1_score**	**Specificity**
Malignant Cases	100	100	100	100	100
Normal Cases	98.17	97.59	97.59	97.59	98.53
Benign Cases	98.17	91.67	91.67	91.67	98.97
Weighted Average	99.11	98.17	98.17	98.17	99.33
Macro Avg	98.78	96.42	96.42	96.42	97.55
Overall Accuracy	98.17				

Table 6.3: The performance evaluation for MobileNetV2



	Malignant	Normal	Benign
Malignant	112	0	0
Normal	0	81	2
Benign	0	2	22

Figure 6.8: The confusion matrix of MobileNet_V2

Now in the case of MobileNet_V2, we can observe from figures 6.9 and 6.10 that the curve of the accuracy graph is fluctuating throughout all epochs but with higher training accuracy, which still refers to it as overfitting. On the other hand, the loss graph immensely fluctuates all the way, it is increasing at the end. Thus, it is not underfitting even though it gives better accuracy.

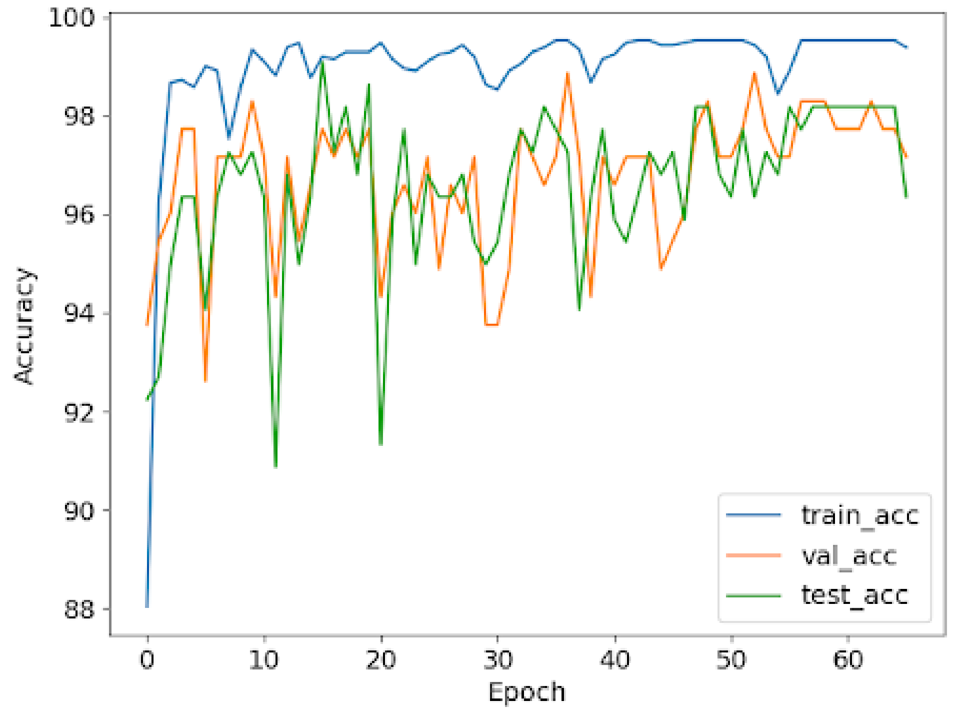


Figure 6.9: The training, validation, and test accuracy graph of MobileNet_V2

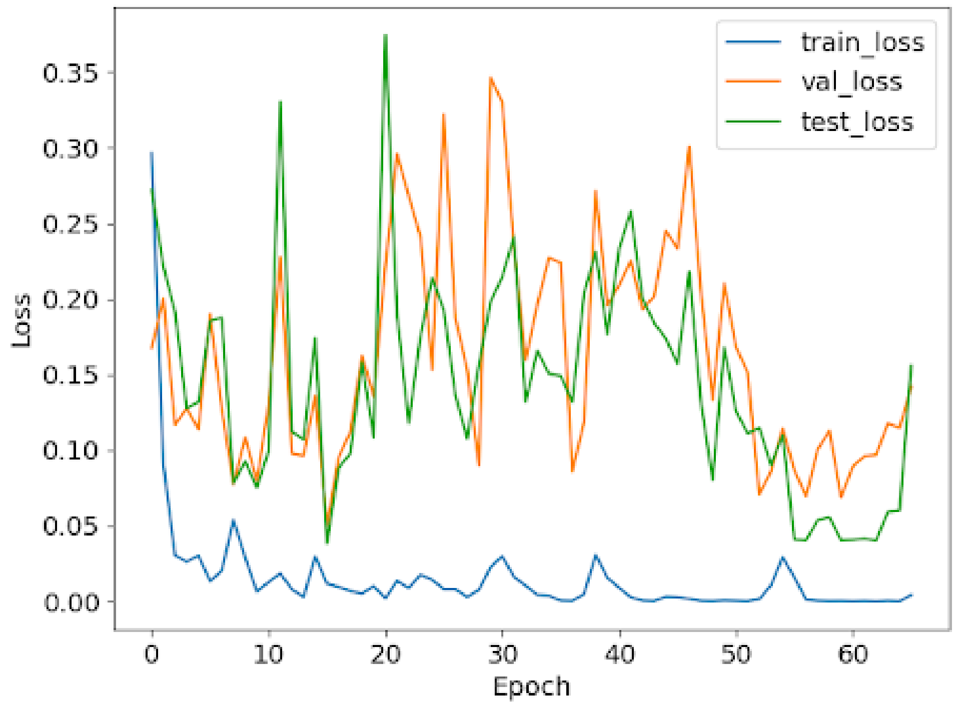


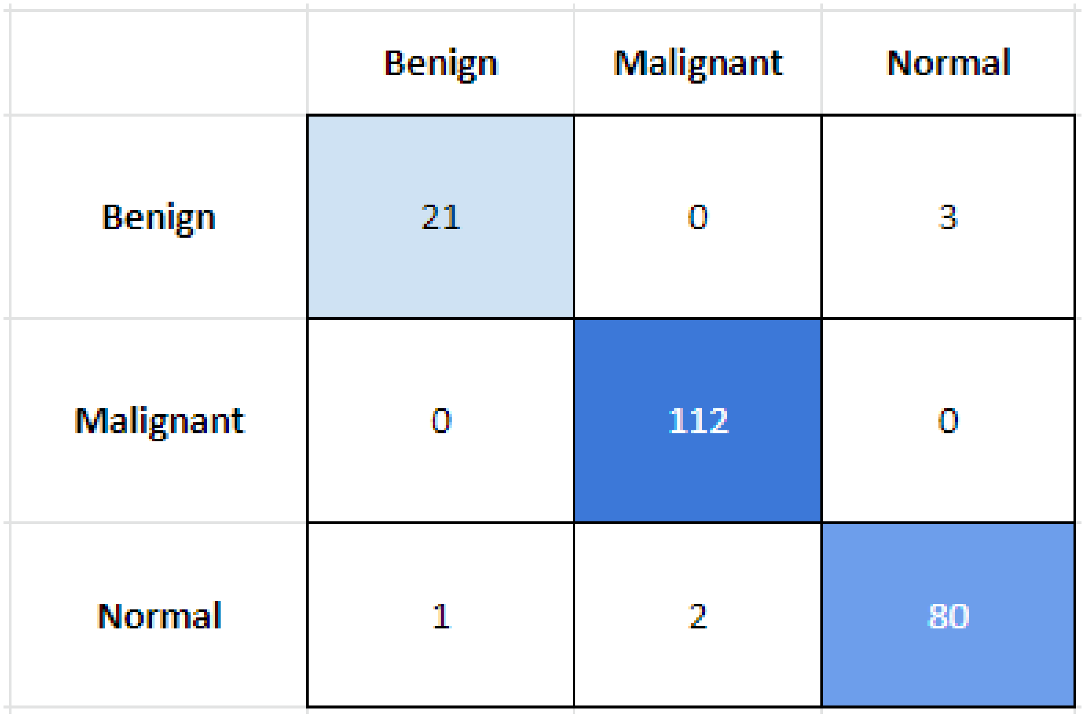
Figure 6.10: The training, validation, and test loss graph of MobileNet_V2

6.3.4 DenseNet161

Then, for DenseNet161 where the accuracy of DenseNet161 is 99.09%. Here the propagation of the input layer to the subsequent layers proves to have beneficial attributes as the front layers are able to learn more features and increase their accuracy. In contrast, the 161 layers that it consists of are sometimes not suitable to be used on a simple single GPU computer as they can slow down the running time.

