Deep Learning based Early Glaucoma Detection

by

Aabrar Islam
20101361
Ayen Aziza Haque
20301487
Najifa Tasnim
20101406
Simin Waliza
20101401

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
January 2024

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Aabrar Islam	Ayen Aziza Haque
20101361	20301487
Najifa Tasnim	Simin Waliza
20101406	20101401

Approval

The thesis titled "Deep Learning based Early Glaucoma Detection" submitted by

1. Aabrar Islam (20101361)
2. Ayen Aziza Haque (20301487)
3. Najifa Tasnim (20101406)
4. Simin Waliza (20101401)

of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 22, 2024.

Examining Committee:

Supervisor:
(Member)

Dewan Ziaul Karim

Lecturer
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Program Coordinator:
(Member)

Dr. Golam Rabiul Alam

Associate Professor
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Abstract

Glaucoma is a severe eye condition that can lead to progressive vision impairment if left untreated. Diagnosis and monitoring of glaucoma at an initial stage is critical for effective treatment of the disease. However, the diagnosis is complex with bare eyes which requires multiple checkups and tests. Image processing is important for diagnosing glaucoma by providing valuable information about the intricate structure of the eye which helps improve the accuracy of diagnosis and allows for earlier detection of the disease. This portrays the requirement for further research in this field. We aim to explore various image-processing models used for image classification and develop an efficient model that can be used in the detection of glaucoma. In this paper, we have proposed a Custom CNN model with 22 layers based on deep learning for glaucoma diagnosis where it detects from the fundus images whether the person has glaucoma or not. The model has been trained using datasets containing 4000 fundus images each with 2 categories which are Glaucoma and Non-Glaucoma. The datasets have been used on the Custom CNN model and six other pre-trained models. Our proposed model has been able to successfully classify the images with an accuracy of 98.71% which was the highest among all the models despite having a lower number of parameters compared to the other models.

Keywords: Fundus Image, Glaucoma diagnosis, Deep Learning, Convolutional Neural Network (CNN).

Dedication

This paper is dedicated to our families and fellow team members. The constant support of the family members and the determination of the team members played a great role in making this paper a success. We could not have finished our thesis without the help of our esteemed supervisor who has been a continual source of guidance and advice. We also dedicate this paper to him.

Acknowledgement

First and foremost, all thanks and praises to the Almighty for the blessings throughout our research without which the thesis would not be successful. We would like to sincerely thank our supervisor Dewan Ziaul Karim sir for his help and guidance. Finally, we want to express our gratitude to our parents and well-wishers who helped us throughout this project and all of the faculty members and students for creating such a humble learning atmosphere in which we could grow personally and do this research successfully.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$AGIS$ Advanced Glaucoma Intervention Study

AI Artificial Intelligence

AP Average Precision

AUC Area Under Curve

BMI Body Mass Index

CAD Computer Aided Diagnosis

CDR Cup-to-Disk Ratio

CNN Convolutional Neural Network

CPF Color Fundus Photography

$DCNN$ DenseNet-201 Deep Convolution Neural Network

DL Deep Learning

FVC Forced Vital Capacity

$GCIPL$ Ganglion Cell Inner Plexiform Layer

$GSS2$ Glaucoma Staging System 2

$HCDR$ Horizontal Cup-to-Disc Ratio

KNN k-Nearest Neighbour

MD Mean Deviation

ML Machine Learning

NN Neural Network

OCT Optical Coherence Tomography

ONH Optic Nerve Head

PEF Peak Expiratory Flow

RF Random Forest

$RNFL$ Retinal Nerve Fiber Layer

SLV Standardized Loss Variance

SVM Support Vector Machine

$VCDR$ Vertical Cup-to-Disc Ratio

VF Visual Field

VGG Visual Geometry Group

Chapter 1

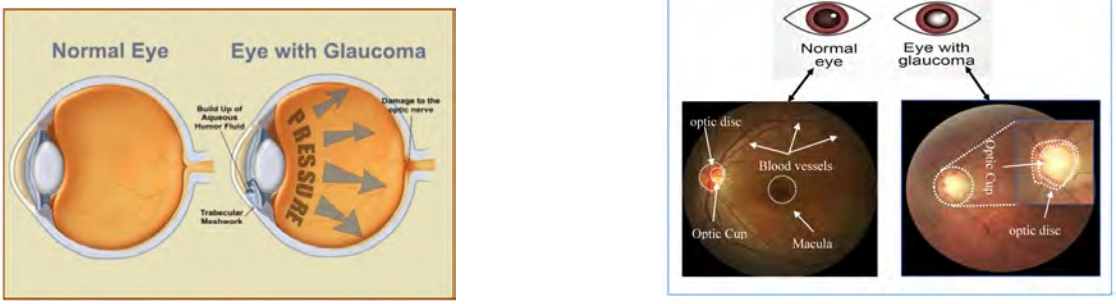
Introduction

Glaucoma is severe, potentially blinding and an irreversible eye disease. It is estimated that more than 50 million people are affected to this date and predicted to affect more than 100 million by 2040. Glaucoma is a major eye condition in which fluid is built up in the eye and is pressed against the retina and the optic nerves. As the disease progresses towards the next stage, defects start to develop in the visual field and if not controlled can result in complete blindness.

Timely detection and predicting the progression of rapid treatment is still one of the major challenges in medical history. The early investigation of the optical disc changes is one of the foremost and necessary steps for later diagnosis. Firstly, the clinical exams and tests are crucial to be performed with a substantial amount of proficiency, for example, optic nerve imaging and visual field testing [14]. But, according to the statistics, the required amount of expertise in performing the tests is inadequate. As a result, the number of patients is increasing at an alarming rate. Secondly, the symptoms of glaucoma are imperceptible until the later stages of the disease. Early intervention necessitates the utilization of novel diagnostic instruments to promptly identify the disease.

As the disease is progressive and irreversible, early detection and diagnosis are of immense importance. Although the disease is asymptomatic in the very early stages, there are some specific functionalities in the eye that can be taken into consideration and tested. The part of our eye that processes the vision we are seeing directly in the front is the macula. The initial signs of vision impairment can potentially be identified by observing the preceding alterations in the macula and the Retinal nerve fiber layer (RNFL). [14]. Most of the studies have been inclined towards color fundus photography (CPF) as it is widely available. On the other hand, many reports have utilized optical coherence tomography (OCT), a procedure where infrared lights are used to generate a 3D image of the retina and visual tests. The advantage of OCT is the ultrasound imaging in processing speed because light travels faster than sound. Furthermore, research has employed the technique of segmenting the optic cup and optic disk from the previously mentioned color fundus photographs. This approach allows for the determination of the cup-to-disk ratio (CDR) as a means to identify indications of glaucoma.

Deep learning is a major research topic in terms of understanding discriminatory representations of the data. The motive of the architecture is to yield more abstract and useful representations by combining various linear and nonlinear operations applied to the dataset. One of the architectures of Deep learning is Convolutional Neural Networks (CNN) which has been proven to have done successful image segmentation and classification. Here, in this paper, we have developed a Custom CNN model that identifies images with or without Glaucoma using binary classification. This model has achieved an accuracy of 98.71% and has high precision, recall, f1 score and accuracy.



(a) Basic explanation

(b) Fundus Image explanation

Figure 1.1: Glaucoma Visualization

1.1 Problem Statement

1. **Laborious and Time-Consuming Process:**
   Glaucoma is one severe disease that results in permanent blindness if goes unchecked in the early stages. But the manual procedure takes too long. The minimum number of check-ups needed is five times for the eye specialist [15]. They have to go through step-by-step diagnosis such as Tonometry, Optical Coherence Tomography, Ophthalmoscopy, Perimetry, and Gonioscopy to analyze a glaucoma-infected eye. These various tests check for eye pressure, any damage to retinal nerve fiber layers, the optic nerve, vision loss, and the cause of high eye pressure one by one. All these are works of lengthy processes that are bothersome for both the doctor and the patient. Here comes the need to construct a system where Glaucoma can be detected in one single attempt instead of these lengthy 5 consecutive check-up reports.

2. **Hardly Identifiable Patterns:**
   Compared to any diagnosis, searching for Glaucoma indications in retinal images is inconspicuous. Through hierarchical feature learning, CNN can capture anything from simple elements to increasingly complex patterns. This is essential to detect the delicate patterns from the fundus images for Glaucoma because these structures can be too intricate to identify by traditional approaches.

3. **Imprecise Localized Feature Extraction:**
   To detect Glaucoma, cup-disk features are examined, and the cup-disk ratio is measured from optic nerve head (ONH) images. With the help of convolutional and pooling layers, CNN can extract those features from the local region pretty accurately which gives us ample opportunities to work with any detail we might require.

4. **Unavailability of Authentic Dataset:**
   The lack of comprehensive and publicly accessible retinal fundus image datasets has posed a significant obstacle to the widespread application of AI in practical Computer-Aided Diagnosis(CAD) to detect glaucoma. The few available datasets, primarily intended for research purposes, often face strict inclusion criteria along with unrealistic image-capturing conditions. [3]

In this research, we are striving to apply deep learning algorithms, focusing mostly on Convolutional Neural Networks to examine the three datasets; REFUGE, ORIGA and G1020. Our primary objective is to predict the early onset of glaucoma by focusing on the optic disc and cup region in fundus images.

1.2 Research Objective

The goals we want to achieve through our research:

- To develop a clinically feasible deep learning based approach to detect glaucoma at the early stage to avoid further vision loss.

- To make glaucoma diagnosis time-efficient and cost-effective by avoiding several attempts and clinical tests.

There have been promising attempts similar to the standards we have set in the Machine Learning sector and Deep Learning. Esteemed researchers have directed their attention toward utilizing fundus images for the early detection of glaucoma. Despite the disease being asymptomatic, its detection is challenging but not insurmountable. As a result, we are focusing on using CNN for its noteworthy ability to analyze the intricate details and structures of the pictures and provide high accuracy.

Chapter 2

Literature Review

Since Glaucoma is a severe eye condition leading to major vision loss many eye specialists and researchers have been working on it, trying to find a way to detect Glaucoma in the early stages otherwise it can get worse to the extent of losing complete vision. Many image processing techniques have been developed. An automatic image processing system was used in this paper [2], calculating the vertical and horizontal cup-to-disc ratios which are VCDR and HCDR. Two algorithms were applied to carry out the experiment where level set and inpainting methods were made to carry out the disc segmentation and on the other hand, segmentation of the cup was carried out using a Type-II fuzzy approach. The images on which the algorithms were used were manually marked initially and the results were later verified compared with the manual results. Still, The combined accuracy of HCDR and VCDR algorithms was 74.2%.

However, in comparison to ophthalmologists and conventional methods, CNN had achieved higher accuracy in detecting Glaucoma [13]. The authors used a valid set of visual fields numbering 300 where CNN achieved an accuracy of 0.876 which was the highest. They even got an accuracy result for ophthalmologists, where resident ophthalmologists had 0.607, attending ophthalmologists showed 0.585 and glaucoma experts got 0.626 of the accuracy. GSS2 and AGIS accomplished a precision of 0.523 and 0.459. Even the standard machine learning algorithms like k-nearest Neighbour (kNN), Support Vector Machine (SVM) and Random Forest (RF) could not keep up in this experiment achieving an average of 0.6 accuracy.

The study [10] using Optical Coherence Tomography (OCT) also had a high accuracy in detecting Glaucoma with deep learning models. They used a training set of 7288 OCT images and applied the (VGG-19) CNN model which is known as the Visual Geometry Group. Distinctive OCT maps were assessed and finally compared with the results from two Glaucoma specialists. The thickness maps and deviation of Retinal Nerve Fiber Layer (RNFL) and Ganglion Cell Inner Plexiform Layer (GCIPL) analyses were mainly focused on in the research. In all, their deep learning model scored a high accuracy rate in detecting Glaucoma showing similar detection patterns as Glaucoma specialists.

In an early paper of 2015 [4], ALADDIN, an automated feature learning model was proposed for glaucoma detection that involved the application of deep learning and

convolutional neural networks (CNN) for feature learning. Instead of using a regular linear layer for convolutional layers, they tried to implement a multilayer perceptron using micro neural networks with a higher complexity structure. Additionally, their paper suggested a deep learning architecture for analyzing fundus images and recognizing glaucoma patterns by employing a hierarchical representation approach. They did achieve an accuracy score of 0.838 and 0.898 on the datasets of ORIGA and SCES according to the result of Area Under Curve (AUC).

To overcome the limitations of current machine learning (ML) approaches with features like the high segmentation errors in retinal thickness maps, a paper has been proposed [14] where they have been able to take a more accurate Machine learning based approach for glaucoma visualization. They have effectively compiled vast datasets comprising information on population characteristics, systemic factors, and eye-related parameters, in addition to color fundus photos (CFPs) and macular Optical Coherence Tomography (OCT) scans. They have built two Deep Learning models with the imaging modality of both retinal OCT and CPFs and respectively named the Retinal OCT model and CPF model which input an image and output the probability of the presence of glaucoma in the input image. Furthermore, they built three models using gradient-boosted decision trees based on ocular features, demographic features: age, gender, and ethnicity; systemic features: Forced Vital Capacity (FVC), Body Mass Index (BMI), Peak Expiratory Flow (PEF), heart rate, diastolic and systolic blood pressure, presence of diabetes, recent caffeine and nicotine intake. Significantly, they have validated the model's accuracy by comparing its results with the clinical interpretation of color fundus photos (CFPs) by healthcare professionals.