	Accuracy	**Precision**	**Sensitivity**	**F1_score**	**Specificity**
Benign Cases	99.09	95.83	95.83	95.83	99.49
Malignant Cases	100	100	100	100	100
Normal Cases	99.09	98.8	98.8	98.8	99.26
Weighted Average	99.56	99.09	99.09	99.09	99.66
Macro Avg	99.39	98.21	98.21	98.21	97.55
Overall Accuracy	99.09				

Table 6.4: The performance evaluation for DenseNet161



	Benign	**Malignant**	**Normal**
Benign	21	0	3
Malignant	0	112	0
Normal	1	2	80

Figure 6.11: The confusion matrix of DenseNet161

In the case of DenseNet161, the accuracy curve in figures 6.12 and 6.13 is also fluctuating almost similar to ResNet18 with some bursts within these 50 epochs and is overfitting. However, the loss graph is also fluctuating but, we perceived that it also gives some huge bursts in the case of training, validation, and test loss.

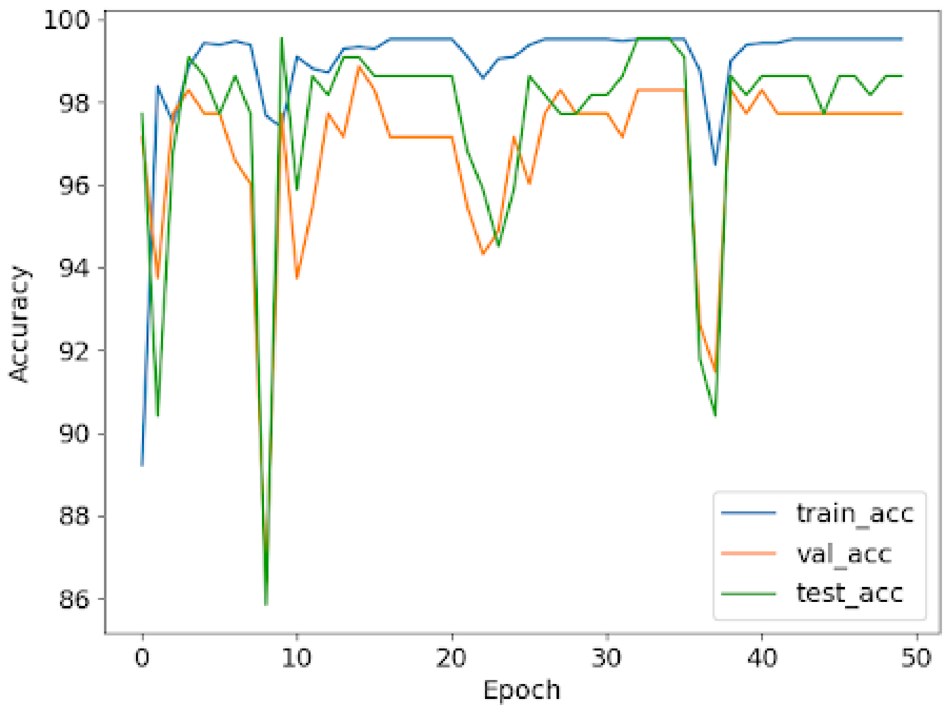


Figure 6.12: The training, validation, and test accuracy graph of DenseNet161

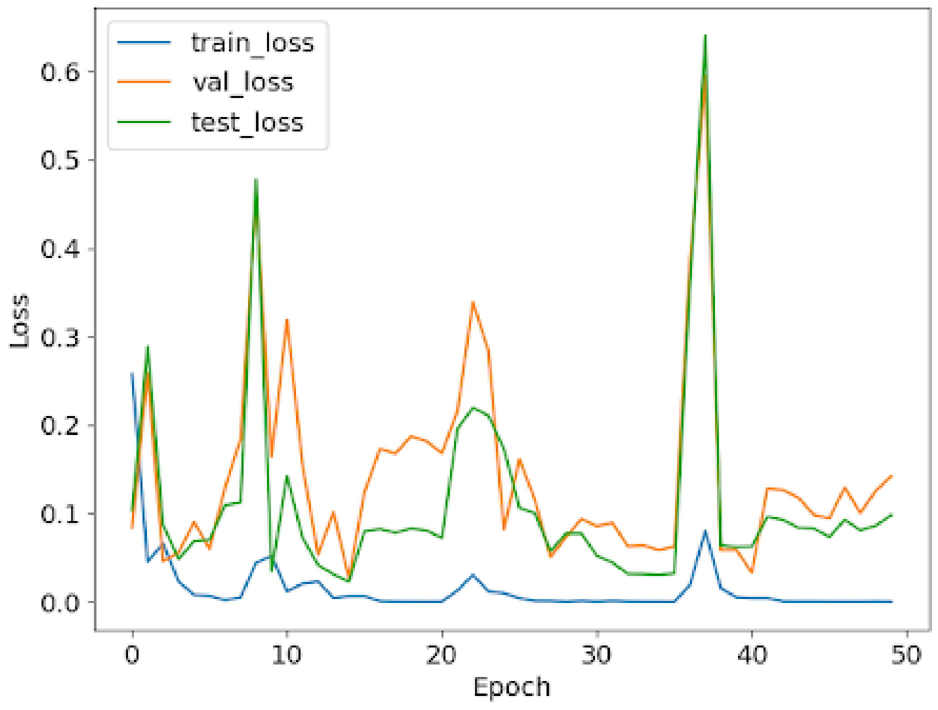


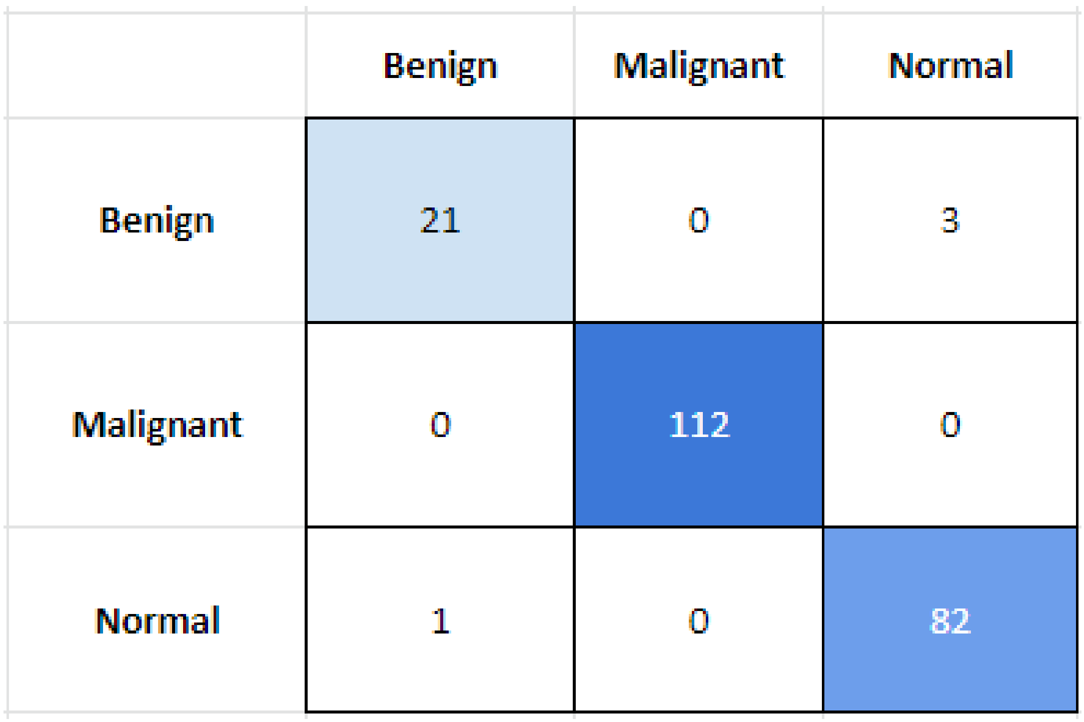
Figure 6.13: The training, validation, and test loss graph of DenseNet161

6.3.5 ShuffleNet

The following performing classifier is ShuffleNet with an exactness of 98.17%. Here, it uses point wise bunch convolution and channel mix to diminish calculation costs while keeping up with exactness. It figures out how to get a lower top-1 blunder than that of the MobileNet framework on ImageNet characterization.

	Accuracy	**Precision**	**Sensitivity**	**F1_score**	**Specificity**
Benign Cases	98.17	95.45	87.5	91.3	99.49
Malignant Cases	100	100	100	100	100
Normal Cases	98.17	96.47	98.8	97.62	97.79
Weighted Average	99.11	98.16	98.18	98.14	99.11
Macro Avg	98.78	97.3	95.43	96.3	97.55
Overall Accuracy	98.17				

Table 6.5: The performance evaluation for ShuffleNet



	Benign	Malignant	Normal
Benign	21	0	3
Malignant	0	112	0
Normal	1	0	82

Figure 6.14: The confusion matrix of Shufflenet

Here, for ShuffleNet in figures 6.15 and 6.16, the curve is fluctuating but can still be considered to be converging. Thus, it can be considered to not be overfitting. The same goes for the loss graph as well, the curve tends to be converging and is also decreasing. Therefore it is defined that it is not underfitting and the amount of data is considered sufficient for this model.

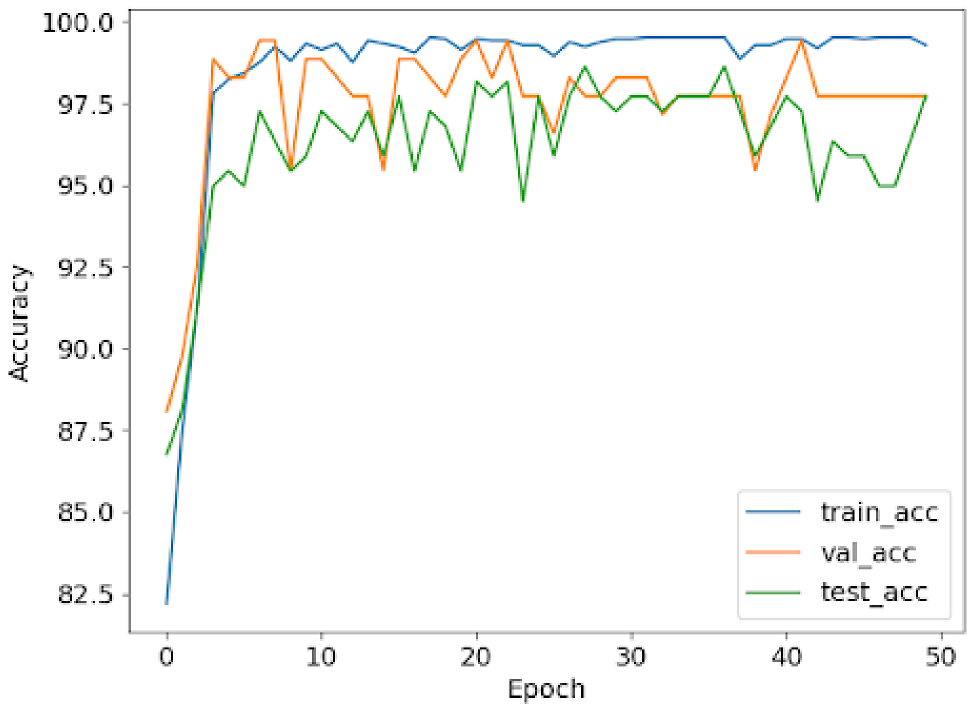


Figure 6.15: The training, validation, and test accuracy graph of Shufflenet

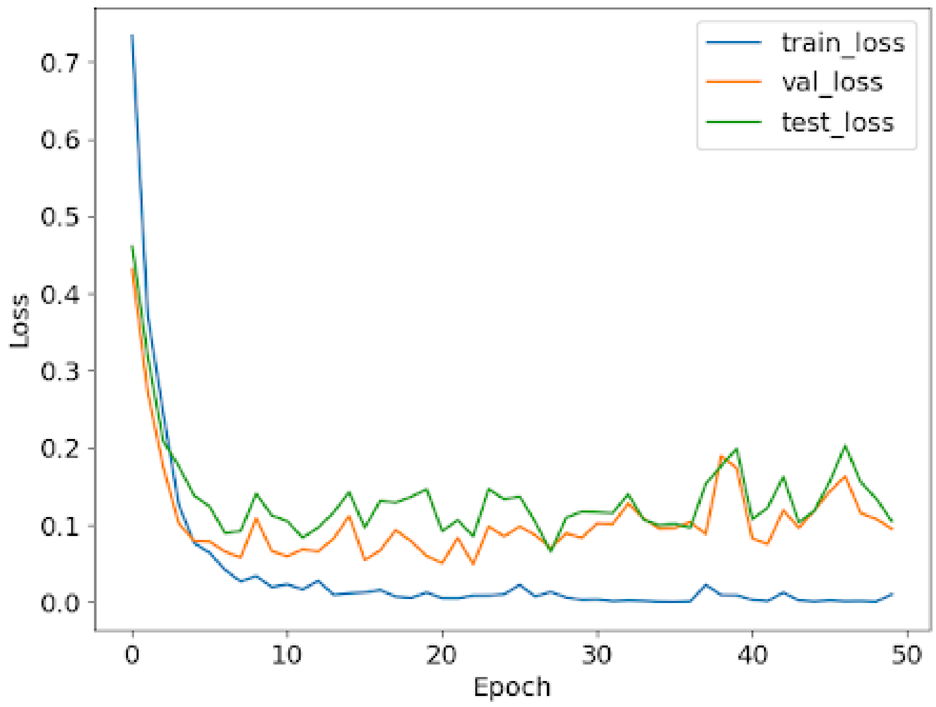


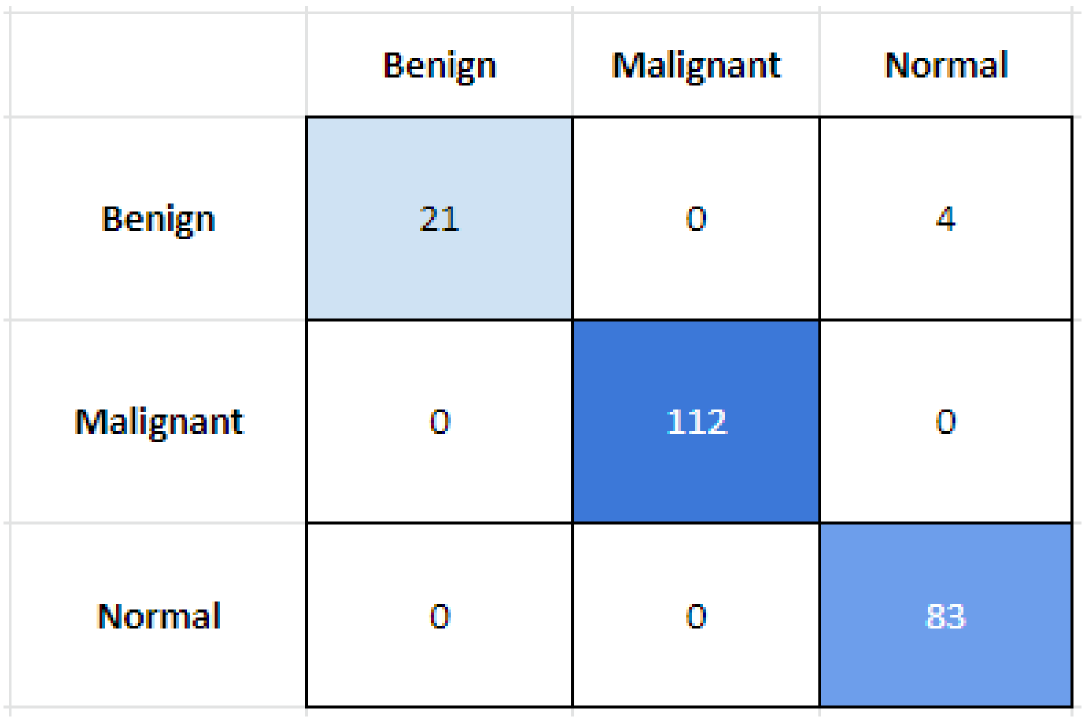
Figure 6.16: The training, validation, and test loss graph of Shufflenet

6.3.6 Inception_V3

The Inception_V3 also came out with an accuracy of 98.17% similar to ShuffleNet. Here, the convolutional neural network which is 48 layers deep, performs factorization into smaller convolutions, and the version of the network is trained on the augmented images.

	Accuracy	**Precision**	**Sensitivity**	**F1_score**	**Specificity**
Benign Cases	98.17	100	83.33	90.91	100
Malignant Cases	100	100	100	100	100
Normal Cases	98.17	95.4	100	97.65	97.06
Weighted Average	99.11	98.26	98.17	98.11	98.89
Macro Avg	98.78	98.46	94.44	96.18	99.02
Overall Accuracy	98.17				

Table 6.6: The performance evaluation for InceptionV3



	Benign	Malignant	Normal
Benign	21	0	4
Malignant	0	112	0