The paper [12] suggested a clinically practical deep-learning system to predict glaucoma with the help of solar fundus photographs (CPF). They have developed three models. The first model has been developed as a diagnostic algorithm for possible glaucoma, DiagnoseNet. The second model has been developed to predict the future depending on the 3 longitudinal CFP images or visual field cohorts data named PredictNet. Lastly, the third was developed to predict the progress of glaucoma from cohorts with existing glaucoma. The accuracy of predicting glaucoma incidence is 90% and glaucoma progression is 91%.

A futuristic method to differentiate between healthy and early-glaucomatous VFs using CNNs has been presented in the paper [11]. They have introduced the concept of Voronoi images, which is a technique to convert visual fields (VFs) into 2D images, overcoming the spatial distribution challenges posed by the test locations within the 30-degree visual field. VFs can be classified with the help of Voronoi images by custom-designing CNN. Two additional maps have been provided here which highlight the VF regions contributing to the CNN decision. In the later step, the effectiveness and verification of the method have been tested. Then, a comparison of the method with a Neural Network (NN) containing no spatial information and those based on MD, SLV and both were investigated. The first dataset gave the best AP score in CNN (0.874 ± 0.095) across all test folds and the third best for the other.

This paper [1] builds a deep-learning model that uses fundus photography to diagnose glaucoma. A total of 1,542 fundus photos (*467 advanced glaucoma, 786 normal controls, and 289 early glaucoma patients*). In the dataset of 1,542 images, there were 464 from the test, 754 from the training and 324 validation datasets. Using the collected datasets, researchers have developed a straightforward logistic classification and Convolutional Neural Network (CNN) using TensorFlow. They have also applied fine-tuning techniques to pre-trained GoogleNet Inception v3 models for enhanced performance. The model showed an 82.9% training accuracy, 79.9% validation accuracy and 77.2% test accuracy. Whereas the convolutional neural networks achieved and accuracy of 92.2%.

A more innovative and safe approach for predicting glaucoma before any symptoms appear is proposed in the paper [9]. The glaucoma images have been analyzed using deep learning where they segment the optic cup that has been pre-trained and model integrated with the U-Net architecture. The DCNN (*DenseNet-201 deep convolutional neural network*) methodology is employed to ascertain the presence of glaucoma in individuals. During the training and testing process, the model achieves an overall accuracy of 90%.

In this paper [17] detecting glaucoma which is feature-based in the retinal fundus, many types of features have been used to form an enhanced deep image analysis model. The features used have parameters which are nasal, inferior, superior, cup-to-disc ratio, and temporal region area and these are combined to form the overall model. This model is composed of four different types of ML algorithms which include individual algorithms like Naive Bayes, SVM, and KNN. There are two classes for the images that are healthy and have glaucoma and the model has been partitioned into three parts which are classification, pre-processing, and deep image feature extraction. The dataset used for this research is public and popular as real-time scenarios are required for the research. The name of the dataset is DRIONS-DB and has 140 images where almost 50% of the images have glaucoma and the rest of the images do not have glaucoma.

In this paper [7], Early Glaucoma Detection is an essential part of diagnosing glaucoma so to efficiently diagnose and treat glaucoma the use of image processing has been prioritized in a wider range. For this research, a unique ensemble-based deep learning model has been used where three pre-trained CNN models were used for detecting glaucoma. The three networks used have different accuracy levels and five different data sets were used to analyze the algorithm. The three networks used were (VGGNet) which is the visual geometry group network, GoogLeNet and (ResNet) which is a residual network. Among the five datasets, the dataset PSGIMSR had an accuracy of 91.11%, a specificity of 95.20% and a sensitivity of 85.55%. Moreover, other datasets including DRIONS-DB, DRISHTI-GS and HRF had different accuracies. Lastly, the combined data sets had an accuracy of 88.96%. As image processing is capable of handling mass data of patients and analyzing, images produced by OCT have been used as it is easier to produce 3D images of the retina by joining several OCT images together.

The paper [5] states that glaucoma is irreversible and can deteriorate the quality

of life. Thus, a deep learning architecture is developed to automatically diagnose glaucoma based on the convolutional neural network. This model can capture and characterize the hidden patterns that develop due to glaucoma. The paper works with two datasets and a total of six layers with weights. These can boost the performance and reduce the overfitting problem. Moreover, there is a preprocessing step that removes the bright fringe and the inputs for the proposed model take less time to process as the images are smaller. The paper [5] also claims the potential of working with multiple ocular diseases in the future.

Another paper [16] creates a generalized DL model that focuses on three different learning architectures namely ResNet50, GoogLeNet and ResNet-152. This model works with 5 different datasets rather than focusing on one or two and thus, is fine-tuned so that it works well for any dataset. For the images, there is also a preprocessing step to standardize the fundus images. It may show variability but 80% of the time it shows better performance overall than previous works. Considering accuracy, it is better 53% of the time. Furthermore, for AUC and specificity performance, the model is better 87% and 100% of the time respectively.

This paper [18] presents an automatic two-stage glaucoma detection system which will be more efficient and further lessen the workload of ophthalmologists. In this structure, DeepLabv3+ segments the region of the optic disc and then many deep CNNs substitute the module. Pretrained deep CNN has been used for learning transfer as well as using a support vector machine for feature descriptors and then creating the whole methods for the previous two proposals. Five datasets have been used to try the methods containing a total of 2787 images. After testing it showed that MobileNet and DeepLabv3+ combined worked best. This method has accuracies above 85% with each of the datasets than the methods normally used would have where it had the best performance with CUHKMED using the Refuge dataset.

Chapter 3

Methodology

We have successfully acquired the Glaucoma Fundus Imaging Datasets and are now prepared to initiate the preprocessing phase. Within this preprocessing stage, we have segregated our glaucoma dataset into three distinct categories, namely, training, testing, and validation. We have allocated them in a proportionate ratio of 70:20:10. During the training phase, 70% of our glaucoma dataset is meticulously utilized to instruct and fine-tune our deep learning models. 20% is employed for the testing phase and in the validation phase, we employ the final 10% which will play a crucial role in fine-tuning model hyper-parameters and ensuring that it can make reliable predictions on unseen data. Our prime motive is to build a custom CNN model that would obtain the highest accuracy in the detection of Early-stage Glaucoma. We are also focusing on using various deep learning models like InceptionNetV3, DenseNet169, EfficientnetB4, VGG16, Alexnet and Squeezenet in our research for transfer learning and to compare the better performance of prediction with our Custom CNN model. We will be thoroughly implementing each model to identify the top models with the best prediction accuracy.

3.1 Workflow of the Methodology

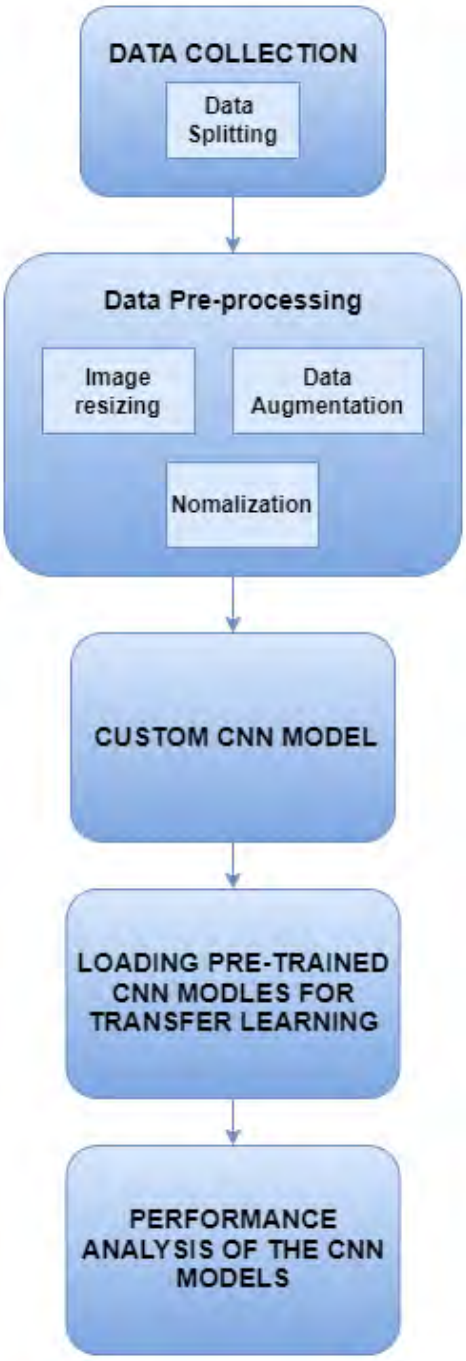


Figure 3.1: Workflow of the Methodology

3.2 Dataset Description

3.2.1 Dataset collection

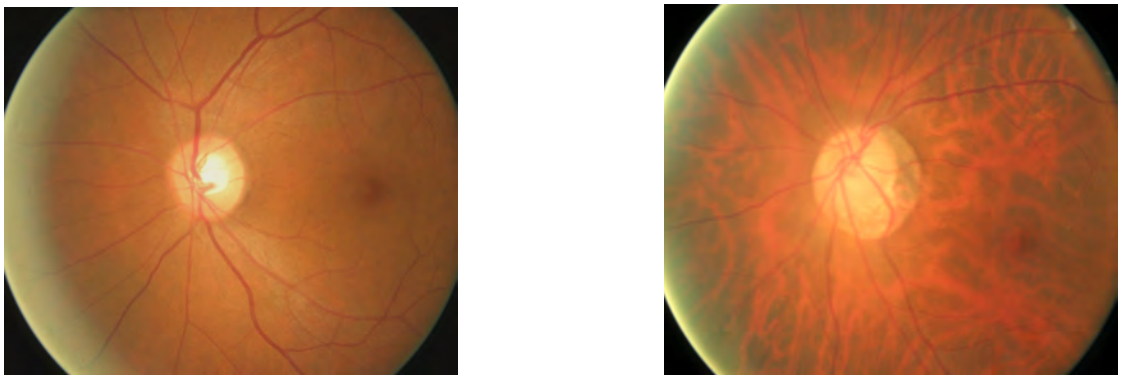
The presence of an optical disk cup image is crucial for identifying glaucoma because it enables the assessment of the cup-to-disk ratio (CDR). The CDR is a crucial factor in diagnosing glaucoma since it tends to be higher in individuals with the condition. [20]
Our dataset contains 4000 images of the human eye which have been altered into a special type of medical image called *fundus image* and have been cropped to focus only on the area containing the optic disc and cup within the eye.
We have categorized them into *Glaucoma* and *Non-Glaucoma* classes according to the diagnosis. We have further arranged the labels into groups of 1 and 0 respectively. Among these images, 2000 have been diagnosed with Glaucoma and 2000 images are normal.
For this study, we have collected the Glaucoma Fundus Imaging Datasets [8] containing the Fundus images and OD/OC masks from ORIGA, REFUGE, and G1020 datasets. From that, we have created two distinct datasets-

1. **Main Dataset (REFUGE):** Our primary dataset originates from the REFUGE dataset which has been later preprocessed and tailored for this paper.

2. **Secondary Experimental Dataset:** We have also worked on a secondary dataset for further comparison. All three sources (ORIGA, REFUGE, G1020) have been merged and preprocessed for this dataset.

3.2.2 Fundus Image Sample

Fundus images of a Glaucoma-infected eye and an uninfected eye are shown in the figures below, which we used to train our models.



(a) Glaucoma (b) Non-Glaucoma

Figure 3.2: Data Sample

In a healthy eye, the optic cup is a minor central dip in the optic disk where the optic nerve exits. In glaucoma's advance, the optic cup enlarges, encroaching on nearby nerve tissue, potentially leading to vision loss.

3.2.3 Data Labelling

We have collected our datasets from [8] and through information from corresponding JSON files, labeled the images in their respective classes. We have labeled our data in two classes implementing binary classifier. The classification has been done into a class of images diagnosed with Glaucoma and another class of images that are healthy.