Normal	0	0	83

Figure 6.17: The confusion matrix of InceptionV3

In figures 6.18 and 6.19, which depict the Inception_V3 training and validation accuracy and loss graphs, show that the curve of the accuracy graph fluctuates with fluctuation bursts occurring between the 50 epochs. The curve is more or less convergent in the case of the loss graph, but it varies between the epochs.

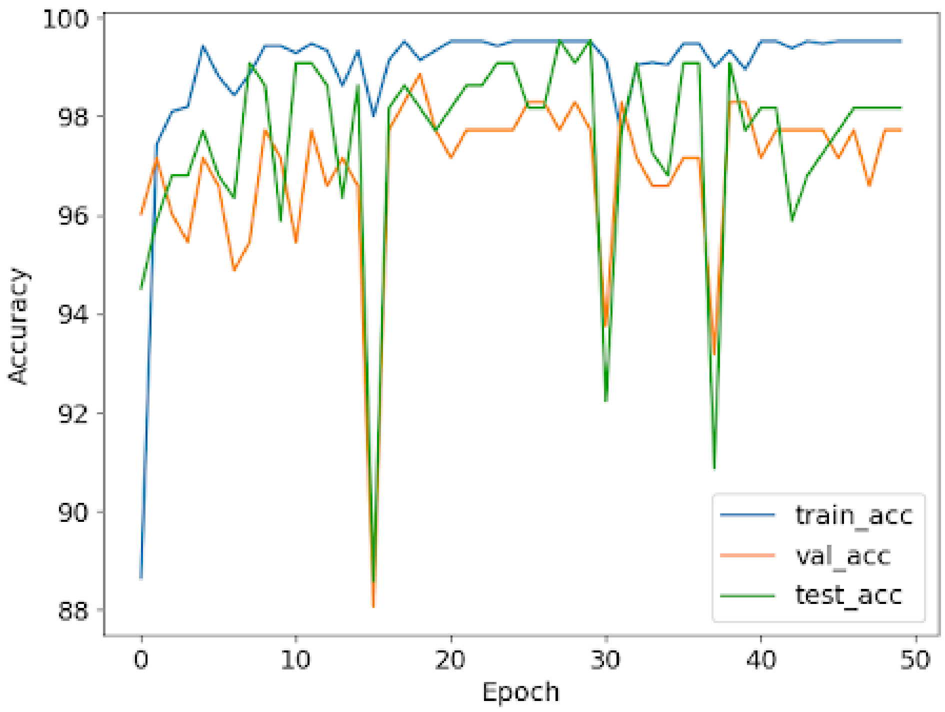


Figure 6.18: The training, validation, and test accuracy graph of InceptionV3

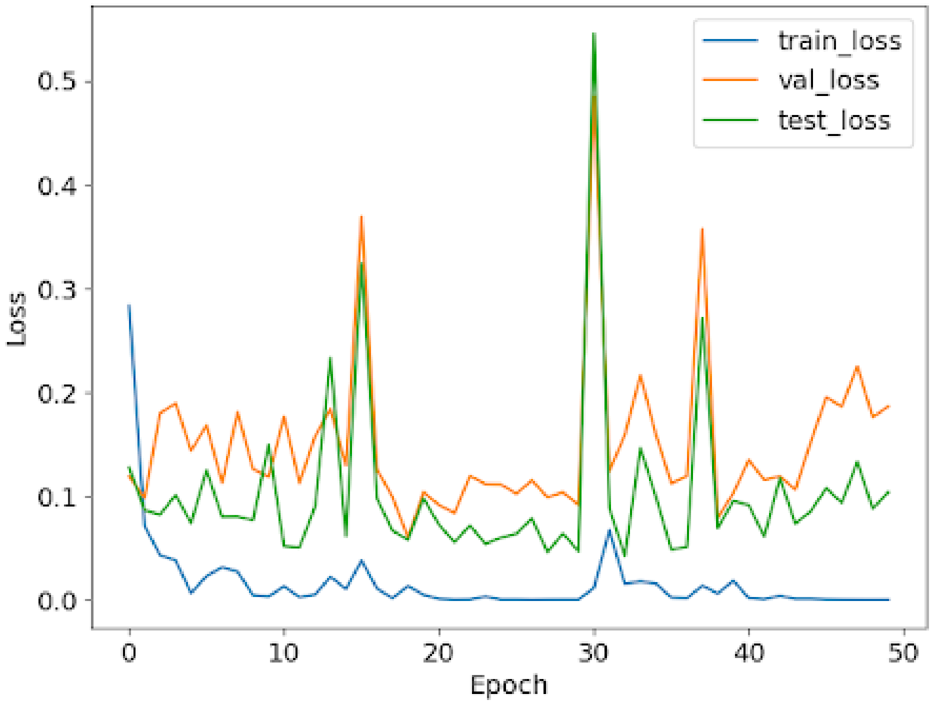


Figure 6.19: The training, validation, and test loss graph of InceptionV3

6.4 Hog Feature Extraction

We have implemented the Hog feature extraction using the above architectures. Compared to the above results, we expected the feature extraction will give us a better result. However, that was not the case and we have witnessed that except for VGG19, the CNN accuracies were comparatively better than that of the feature extraction.

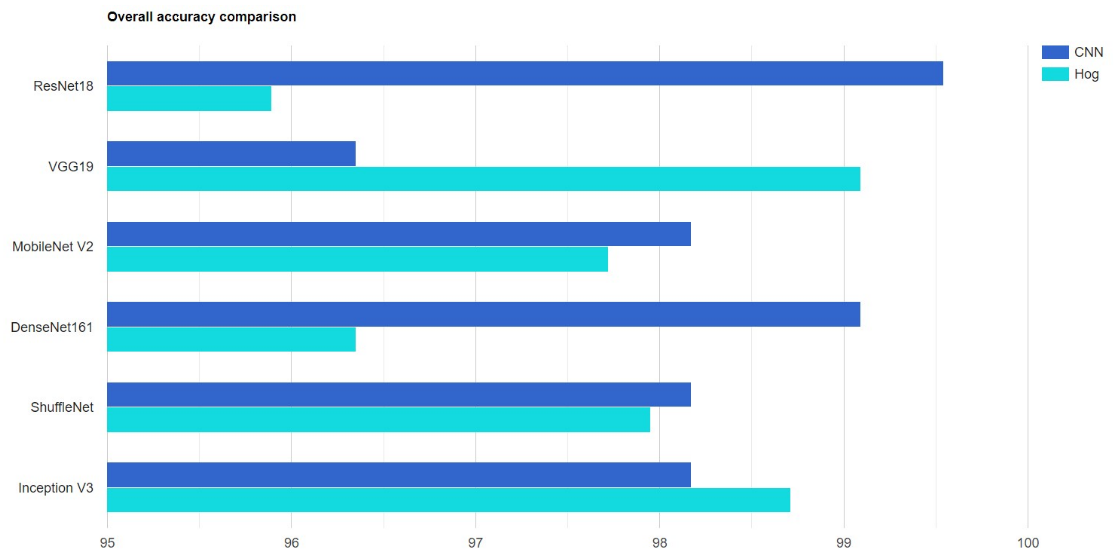


Figure 6.20: Comparison of the CNN classifiers and Hog Feature Extraction models

Moreover, we can observe that HOG focuses on the structure of the object. It extracts information about the edges' magnitude as well as the orientation of the edges. Here, it uses a detection window of 64x128 pixels, so the image is first converted into a (64, 128) shape. The image is then further divided into small parts, and then the gradient and orientation of each part is calculated. Next, it is divided into 8x16 cells into blocks with 50% overlap, so there are $7x15 = 105$ blocks in total, and each block consists of 2x2 cells with $8x8$ pixels. Finally, we take the 64 gradient vectors of each block ($8x8$ pixel cell) and put them into a 9-bin histogram.

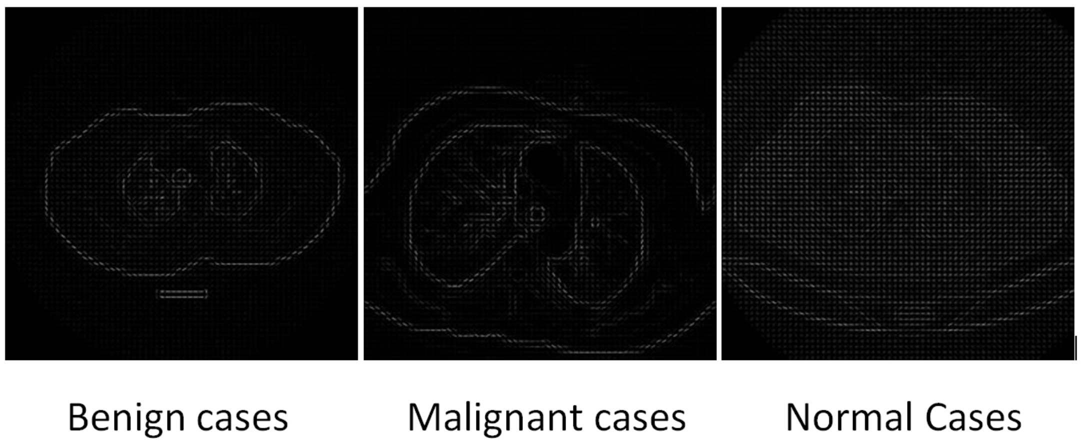


Figure 6.21: Extracted images after performing HOG

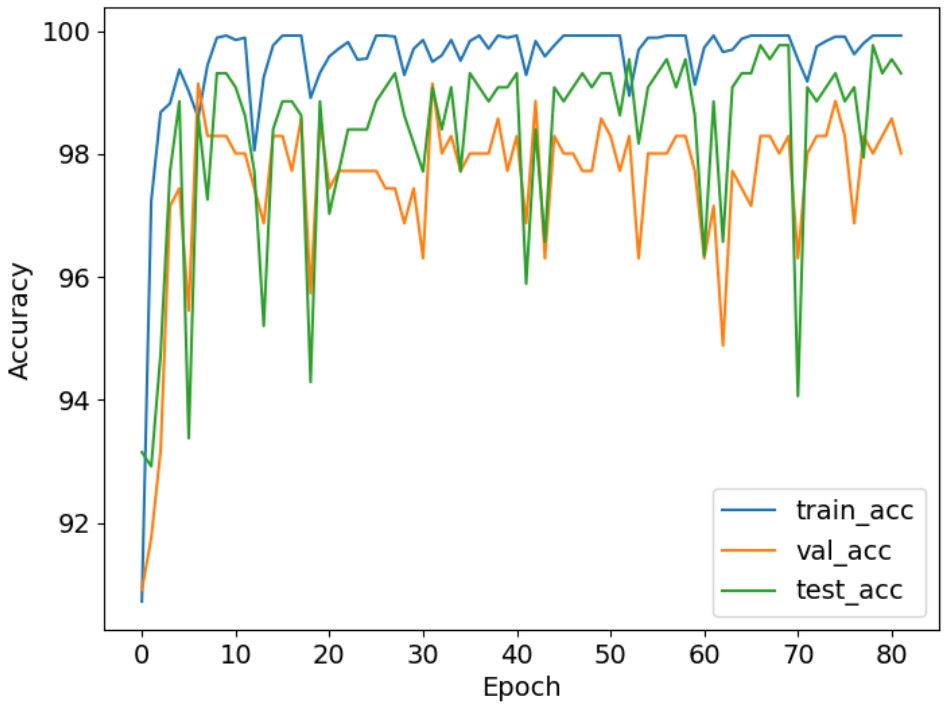


Figure 6.22: The training, validation, and test accuracy graph of ResNet18 for HOG