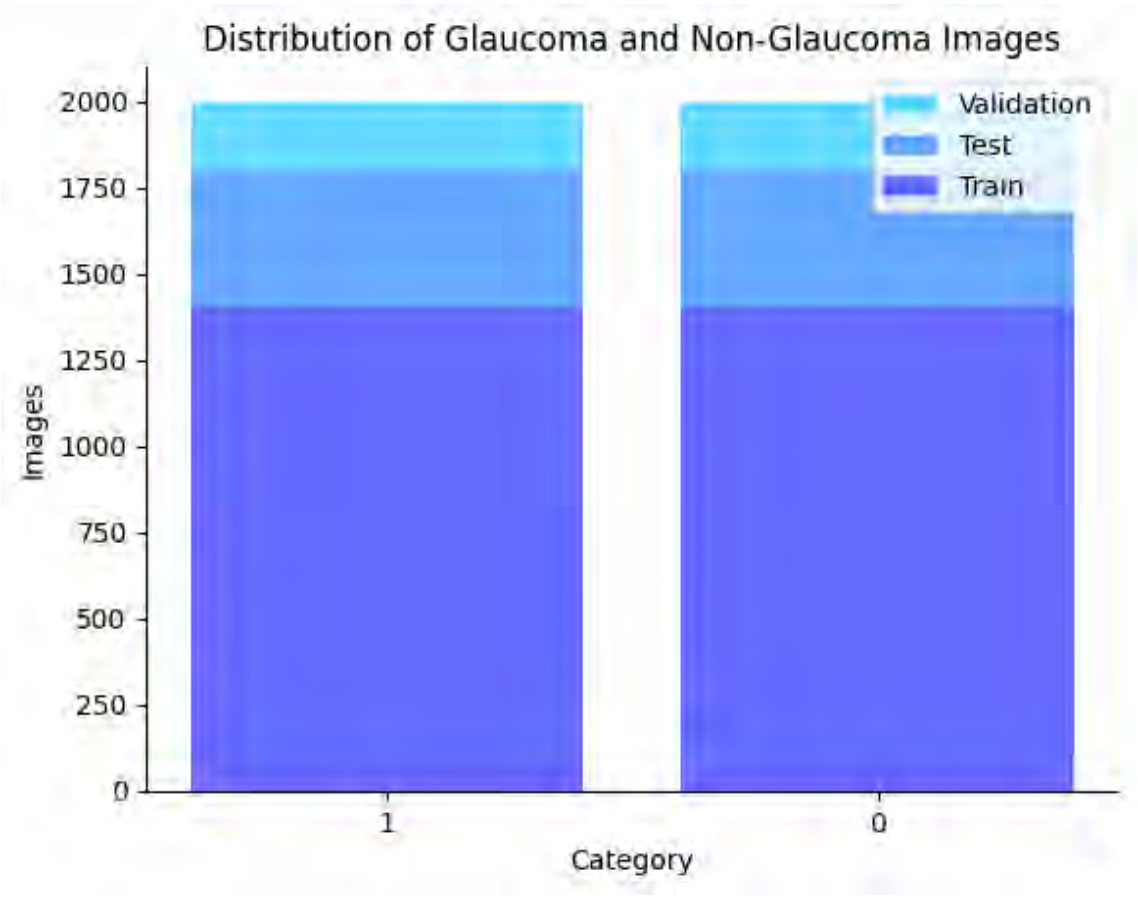


Figure 3.3: Bar-Chart distribution of data

According to the figure, we have categorized Glaucoma images as **1** and Non-Glaucoma images as **0** with the respective histogram. As we can see, the blue bar denotes the **train set**, the orange bar denotes the **test set** and lastly, the green bar denotes the **validation set**.

3.2.4 Dataset Classification

By our research methodology, we have allocated 70% of the glaucoma dataset for the training phase, while reserving 20% for rigorous testing and dedicating the remaining 10% for the crucial validation process. This partitioning strategy is employed to ensure the robustness and reliability of our experimental results.

Categories	**Train**	**Test**	**Validation**	**Total**
Glaucoma	1400	400	200	2000
Non-Glaucoma	1400	400	200	2000

Train Set: The training dataset is usually a large chunk of the dataset that is used to train the model to recognize subtle patterns and features. In this case, the dataset is used on fundus images linked to both healthy and glaucoma-labeled images. It allows the training data promote generalization, enabling the machine to predict new, unidentified fundus images improving the model's capacity to recognize glaucoma accurately,

Test Set: The test dataset is used to determine whether the built model works correctly or not. The model's capacity to generalize the information it has learned is assessed by testing if it can successfully diagnose glaucoma in actual situations on a different dataset. It ensures a trustworthy indication of the model's performance.

Validation Set: During the process of training the dataset, this helps in identifying the effectiveness of the model in learning new information independent of the training data. The performance on the validation set helps us decide the hyperparameter tuning, changing model architecture and also detecting over-fitting in the early stage.

3.3 Data Pre-processing

3.3.1 Data Augmentation

To expand our existing dataset, we have augmented images from the smaller datasets we had and created a combined balanced dataset. Furthermore, to expand our dataset, we used **contrast modification**, **flipping**, **left and right rotation**, **skewing**, and **zooming in and out**. This helped to train the models efficiently for better predictions and thus give better results. The images in the dataset have been cropped to focus on the core glaucoma part which is the area of interest.

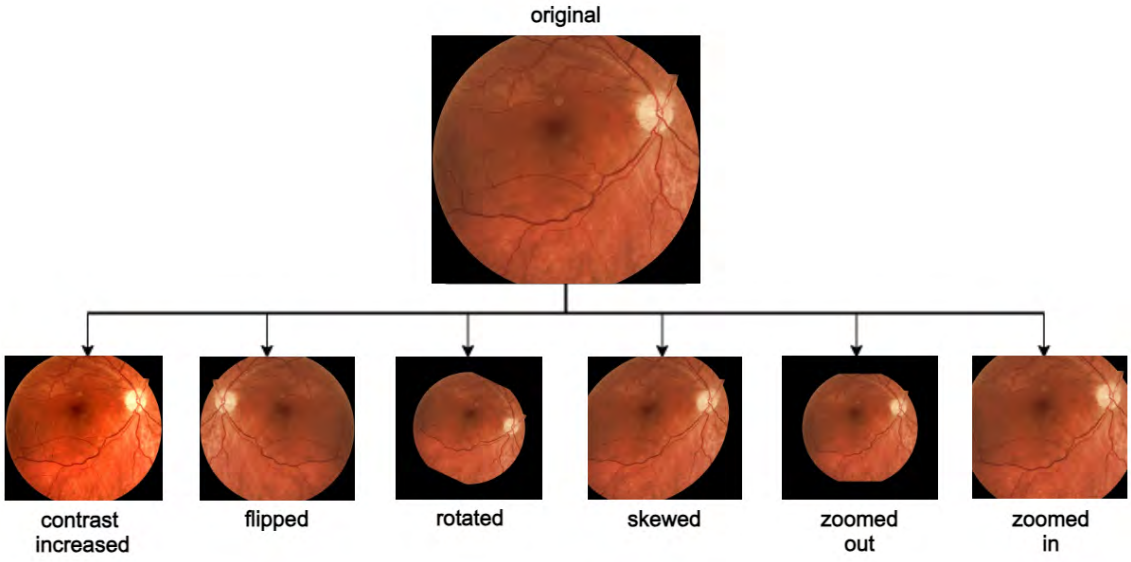


Figure 3.4: Before and After Data Augmentation

3.3.2 Resizing Image

To keep the dimensions of the images consistent the pixel sizes were mainly 224*224 but it was also suggested for the models we used. However, for the Custom CNN model, we kept the pixel sizes to 256*256 to ensure that the features of the images are reserved. Due to resizing, only the dimensions have changed but the aspect ratio remains the same. Changing the number of pixels enabled the model to train better as the new images had different size characteristics.

3.3.3 Normalization

The pixel values were normalized to a certain standardized range mainly between [0,1]. This ensured that the images were consistent throughout. Normalization helps the models to converge faster so that training is done efficiently as the data is scaled to a certain range. We have used min-max scaling for some models and standardization for a few other models to ensure better accuracy.

Chapter 4

Proposed Custom CNN Model

Our custom model defines a Convolutional Neural Network (CNN) for image classification using the TensorFlow and Keras library. The model is designed to classify fundus images to detect Glaucoma.

1. **Data Generators:** Data generators are employed to load and preprocess data in batches, which is especially useful for working with large image datasets. These generators use strategies for data augmentation to broaden the variety of data.

2. **Encoder-Decoder Architecture:** At the core of our model, we have an encoder-decoder architecture:

   - **Encoder:** The encoder consists of multiple convolutional layers, followed by max-pooling and batch normalization. It reduces spatial dimensions while increasing the depth (number of channels) to encode critical information.
   - **Decoder:** Mirroring the encoder, the decoder gradually upsamples the spatial dimensions while reducing the number of channels. Batch normalization is incorporated in each decoder block for enhanced stability.

3. **Convolutional Layers:**

   - **Block 1:** 1024 filters, 3×3 kernel
   - **Block 2:** 512 filters, 3×3 kernel
   - **Block 3:** 256 filters, 3×3 kernel
   - **Block 4:** 128 filters, 3×3 kernel
   - **Block 5:** 128 filters, 2×2 kernel
   - **Block 6:** 64 filters, 2×2 kernel
   - **Block 7:** 32 filters, 2×2 kernel
   - **Block 8:** 32 filters, 2×2 kernel

   ReLU (Rectified Linear Unit) **Activation** is utilized after each Convolutional layer. ReLU Activation Ensures non-linearity in the model, enabling it to learn complex patterns. **Batch Normalization** and **Max Pooling** are applied after each Conv2D layer for improved training stability and spatial dimension

reduction. Batch Normalization Improves training efficiency and stability by normalizing the inputs of each layer. Max Pooling Reduces spatial dimensions, aiding in reducing overfitting and computational load. Finally, **Dense Layers** have been implied which helps capture high-level features and patterns from the extracted spatial information.

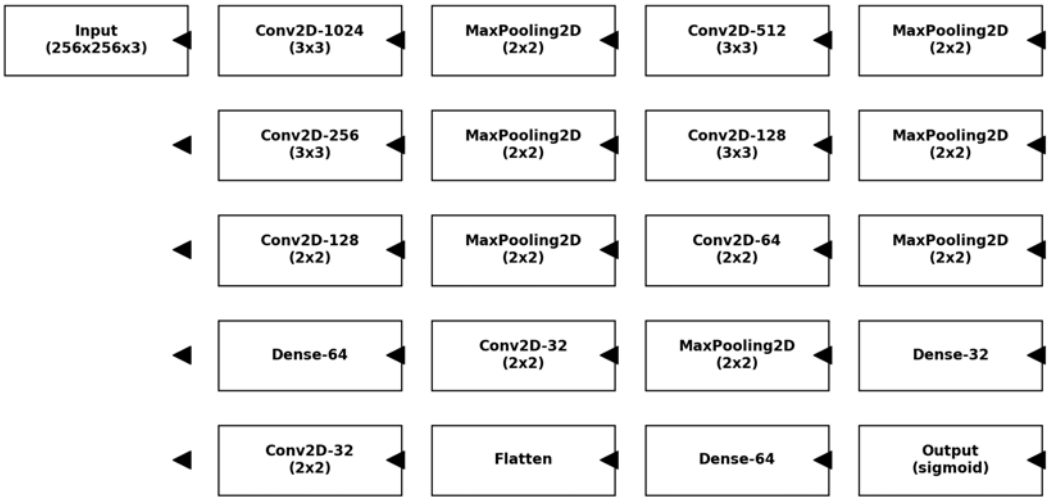


Figure 4.1: Structure of Custom CNN model

4. **Final Layers and Output:** The model concludes with a flatten layer to transition from convolutional to dense layers, followed by a dense layer of 64 neurons with ReLU activation and another dense layer with a dropout set at 0.3. The final output layer uses a single neuron with sigmoid activation for binary classification. The formula of sigmoid is

$$1/(1 + e^{-x})$$

5. **Compilation and Optimization:** The model concludes with a flattened layer to transition from convolutional to dense layers, followed by a dense layer of 64 neurons with ReLU activation.

   - **Loss Function:** Binary Cross entropy is chosen for its suitability in binary classification tasks, particularly in medical diagnostics where probability scores are crucial.
   - **Optimizer:** The Adam optimizer, with a learning rate of 0.001, is selected for its efficiency in handling sparse gradients and adaptive learning rate capabilities, essential in medical image analysis.

6. **Training Configuration:** Training is performed using the fit function with data generators, allowing for efficient training on large datasets. It also utilizes callbacks, such as model check-pointing, early stopping and learning rate reduction.

7. **Monitoring Training:** During training, the model's performance is monitored for both training and validation datasets. Key metrics like accuracy and loss are tracked over multiple epochs.

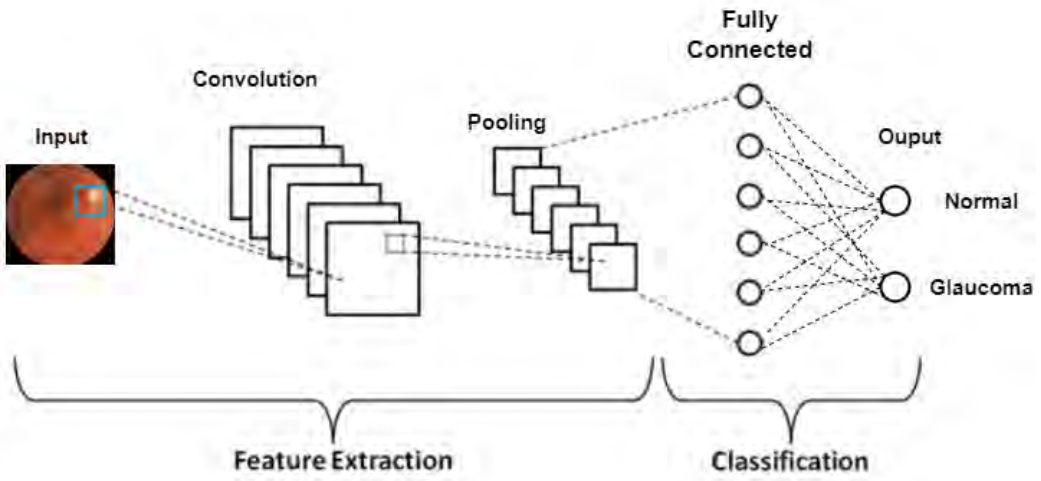


Figure 4.2: Simplified Architecture of Proposed CNN Model

4.1 Model Summary of Custom CNN

This model is specially designed to utilize the advantages of deep learning in image processing and make it suitable for tasks that require intricate feature extractions and analysis from elaborate and complicated images. The combination of encoder and decoder blocks enables a complete and thorough analysis of the data, which ensures an elaborate contextual understanding and a detailed local feature extraction of the input data.

Layer (type)	Output Shape	Parameter #
input_2 (InputLayer)	[(None, 256, 256, 3)]	0
conv_1 (Conv2D)	(None, 254, 254, 1024)	28672
norm_1 (BatchNormalization)	(None, 254, 254, 1024)	4096
max_pooling2d_3 (MaxPooling2D)	(None, 127, 127, 1024)	0
conv_2 (Conv2D)	(None, 125, 125, 512)	4719104
norm_2 (BatchNormalization)	(None, 125, 125, 512)	2048
max_pooling2d_4 (MaxPooling2D)	(None, 62, 62, 512)	0
conv_3 (Conv2D)	(None, 60, 60, 256)	1179904
norm_3 (BatchNormalization)	(None, 60, 60, 256)	1024
max_pooling2d_5 (MaxPooling2D)	(None, 30, 30, 256)	0
conv_4 (Conv2D)	(None, 28, 28, 128)	295040
norm_4 (BatchNormalization)	(None, 28, 28, 128)	512
max_pooling2d_6 (MaxPooling2D)	(None, 14, 14, 128)	0
conv_5 (Conv2D)	(None, 13, 13, 128)	65664
norm_5 (BatchNormalization)	(None, 13, 13, 128)	512
max_pooling2d_7 (MaxPooling2D)	(None, 6, 6, 128)	0
conv_6 (Conv2D)	(None, 6, 6, 64)	32832
norm_6 (BatchNormalization)	(None, 6, 6, 64)	256
max_pooling2d_8 (MaxPooling2D)	(None, 3, 3, 64)	0
dense (Dense)	(None, 3, 3, 64)	4160
conv_7 (Conv2D)	(None, 3, 3, 32)	8224
norm_7 (BatchNormalization)	(None, 3, 3, 32)	128
max_pooling2d_9 (MaxPooling2D)	(None, 1, 1, 32)	0
dense_1 (Dense)	(None, 1, 1, 32)	1056
conv_8 (Conv2D)	(None, 1, 1, 32)	4128
norm_8 (BatchNormalization)	(None, 1, 1, 32)	128
flatten (Flatten)	(None, 32)	0
dense_2 (Dense)	(None, 64)	2112
dense_3 (Dense)	(None, 1)	65

Total params: 6,349,665
Trainable params: 6,345,313
Non-trainable params: 4,352

Figure 4.3: Custom CNN Model Summary

Chapter 5

Pre-Trained CNN Models

5.1 InceptionV3

InceptionV3 is a 42-layer CNN model that is trained on the ImageNet dataset for a variety of computer vision implementations. As a residual network, InceptionV3 employs residual blocks to learn intricate features. Factorized convolutions is another approach used in InceptionV3. Convolutions that are factorized divide a larger convolution into two smaller convolutions that can be computed more quickly.[19]

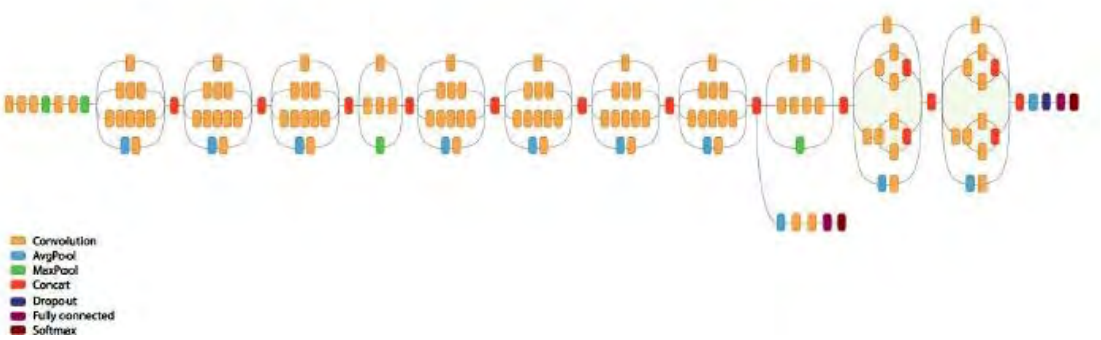


Figure 5.1: Architecture of InceptionV3

After the initial steps of the input layer and max pooling layer:

- The output is passed through a series of inception modules.
- Each inception module consists of a series of convolutional layers with kernel sizes such as 5x5, 3x3, or 1x1 convolutions. This allows the inception module to learn features at different scales.

After passing through the global max pooling layer, the procedure is similar to the CNN models. Through its parallel branches, InceptionV3 is renowned for its ability to capture fine-grained information and has attained top performance in several computer vision tests.

```
Model: "InceptionV3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 75, 75, 3)]       0
conv2d (Conv2D)              (None, 37, 37, 32)        864
batch_normalization (Batch...) (None, 37, 37, 32)      96
activation (Activation)      (None, 37, 37, 32)        0
...
(multiple layers with convolutions, activations, pooling)
...
mixed9 (Concatenate)         (None, 1, 1, 2048)        0
global_average_pooling2d (Glo (None, 2048)             0
dropout (Dropout)            (None, 2048)              0
dense (Dense)                (None, 1)                 2049
=================================================================
Total params: 21,804,833
Trainable params: 21,770,401
Non-trainable params: 34,432
```

Figure 5.2: InceptionV3 Model Summary

5.2 Densenet169

All the layers of DenseNet169's (CNN) architecture are linked to every other layer. The feature maps of the layers act as input for the above layers and vice versa. The ImageNet dataset has over 1 million images and 1,000 classifications, and was used to train this 169-layer CNN. In this architecture [6]:

- Input image is passed through a 7x7 convolution layer with 64 filters.
- Afterwards, the output is passed through a stride 2 max pooling layer. The image's dimensions are cut in half during this max pooling operation, which also selects the highest value found in each 2x2 pixel block. This procedure lowers the model's parameter count and improves the effectiveness of its training.
- The max pooling layer output is passed through a series of dense blocks. In every dense block, there is a sequence of convolutional layers, and each of these layers is linked to every other layer within the block.
- The last dense block output is passed through a global average pooling layer. This layer calculates the average of all pixel values in the image, contributing to a reduction in the model's parameter count and enhancing its resilience to noise.
- The global average pooling layer is passed through a completely connected layer with their outputs.

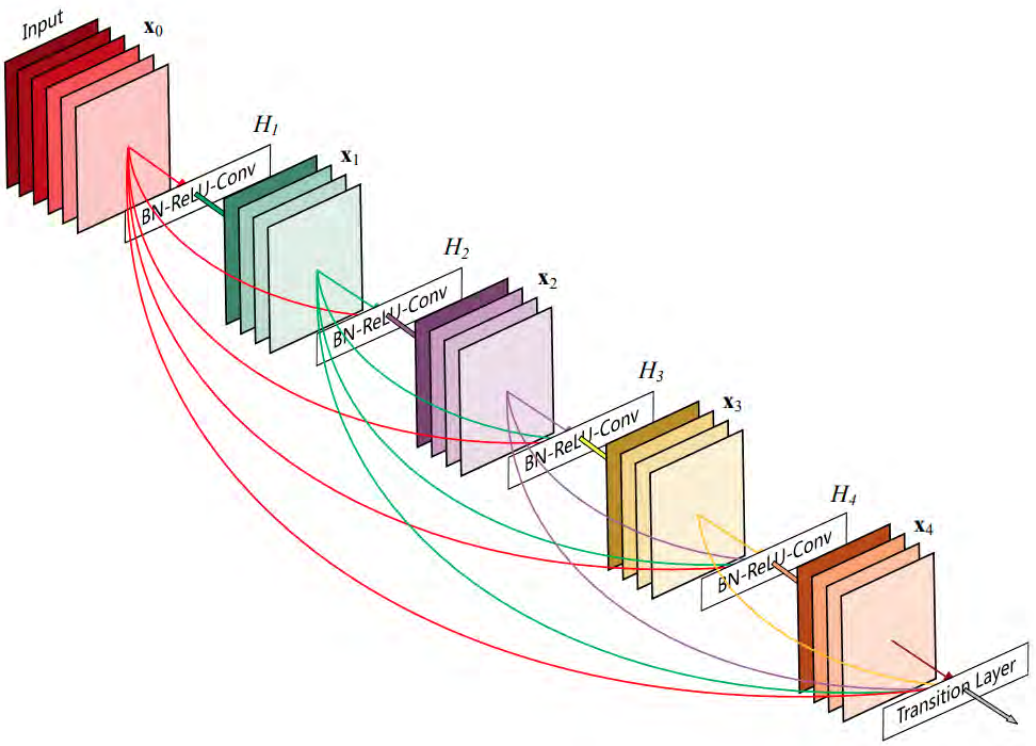


Figure 5.3: Architecture of DenseNet169

```
Model: "DenseNet169"
_________________________________________________________________
Layer (type)                 Output Shape              Parameter #
=================================================================
input_1 (InputLayer)         [(None, 64, 64, 3)]       0
conv2d (Conv2D)              (None, 32, 32, 64)        9472
batch_normalization (Batch...) (None, 32, 32, 64)      256
re_lu (ReLU)                 (None, 32, 32, 64)        0
max_pooling2d (MaxPooling2D) (None, 16, 16, 64)        0
...
(multiple layers with convolutions, batch normalizations, ReLU
activations, concatenations)
...
conv2d_85 (Conv2D)           (None, 2, 2, 32)          18464
concatenate_81 (Concatenate) (None, 2, 2, 64)          0
max_pooling2d_4 (MaxPooling2D) (None, 1, 1, 64)        0
flatten (Flatten)            (None, 64)                0
dense (Dense)                (None, 128)               8320
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 1,532,193
Trainable params: 1,521,377
Non-trainable params: 10,816
```

Figure 5.4: DenseNet169 Model Summary

5.3 EfficientNetB4

This is an EfficientNetB4 architecture-based, 380x380 convolutional neural network (CNN) model that is pre-trained on ImageNet (a large dataset). It starts with the EfficientNetB4 base model, which has already learned valuable image features from a diverse set of images.

EfficientNetB4 is a type of neural network that scales all of its dimensions (depth, width, and resolution) equally. This makes it more accurate and efficient than other neural networks that are scaled more traditionally.

- After each convolutional layer, batch normalization and activation techniques like ReLU are frequently used to speed up training and improve the execution of the network.
- Efficient Channel Attention (ECA) is a method that EfficientNetB4 uses to focus on important channels inside feature maps while minimizing computational complexity.
- It has a deeper and wider (more channels) design than its smaller counterparts. The larger breadth improves feature representation, and the deeper the model can go, the more complicated and hierarchical features it can capture.
- Operates more frequently on high-resolution photos than on tiny variations. For activities needing fine-grained details, this may be advantageous.
- Over the basic model, a Global Average Pooling 2D layer is added. This layer reduces the spatial size of the feature maps while maintaining critical information. A dense layer with a single neuron and a sigmoid activation algorithm is added to produce the final result.

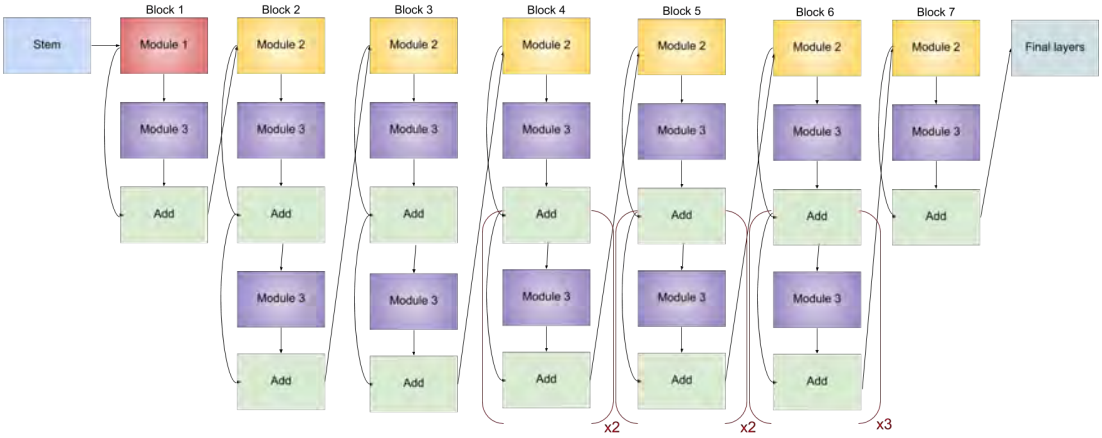


Figure 5.5: Architecture of EfficientNetB4

```
Layer (type)                 Output Shape              Param #     Connected to
==================================================================================================
input_2 (InputLayer)         [(None, 112, 112, 3)] 0           []
rescaling_2 (Rescaling)      (None, 112, 112, 3)   0           ['input_2[0][0]']
normalization_1 (Normalization)(None, 112, 112, 3)   7           ['rescaling_2[0][0]']
rescaling_3 (Rescaling)      (None, 112, 112, 3)   0           ['normalization_1[0][0]']
stem_conv_pad (ZeroPadding2D)  (None, 113, 113, 3)   0           ['rescaling_3[0][0]']
stem_conv (Conv2D)           (None, 56, 56, 48)    1296        ['stem_conv_pad[0][0]']
stem_bn (BatchNormalization)   (None, 56, 56, 48)    192         ['stem_conv[0][0]']
stem_activation (Activation)   (None, 56, 56, 48)    0           ['stem_bn[0][0]']
... [Layers continue in similar pattern] ...
block7b_project_conv (Conv2D)  (None, 4, 4, 448)     1204224     ['block7b_se_excite[0][0]']
block7b_project_bn (BatchNorm) (None, 4, 4, 448)     1792        ['block7b_project_conv[0][0]']
block7b_drop (Dropout)       (None, 4, 4, 448)     0           ['block7b_project_bn[0][0]']
block7b_add (Add)            (None, 4, 4, 448)     0           ['block7b_drop[0][0]']
top_conv (Conv2D)            (None, 4, 4, 1792)    802816      ['block7b_add[0][0]']
top_bn (BatchNormalization)    (None, 4, 4, 1792)    7168        ['top_conv[0][0]']
top_activation (Activation)    (None, 4, 4, 1792)    0           ['top_bn[0][0]']
global_avg_pooling2d (GlobalAv (None, 1792)          0           ['top_activation[0][0]']
dropout (Dropout)            (None, 1792)          0           ['global_avg_pooling2d[0][0]']
dense (Dense)                (None, 1)             1793        ['dropout[0][0]']
==================================================================================================
Total params: 17,675,616
Trainable params: 17,550,409
Non-trainable params: 125,207
```

Figure 5.6: EfficientNetB4 Model Summary

5.4 VGG16

Despite being a relatively straightforward CNN architecture, VGG16 is quite successful. The input image is processed using a series of convolution layers and max pooling layers to extract its features. To categorize the image, the characteristics are subsequently transferred to a fully linked layer.