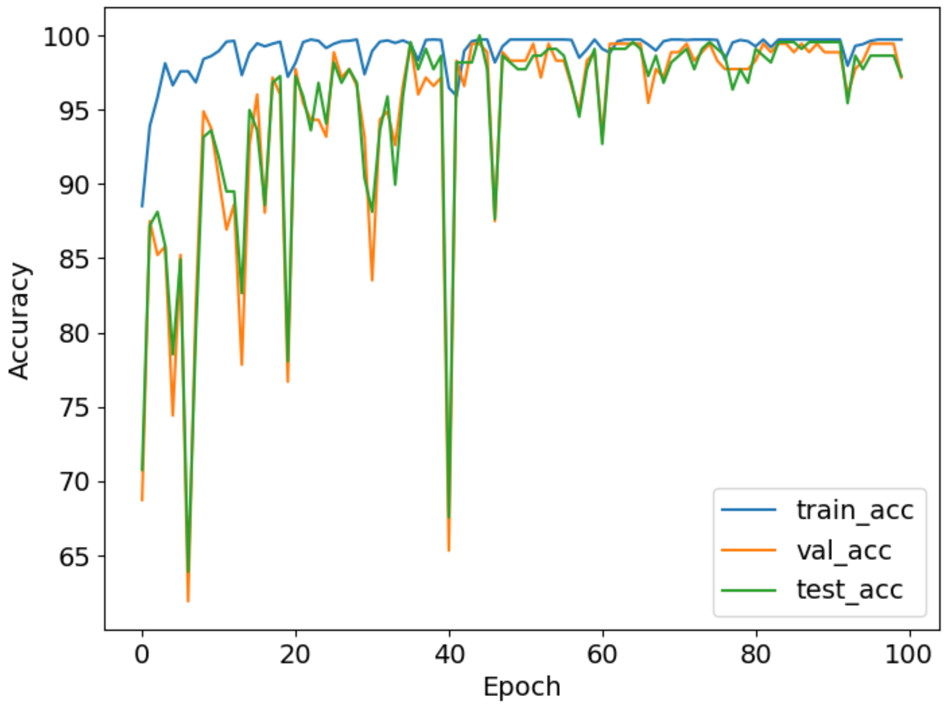


Figure 6.23: The training, validation, and test loss graph of VGG19 for HOG

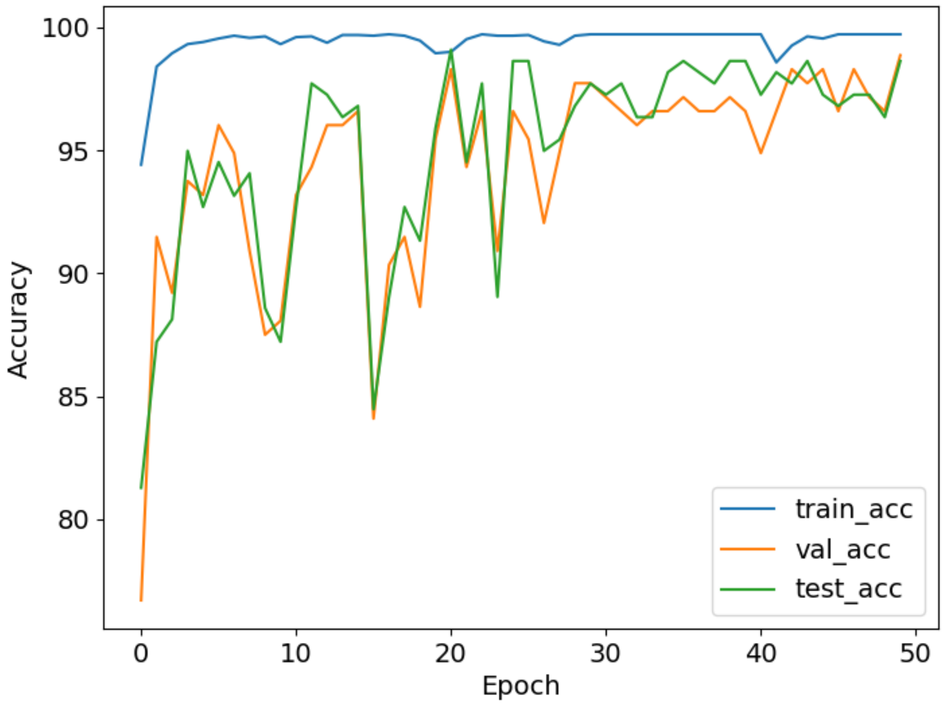


Figure 6.24: The training, validation, and test accuracy graph of MobileNetV2 for HOG

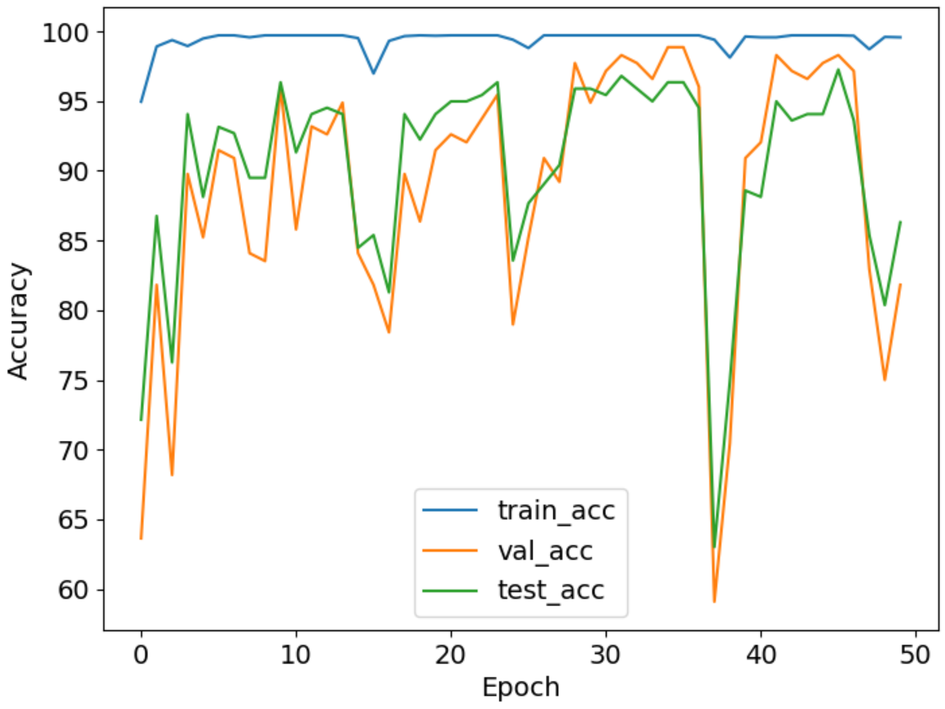


Figure 6.25: The training, validation, and test loss graph of DenseNet161 for HOG