- The depth of VGG16 is 16 layers, as per the name.
- To extract the features, VGG16 uses 3x3 convolution layers with stride 1 where the filter moves one pixel at a time.
- After many convolution and max-pooling layers, VGG16 usually contains one or more fully connected layers. Usually, these layers are used for categorization at the end.

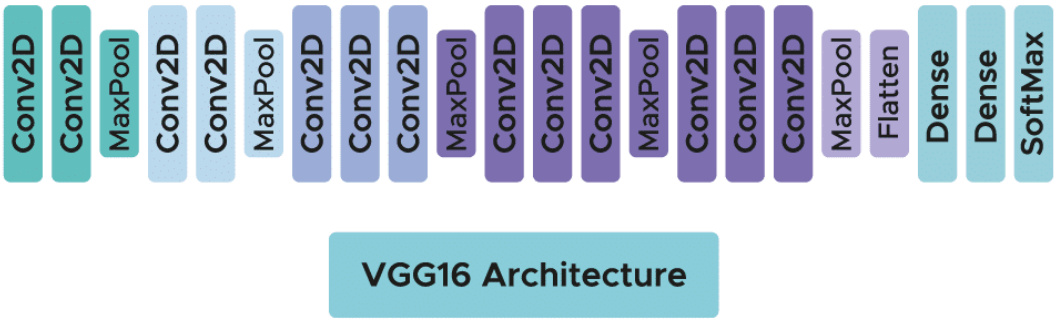


Figure 5.7: Architecture of VGG16

```
Model: "VGG16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 64)      1792
conv2d_1 (Conv2D)            (None, 224, 224, 64)      36928
max_pooling2d (MaxPooling2D) (None, 112, 112, 64)      0
conv2d_2 (Conv2D)            (None, 112, 112, 128)     73856
conv2d_3 (Conv2D)            (None, 112, 112, 128)     147584
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 128)     0
conv2d_4 (Conv2D)            (None, 56, 56, 256)       295168
conv2d_5 (Conv2D)            (None, 56, 56, 256)       590080
conv2d_6 (Conv2D)            (None, 56, 56, 256)       590080
max_pooling2d_2 (MaxPooling2D) (None, 28, 28, 256)     0
conv2d_7 (Conv2D)            (None, 28, 28, 512)       1180160
conv2d_8 (Conv2D)            (None, 28, 28, 512)       2359808
conv2d_9 (Conv2D)            (None, 28, 28, 512)       2359808
max_pooling2d_3 (MaxPooling2D) (None, 14, 14, 512)     0
conv2d_10 (Conv2D)           (None, 14, 14, 512)       2359808
conv2d_11 (Conv2D)           (None, 14, 14, 512)       2359808
conv2d_12 (Conv2D)           (None, 14, 14, 512)       2359808
max_pooling2d_4 (MaxPooling2D) (None, 7, 7, 512)       0
flatten (Flatten)            (None, 25088)             0
dense (Dense)                (None, 4096)              102764544
dense_1 (Dense)              (None, 4096)              16781312
dense_2 (Dense)              (None, 2)                 8194
=================================================================
Total params: 134,268,738
Trainable params: 134,268,738
Non-trainable params: 0
```

Figure 5.8: VGG16 Model Summary

5.5 Squeezenet

SqueezeNet is intended to be a compact and effective CNN architecture for usage in embedded and mobile systems. Useful features have been learned using a pre-trained SqueezeNet model from torchvision that has been pre-trained on a sizable dataset (like ImageNet). Two output neurons (one for each class in binary classification) are added to the model's last layer.

- *Fire Modules* are the architecture's building elements that make up the majority of SqueezeNet. Two basic parts make up a fire module, an *expand* layer and a *squeeze* layer.
- A 1x1 convolution layer (squeeze layer) with a small number of filters.
- SqueezeNet is distinguished by its small model size, which was attained by using 1x1 convolutions and channel-wise compression.

Due to the architecture's efficiency, it can be installed on hardware with constrained computing capabilities.

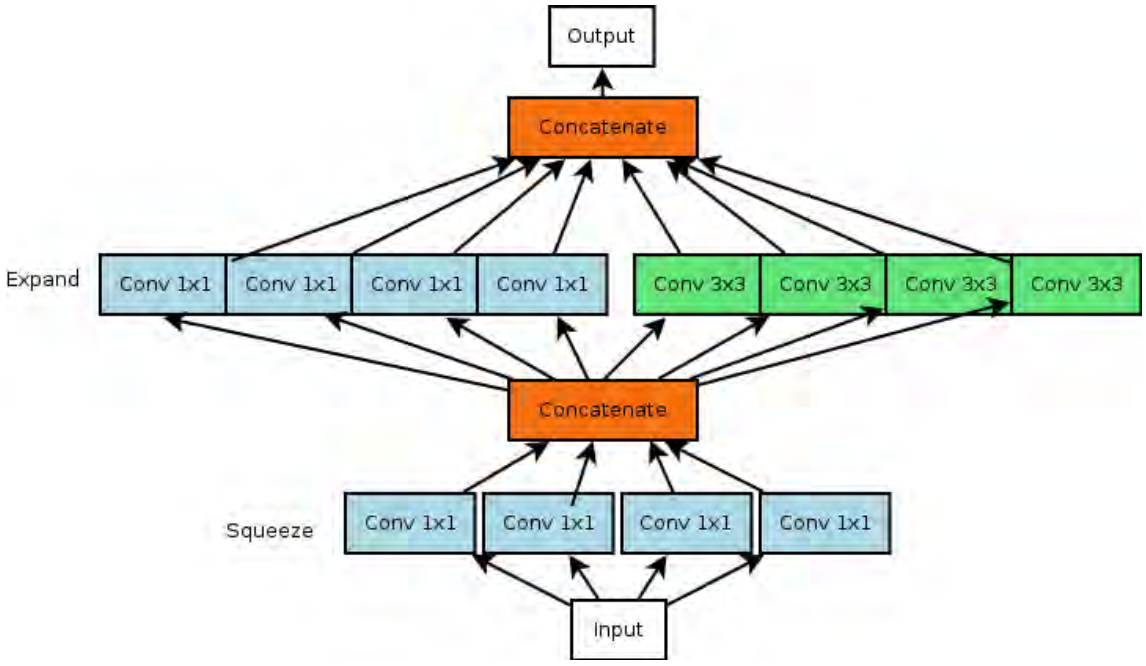


Figure 5.9: Architecture of Squeezenet

Layer (type)	Output Shape	Parameter #
Conv2d-1	[-1, 64, 111, 111]	1,792
ReLU-2	[-1, 64, 111, 111]	0
MaxPool2d-3	[-1, 64, 55, 55]	0
Conv2d-4	[-1, 16, 55, 55]	1,040
ReLU-5	[-1, 16, 55, 55]	0
Conv2d-6	[-1, 64, 55, 55]	1,088
ReLU-7	[-1, 64, 55, 55]	0
Conv2d-8	[-1, 64, 55, 55]	9,280
ReLU-9	[-1, 64, 55, 55]	0
Fire-10	[-1, 128, 55, 55]	0
... (Similar patterns with Conv2d, ReLU, Fire modules) ...		
Conv2d-59	[-1, 256, 13, 13]	147,712
ReLU-60	[-1, 256, 13, 13]	0
Fire-61	[-1, 512, 13, 13]	0
Dropout-62	[-1, 512, 13, 13]	0
Conv2d-63	[-1, 2, 13, 13]	1,026
ReLU-64	[-1, 2, 13, 13]	0
AdaptiveAvgPool2d-65	[-1, 2, 1, 1]	0

Total params: 723,522
Trainable params: 723,522
Non-trainable params: 0

Figure 5.10: Squeezenet Model Summary

5.6 AlexNet

AlexNet is a prominent CNN model architecture that did great in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. This model is pre-trained on ImageNet with a vast dataset.

- **Layer Design:** The architecture starts with convolution layers, followed by batch normalization and Rectified Linear Unit (ReLU).
- **Local Response Normalization (LRN):** AlexNet implements LRN which increases the model's ability to capture intricate patterns and details.
- **Multiple GPUs:** AlexNet uses two GPUs during training.
- **Dropout:** Dropout is implemented in the fully connected layers to prevent over-fitting, enhancing the model's generalization capabilities.
- **Deeper Architecture:** AlexNet's architecture is relatively deep for its time, comprising eight weight layers. The increased depth allows for the capture of hierarchical features in image data.
- **Global Average Pooling (GAP):** A Global Average Pooling 2D layer is employed to reduce the spatial dimensions of the feature maps while retaining essential information. A final dense layer with a single neuron and sigmoid activation generates the ultimate prediction.

The AlexNet model has had a profound impact on the field of computer vision, setting the stage for subsequent advancements in deep learning architectures.

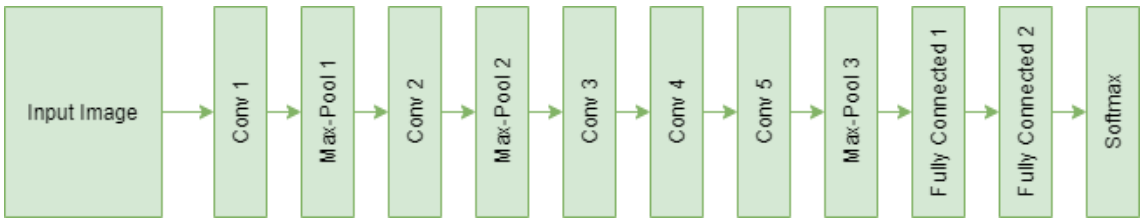


Figure 5.11: Architecture of AlexNet

Layer (type)	Output Shape	Parameter#
Conv2d-1	[-1, 64, 55, 55]	23,296
ReLU-2	[-1, 64, 55, 55]	0
MaxPool2d-3	[-1, 64, 27, 27]	0
Conv2d-4	[-1, 192, 27, 27]	307,392
ReLU-5	[-1, 192, 27, 27]	0
MaxPool2d-6	[-1, 192, 13, 13]	0
Conv2d-7	[-1, 384, 13, 13]	663,936
ReLU-8	[-1, 384, 13, 13]	0
Conv2d-9	[-1, 256, 13, 13]	884,992
ReLU-10	[-1, 256, 13, 13]	0
Conv2d-11	[-1, 256, 13, 13]	590,080
ReLU-12	[-1, 256, 13, 13]	0
MaxPool2d-13	[-1, 256, 6, 6]	0
AdaptiveAvgPool2d-14	[-1, 256, 6, 6]	0
Dropout-15	[-1, 9216]	0
Linear-16	[-1, 4096]	37,752,832
ReLU-17	[-1, 4096]	0
Dropout-18	[-1, 4096]	0
Linear-19	[-1, 4096]	16,781,312
ReLU-20	[-1, 4096]	0
Linear-21	[-1, 2]	8,194

Total params: 57,012,034
Trainable params: 57,012,034
Non-trainable params: 0

Figure 5.12: AlexNet Model Summary

Chapter 6

Result and Analysis

In our research, we have presented the accuracy and loss graph for each epoch in the training and validation phases. The graphical representation helps us to identify the performance and generalization of each model to detect Glaucoma on unseen data.

We have used two groups of datasets and have presented the graphical charts for each. We have analyzed each model on the Main Dataset (REFUGE) and the Secondary Dataset (Combined) after data pre-processing.