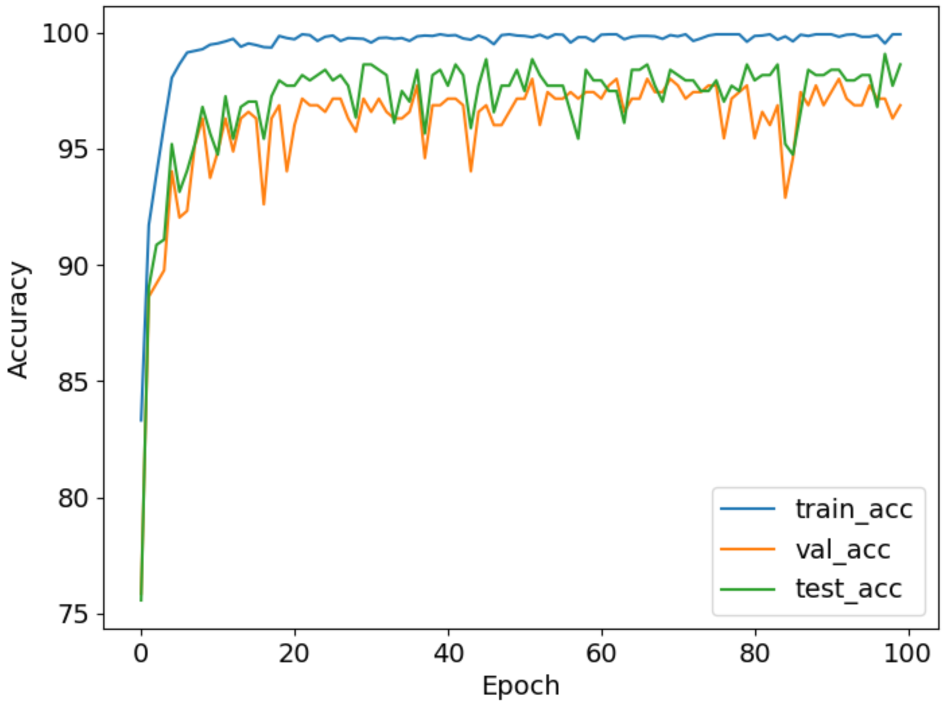


Figure 6.26: The training, validation, and test accuracy graph of ShuffleNet for HOG

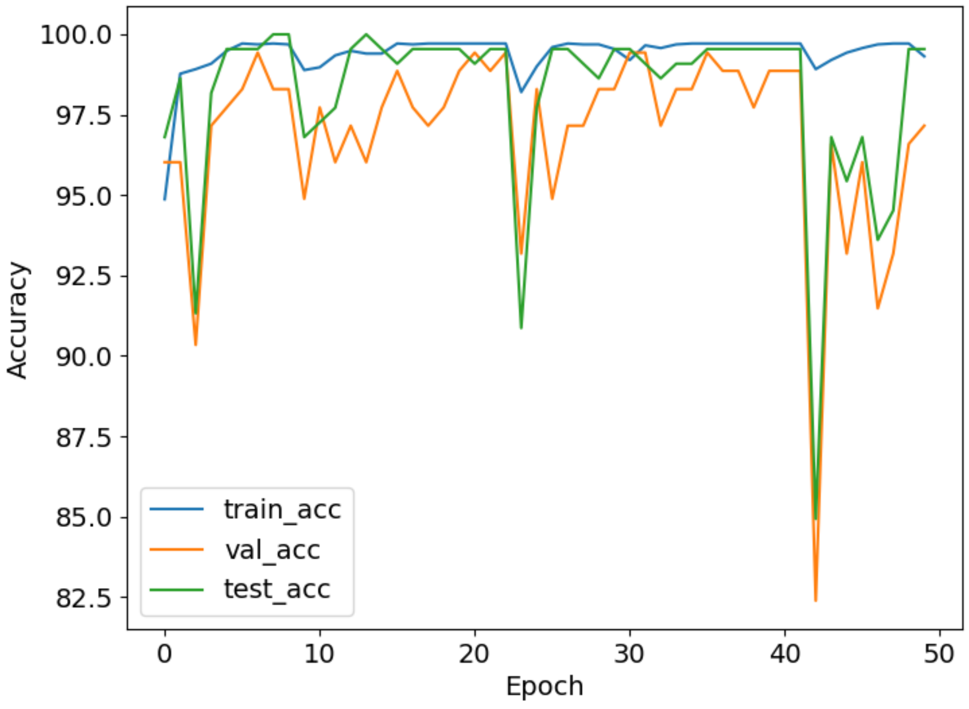


Figure 6.27: The training, validation, and test loss graph of InceptionV3 for HOG

Thus from the graphs, we perceived that in the case of ResNet18, we perceived that the CNN accuracy is 99.54% and for Hog it is 95.89%, showing that the CNN model is comparatively better. Similarly, the MobileNet_V2 gives 96.35% for CNN and 97.72% for Hog, the DenseNet161 has an accuracy of 99.09% for CNN and 96.35% for Hog, and finally, Shufflenet gives an accuracy of 98.17% for CNN and 97.95% for Hog. Thus, they all give better overall accuracy for the CNN classifiers rather than the Hog extractions. However, Inception_V3 gave the same accuracies for both CNN and Hog. i.e. 98.17%. Nevertheless, VGG19 was completely different in this case. It gave a far better result for Hog than the CNN classifier.

6.5 Discussion

Our result concludes an effective augmentation technique and a combination of CNN architectures as an effective classification technique for datasets with limited training samples. For the sake of our research, we have utilized the lung cancer dataset from The Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases which was collected in the above-mentioned specialist hospitals over a period of three months in fall 2019. The idea was to take the images, 1100 in total provided in the dataset as reference for the model we will train to recognize and distinguish between malignant, benign and normal and healthy lung CT scan images. Meanwhile, Cross-validation comes up as an excellent method to create machine learning applications with greater accuracy or performance. The test split is not sacrificed when using cross-validation methods like k-fold cross-validation to estimate a model's performance. Thus, cross-validation refers to the process of determining how accurate the model is by testing it on a number of distinct and separate subsets of data. As a result, can certainly generalize satisfactorily to the data that will continue to collect. The accuracy of the model is enhanced as a result.

There are 120 images with benign cases, 561 with malignant cases, and 416 with normal or healthy lung images. For our research purposes, this amount was not enough and hence we had to augment these images. We have implemented some renowned versions of architectures ('squeezenet1_0', 'vgg16', 'vgg19', 'resnet18', 'resnet50', 'resnet101', 'resnet152', 'inception_v3', 'inceptionresnetv2', 'xception', 'chexnet', 'nasnetalarge', 'densenet121', 'densenet161', 'densenet201', 'shufflenet' , 'googlenet', 'mobilenet_v2', 'alexnet') and out of those we have focused on the 'resnet18', 'vgg19', 'shufflenet', 'inception_v3', 'densenet161' and 'mobilenet_v2'. Comparatively, we examined that 'resnet18' came up with the best result. It is evident from the comparison table that 'densenet161' gave a better result than 'densenet121'.

The reason behind such reliable performance of ResNet is that it can overcome the vanishing gradient problem. When the layers of a neural network are too deep, the gradients while updating, due to the chain rule of gradient multiplication, shrink to zero. As a result, the weights stop updating and the layer stops learning anything during subsequent gradient updates. But in ResNet, gradients flow directly through skip connections. So we develop deeper networks with ResNet. Also, the ResNet down sampling is achieved by increasing the filter size rather than applying Max-pooling. By applying a larger filter, the network learns features from images on a larger area without losing any information. From the accuracy and training curve, we can observe that the loss value on the training and validation set is decreasing, which implies that the model is learning and minimizing its cost. Also, training and validation accuracy is increasing gradually, which means the performance is improving at each iteration.

t-SNE plot is a method to reduce dimensionality and visualize high dimensional features extracted from images by the classifier model into a 2D graph. Observing the t-SNE plot we can see that our developed classifier model extracts discriminating features from normal and malignant case images, but there are some overlapping between normal and benign cases. That is why some benign cases are misclassified as normal cases. From the accuracy and training curve, we can observe that the loss value on the training and validation set is decreasing, which implies that the model is learning and minimizing its cost. Also, training and validation accuracy is increasing gradually, which means the performance is improving at each iteration. This shows that all the architectures perform greatly with their novelty modules that are designed for high accuracy while maintaining a good number of parameters. Meanwhile, we get a better result in most of the cases for CNN classifiers rather than the Hog Feature extraction.