6.1 Model Comparison with Pre-trained CNN Models

Model Name	Accuracy	Loss	Validation Accuracy	Validation Loss	Parameters(Million)
Custom	98.71%	0.0682	97.75%	0.0531	6.35
AlexNet	50.13%	0.6931	50.00%	0.6934	62.3
DenseNet169	72.31%	0.5671	68.01%	0.8391	14.14
EfficientNetB4	95.05%	0.0142	50.00%	0.7730	17.55
InceptionV3	90.02%	0.3811	91.16%	0.2011	25.12
SqueezeNet	50.56%	0.6931	50.00%	0.6931	0.732
VGG16	53.02%	0.6930	52.89%	0.6931	167.83

Table 6.1: Table for Model Performance Comparison (Main Dataset-REFUGE)

Model Name	Accuracy	Loss	Validation Accuracy	Validation Loss	Parameters(Million)
Custom	65.01%	0.6152	72.17%	0.4891	6.35
AlexNet	51.96%	0.6930	50.16%	0.6940	62.3
DenseNet169	53.33%	0.0982	52.10%	0.7341	14.14
EfficientNetB4	94.38%	0.1214	63.46%	1.4322	17.55
InceptionV3	91.29%	0.1985	80.88%	0.3433	25.12
SqueezeNet	52.09%	0.6931	51.44%	0.6931	0.732
VGG16	67.49%	0.6003	61.08%	0.6420	167.83

Table 6.2: Table for Model Performance Comparison (Secondary Dataset)

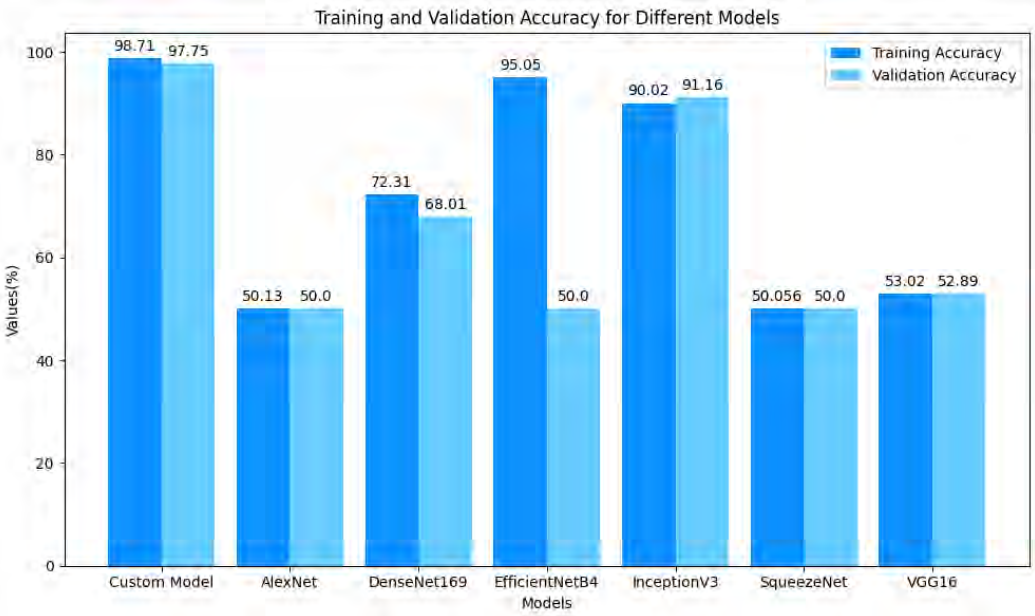


Figure 6.1: Bar-Chart of Model Comparison (Main Dataset-REFUGE)

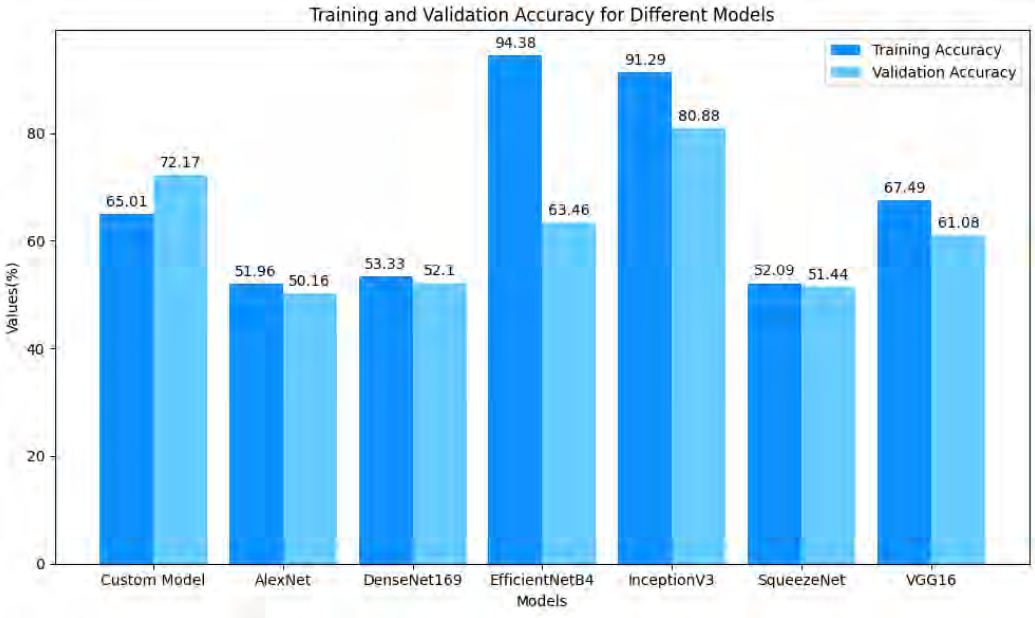
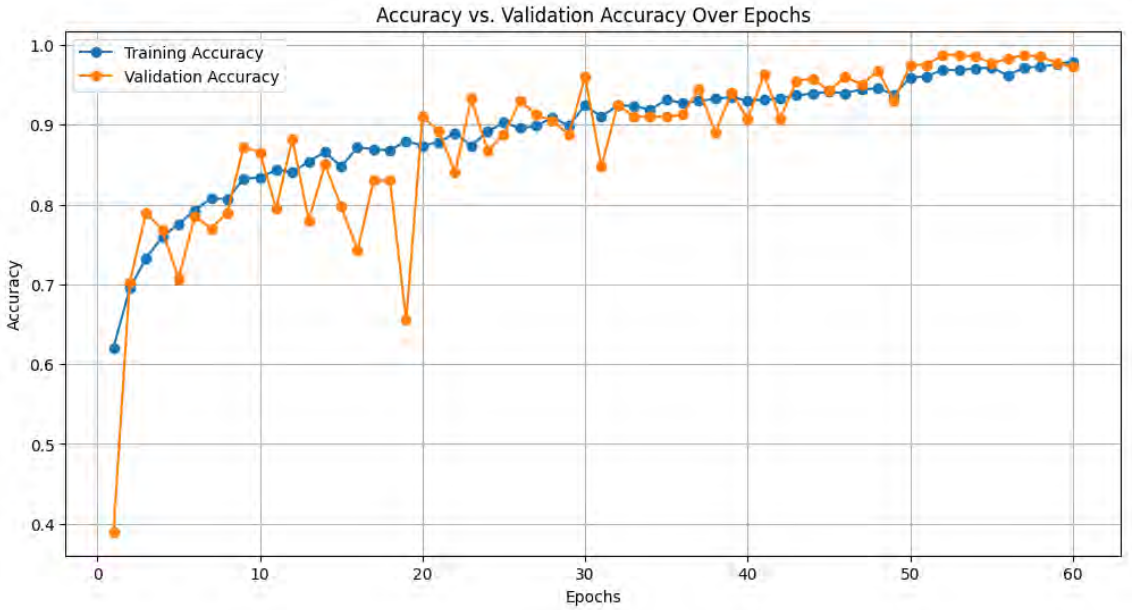


Figure 6.2: Bar-Chart of Model Comparison (Secondary Dataset)

6.2 Performance Analysis

6.2.1 Custom CNN Model

Our custom model was able to achieve a successful peak of 98.71% in the training accuracy which is the highest in comparison with the pre-trained models on the main dataset. From the very beginning, the accuracy escalated to 60% and throughout the 60 epochs, it continuously showed impressive fluctuations. The validation accuracy showed similar performance and reached 97.75%. Even though it dropped to 70% at 20 epochs, but it could reach back to 90% fast. Whereas, the training loss gradually declined from 0.73 to 0.0682. The validation loss on the other hand had more fluctuations throughout but ended down at 0.0531.



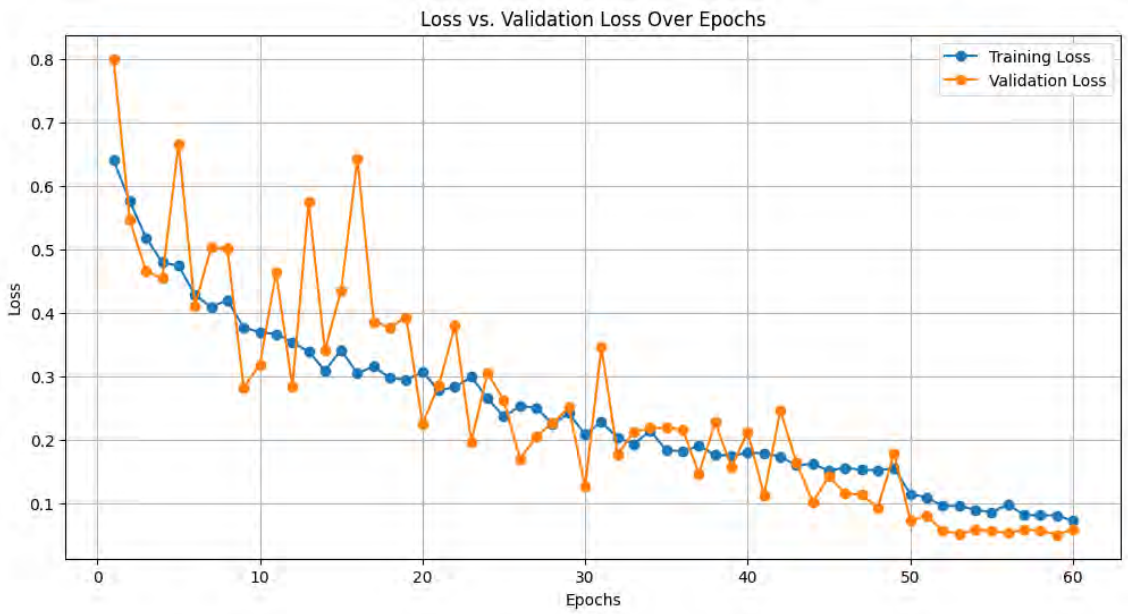
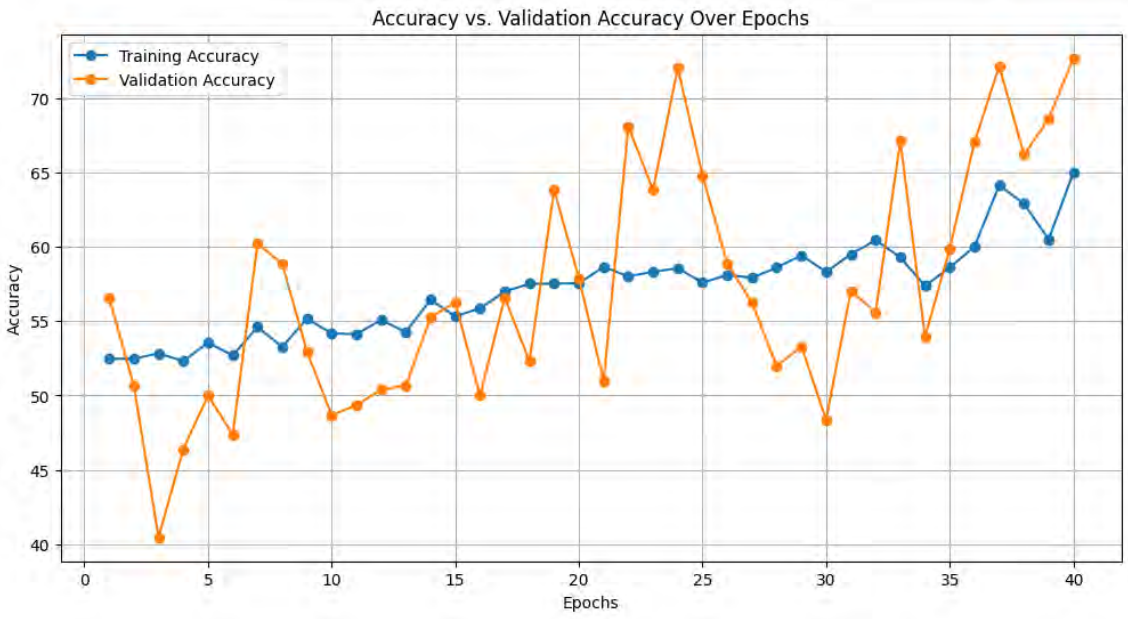


Figure 6.3: Custom CNN Training and Validation (Main Dataset)

For the secondary dataset, our custom model obtained 65.01% in training accuracy. For the validation accuracy, we can observe different shifts in the graph in almost every 3 epochs. The value peaked at 72.17%. The training and validation loss, on the other hand, had a similar pattern resulting in 0.6152 and 0.4891 respectively.



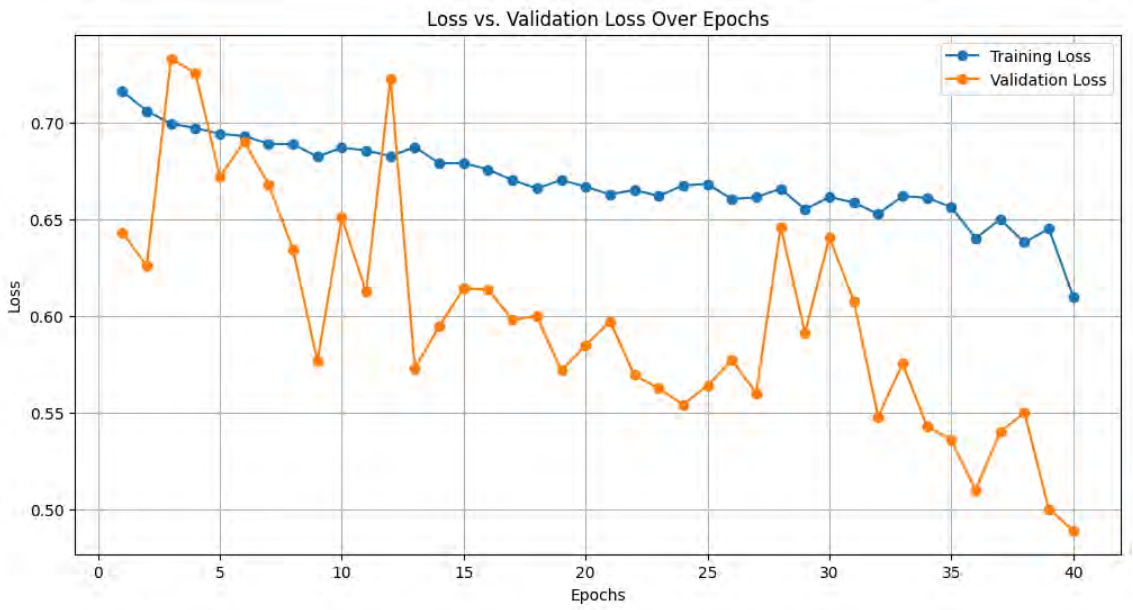
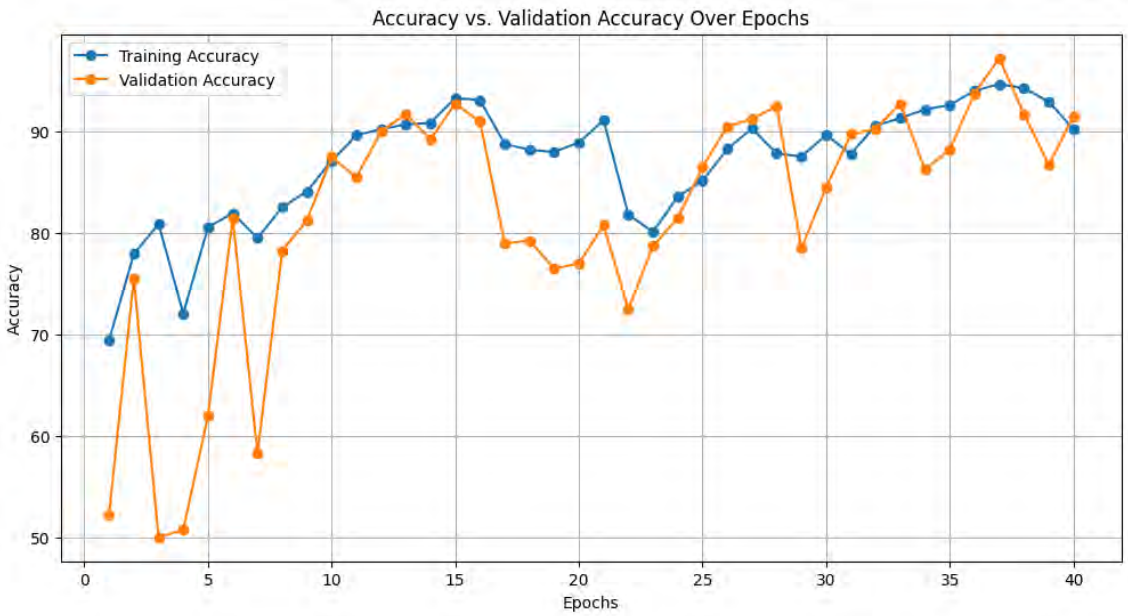


Figure 6.4: Custom CNN Training and Validation (Secondary Dataset)

6.2.2 InceptionV3

The InceptionV3 model has thrivingly achieved 90.02% training accuracy as we trained with our main dataset which is amongst one of the highest we were able to achieve. Even though there was a rough start in the beginning within the range of the first 5 epochs between 70%-80% after that, the accuracy gradually increased to 90%. Towards the 22nd epoch, even though we observed a drop from 90% to 80%, the model was able to achieve the height of 91% and ended with 90.02%. But, in terms of the Validation accuracy, the fluctuations have been more and frequent. Starting from massive heights to about 78% to huge drops to less than 50% within the first 5 epochs. With the shifts, the validation accuracy reached its peak after the 35 epochs and ended at 91.16%. On the other hand, the training loss was smoother, the validation loss, being similar to the validation accuracy, went through variations of shifts throughout.



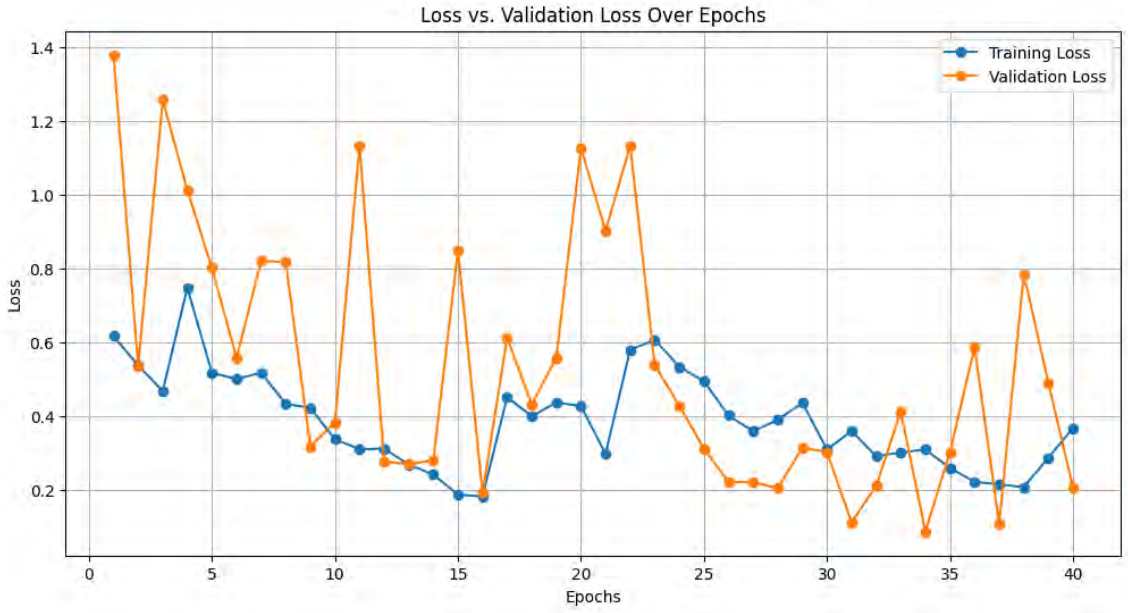


Figure 6.5: InceptionV3 Training and Validation (Main Dataset)

The InceptionV3 model achieved a training accuracy of 92.29% with balanced performance when trained with the secondary dataset. Throughout 60 epochs, training and validation losses gradually decrease from over 0.7 where the validation loss decreases from 0.7 to 0.35 and the training loss decreases from over 0.7 to below 0.2. Alongside, the validation and training accuracy increases over time from the first epoch to the last epoch. The training accuracy increases from over 50% and goes above 90% and the validation accuracy increases from over 50% to approximately 80%.

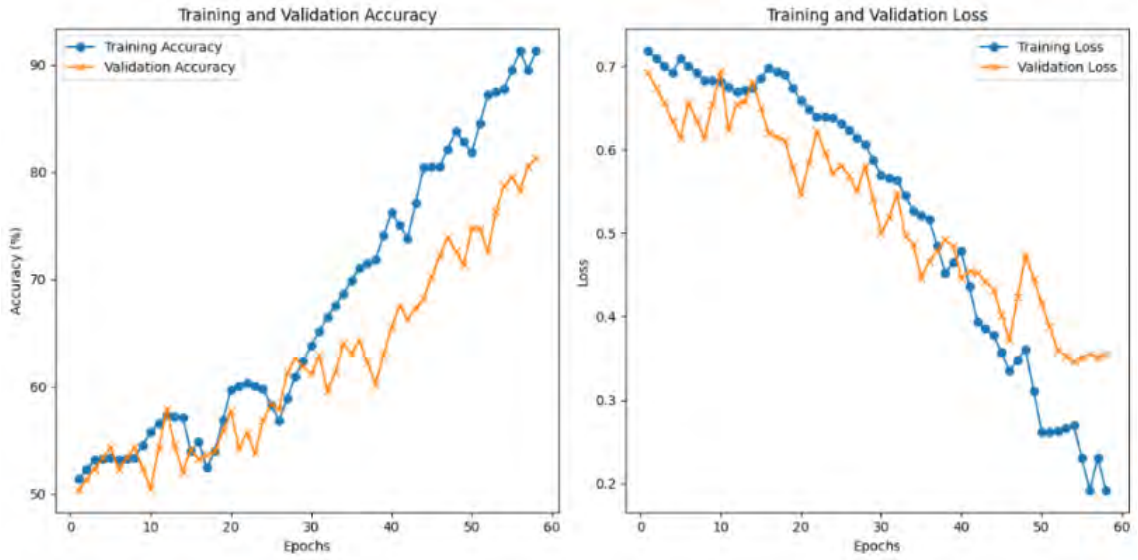


Figure 6.6: InceptionV3 Training and Validation (Secondary Dataset)

6.2.3 DenseNet169

Up until the first 10 epochs, the training accuracy remained quite unsatisfactory for our main dataset, which is less than 60%. But, it rapidly escalated after the next 5-7 epochs, where we can observe the value to be about 64%. By the end of 40 epochs, we were able to achieve 72.13%. The validation accuracy remained nearly constant in the first 6 epochs but peaked suddenly at 18 epochs which was above 60%. With frequent alterations, it ended with 68.01%. For our main dataset, the performance of DenseNet169 is observed to be substandard. The training loss peaked at the very end with comparably fewer shifts in the middle and the validation loss gradually decreased to less than 60%.

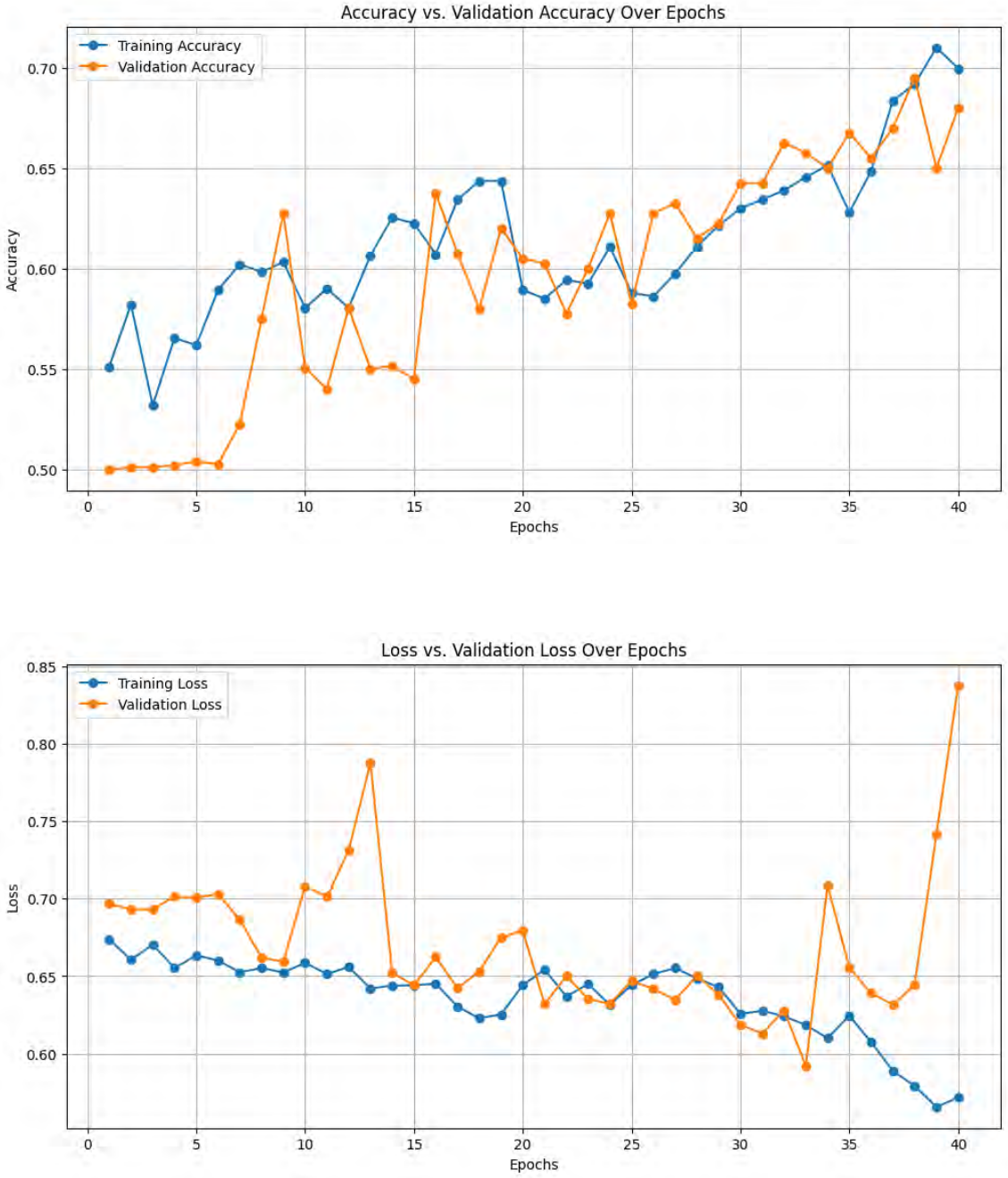


Figure 6.7: DenseNet169 Training and Validation (Main Dataset)

This information provides a snapshot of the validation and training performance of the DenseNet169 model on the secondary dataset. Throughout 40 epochs, the training loss gradually decreases from approximately 0.7025 to 0.6926, indicating that the model is learning to fit the training data better. The training accuracy has different alterations but could only achieve 53.33%, demonstrating that the model becomes moderately accurate in predicting the training data. Conversely, the validation loss fluctuates but generally decreases from approximately 0.6927 to 0.6924, showing that the model generalizes well to the validation data. The validation accuracy also improves from 50.56% to 52.10%, indicating that the model's performance on unseen data is not very satisfying.