Chapter 7

Conclusion and Future Works

Cancer has an enormous frequency and death rate around the world. However, reliable and consistent information isn't accessible because it lacks records. Lung Cancer has the most elevated death rate, making it more essential to investigate the accessible information possibly it is inadequate. Analyzing such information is one of the most challenging errands in Ensemble Learning, and a practical algorithm ought to be chosen to perform it. We inspect the machine learning techniques, their applications in medical care, and malignant growth visualization and location. Specialists in the past created different mechanized devices for their initial identification. The discoveries uncovered that the Reconstruction ICA(RICA) and inadequate channel-based elements utilizing Support Vector Machine(SVM) Radial Basis Function kernel(RBF), polynomial, and Naïve Bayes gave the most elevated identification execution. As per our paper, among the 3 architectures we implemented, ResNet18 is bet amongst the three.

7.1 Future Works

Future endeavours in this exploration field ought to provide more measurable proof to validate the hypothetical framework. To finish up, profound learning and, as a rule, Deep learning has the chance to help pathologists and doctors by working on the productivity of their work, normalizing quality, and giving better visualization. We hope to further visualize and explore the possibilities of Deep Learning in the medical field and especially implementations of this sorts and optimistically the researchers relevant to these fields will conduct research further and provide validity to our research.

Bibliography

[1] S. A. Dwivedi, R. Borse, and A. M. Yametkar, "Lung cancer detection and classification by using machine learning & multinomial bayesian," *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, vol. 9, no. 1, pp. 69–75, 2014.

[2] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[3] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016.

[4] G. Kang, K. Liu, B. Hou, and N. Zhang, "3d multi-view convolutional neural networks for lung nodule classification," *PloS one*, vol. 12, no. 11, e0188290, 2017.

[5] W. Rahane, H. Dalvi, Y. Magar, A. Kalane, and S. Jondhale, "Lung cancer detection using image processing and machine learning healthcare," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, IEEE, 2018, pp. 1–5.

[6] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[7] A. Adhikari, *Skin Cancer Detection Using Generative Adversarial Networkand an Ensemble of Deep Convolutional Neural Networks.* The University of Toledo, 2019.

[8] M. Coccia, "Deep learning technology for detection of lung and breast cancer: Application in clinical practice," 2019.

[9] C. Dev, K. Kumar, A. Palathil, T. Anjali, and V. Panicker, "Machine learning based approach for detection of lung cancer in dicom ct image," in *Ambient communications and computer systems*, Springer, 2019, pp. 161–173.

[10] J. Fu, "Application of modified inception-resnet and condensenet in lung nodule classification," in *3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)*, Atlantis Press, 2019, pp. 186–194.

[11] L. Hussain, S. Rathore, A. A. Abbasi, and S. Saeed, "Automated lung cancer detection based on multimodal features extracting strategy using machine learning techniques," in *Medical Imaging 2019: Physics of Medical Imaging*, SPIE, vol. 10948, 2019, pp. 919–925.

[12] B. Madan, A. Panchal, and D. Chavan, "Lung cancer detection using deep learning," in *2nd International Conference on Advances in Science & Technology (ICAST)*, 2019.

[13] I. M. Nasser and S. S. Abu-Naser, "Lung cancer detection using artificial neural network," *International Journal of Engineering and Information Systems (IJEAIS)*, vol. 3, no. 3, pp. 17–23, 2019.

[14] P. Radhika, R. A. Nair, and G. Veena, "A comparative study of lung cancer detection using machine learning algorithms," in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, 2019, pp. 1–4.

[15] P. Radhika, R. A. Nair, and G. Veena, "A comparative study of lung cancer detection using machine learning algorithms," in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, 2019, pp. 1–4.

[16] N. Radwan, "Leveraging sparse and dense features for reliable state estimation in urban environments," Ph.D. dissertation, University of Freiburg, Freiburg im Breisgau, Germany, 2019.

[17] N. Aburaed, A. Panthakkan, M. Al-Saad, S. A. Amin, and W. Mansoor, "Deep convolutional neural network (dcnn) for skin cancer classification," in *2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, IEEE, 2020, pp. 1–4.

[18] H. F. Kareem, "The iq-oth/nccd lung cancer dataset," in *The IQ-OTH/NCCD lung cancer dataset*, The Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases, vol. 1098, 2020.

[19] H. K. Kondaveeti and P. Edupuganti, "Skin cancer classification using transfer learning," in *2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI)*, IEEE, 2020, pp. 1–4.

[20] F. Ramzan, M. U. G. Khan, A. Rehmat, *et al.*, "A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks," *Journal of medical systems*, vol. 44, no. 2, pp. 1–16, 2020.

[21] S. S. Raoof, M. A. Jabbar, and S. A. Fathima, "Lung cancer prediction using machine learning: A comprehensive approach," in *2020 2nd International conference on innovative mechanisms for industry applications (ICIMIA)*, IEEE, 2020, pp. 108–115.

[22] N. Rezaoana, M. S. Hossain, and K. Andersson, "Detection and classification of skin cancer by using a parallel cnn model," in *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, IEEE, 2020, pp. 380–386.

[23] G. Singhal, "Introduction to densenet with tensorflow," *Pluralsight. com.[Online]. Available: https://www. pluralsight. com/guides/introduction-to-densenet-with-tensorflow.[Accessed: 02-Dec*, 2020.

[24] K. Tuncal, B. Sekeroglu, and C. Ozkan, "Lung cancer incidence prediction using machine learning algorithms," *Journal of Advances in Information Technology Vol*, vol. 11, no. 2, 2020.

[25] S. Abd ElGhany, M. Ramadan Ibraheem, M. Alruwaili, and M. Elmogy, "Diagnosis of various skin cancer lesions based on fine-tuned resnet50 deep network," *Comput. Mater. Continua*, vol. 68, pp. 117–135, 2021.

[26] D. M. Abdullah, A. M. Abdulazeez, and A. B. Sallow, "Lung cancer prediction and classification based on correlation selection method using machine learning techniques," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 141–149, 2021.

[27] M. S. Ali, M. S. Miah, J. Haque, M. M. Rahman, and M. K. Islam, "An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models," *Machine Learning with Applications*, vol. 5, p. 100 036, 2021.

[28] T. A. M. Devi and V. M. Jose, "Three stream network model for lung cancer classification in the ct images," *Open Computer Science*, vol. 11, no. 1, pp. 251–261, 2021.

[29] M. Dildar, S. Akram, M. Irfan, *et al.*, "Skin cancer detection: A review using deep learning techniques," *International journal of environmental research and public health*, vol. 18, no. 10, p. 5479, 2021.

[30] S. N. Ghorpade, M. Zennaro, and B. S. Chaudhari, "Iot-based hybrid optimized fuzzy threshold elm model for localization of elderly persons," *Expert Systems with Applications*, vol. 184, p. 115 500, 2021.

[31] O. Obulesu, S. Kallam, G. Dhiman, *et al.*, "Adaptive diagnosis of lung cancer by deep learning classification using wilcoxon gain and generator," *Journal of Healthcare Engineering*, vol. 2021, 2021.

[32] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm," *Sensors*, vol. 21, no. 8, p. 2852, 2021.

[33] R. Sujitha and V. Seenivasagam, "Classification of lung cancer stages with machine learning over big data healthcare framework," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 5, pp. 5639–5649, 2021.

[34] K. M. M. Haider, M. Dhar, F. Akter, S. Islam, S. R. Shariar, and M. I. Hossain, "An enhanced cnn model for classifying skin cancer," in *Proceedings of the 2nd International Conference on Computing Advancements*, 2022, pp. 456–459.