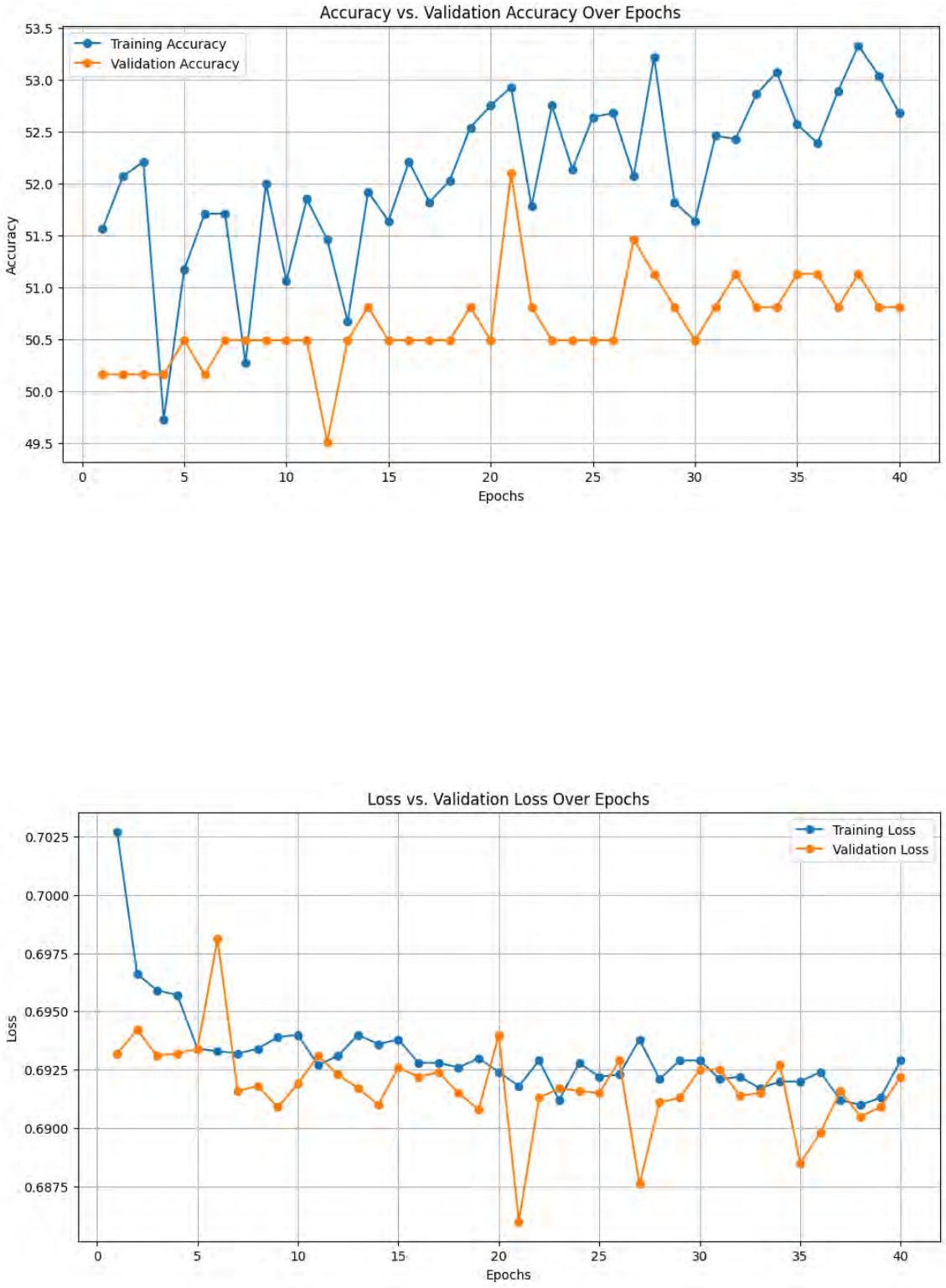


Figure 6.8: DenseNet169 Training and Validation (Secondary Dataset)

6.2.4 EfficientNetB4

The EfficientNetB4 model was able to give a rather constant set of accuracy throughout the 40 epochs for our main dataset. It peaked at the very start of model training straight to 90% and by the end, had achieved 95.05% which is the second highest in our training accuracy collection. Observing the performance overall, we are identifying it to be over-fitting. The validation accuracy on the other hand had some shifts but ended up with 50% accuracy. Similar to the accuracy plotting, the training loss plotting has the same patterns of constant values ending at 0.0142, but the validation loss graph had the most frequent fluctuations this time ending at above 0.5.

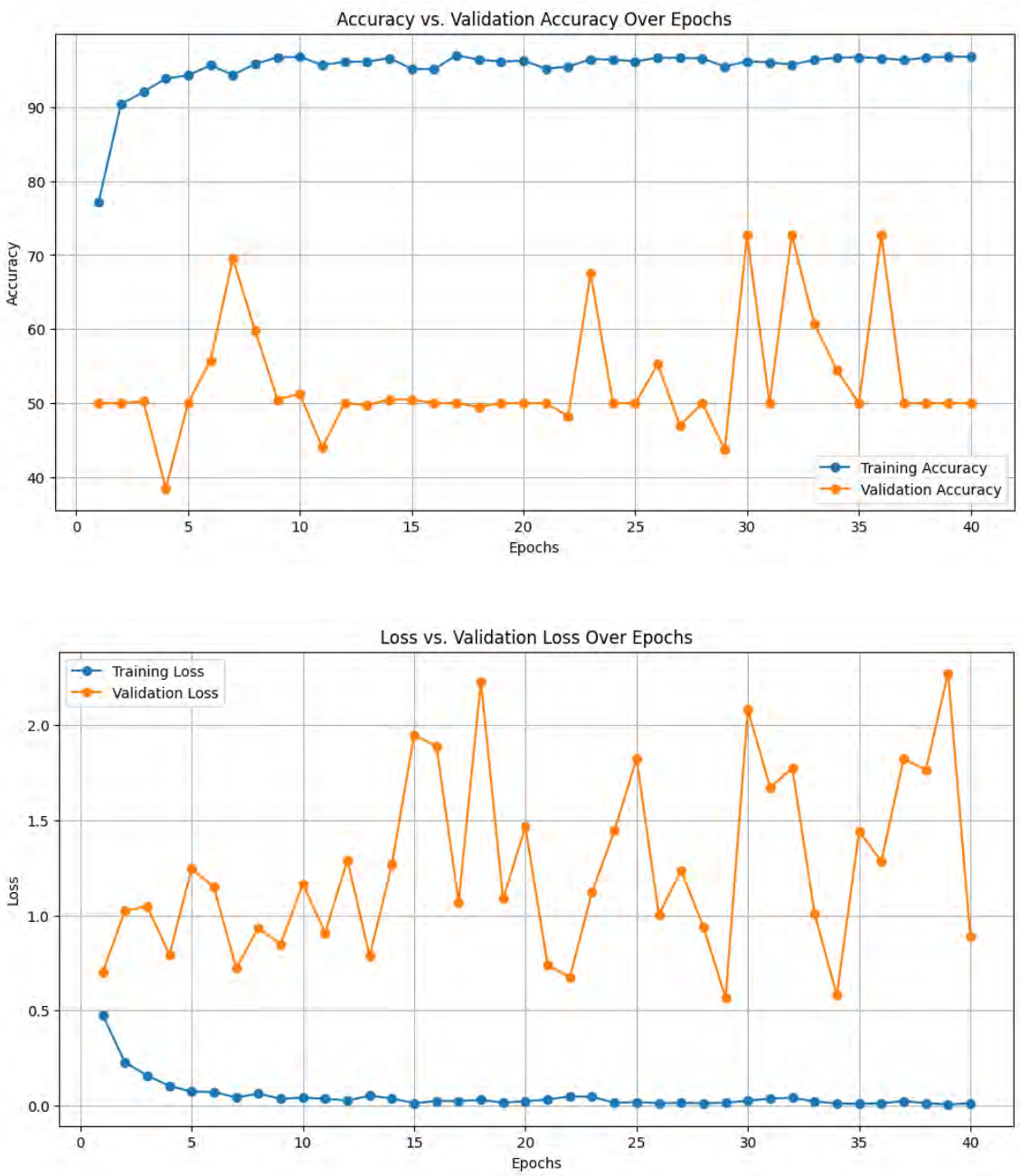


Figure 6.9: EfficientNetB4 Training and Validation (Main Dataset)

The training loss steadily decreased from epoch 1 to epoch 60. This indicates that the model is learning to reduce the differences between the predictions and the actual target values in the training dataset. The loss reduction signifies that the model is converging during training. The training accuracy steadily increases from around 51.62% in the first epoch to approximately 94.38% in the last epoch. This indicates that the model is improving its ability to correctly classify training samples over time. The validation accuracy, which measures the model's performance on data it has never seen before, also shows fluctuations but generally is constant. It starts at 50% in the first epoch and reaches 49.84% in the last epoch.

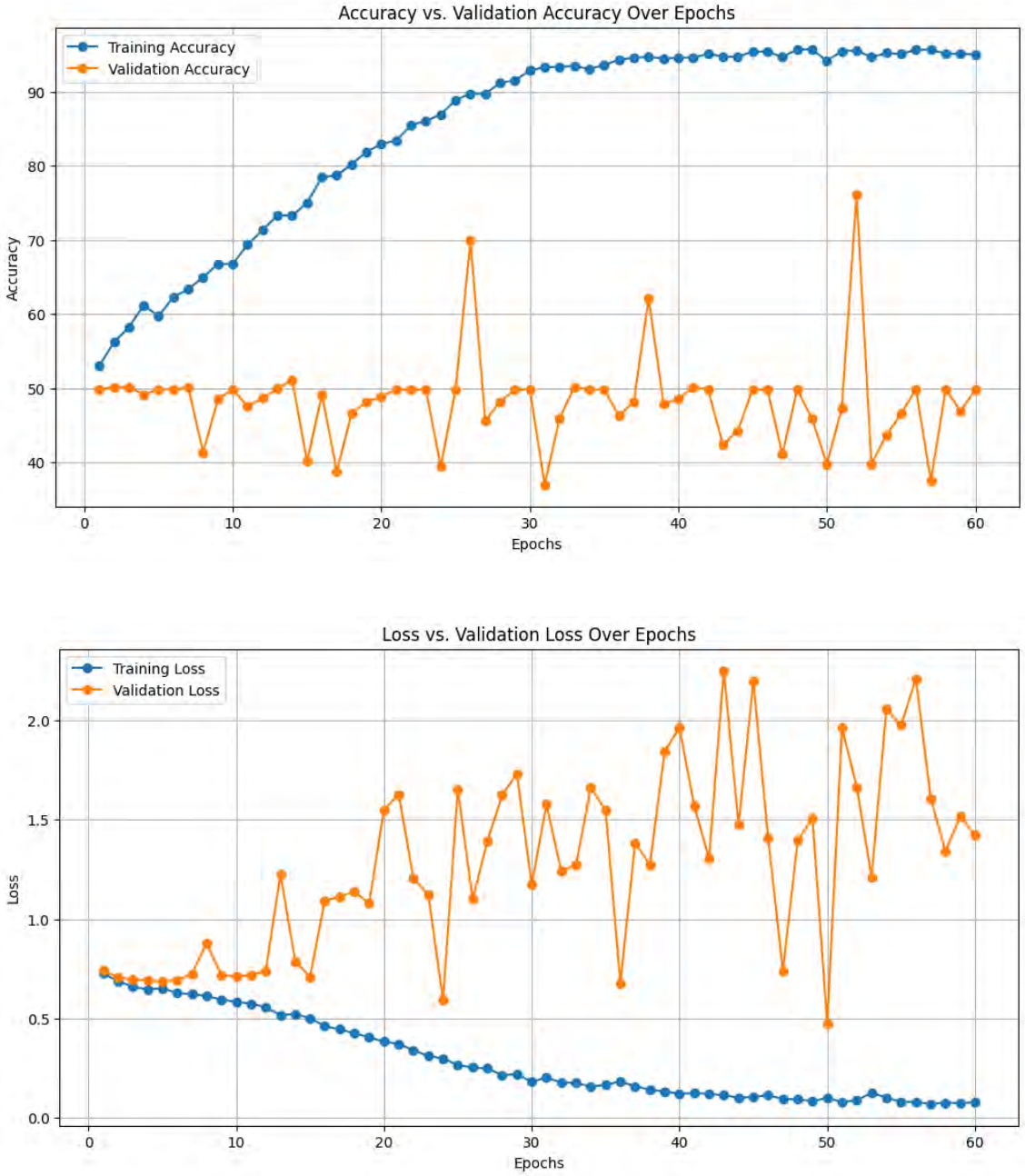


Figure 6.10: EfficientNetB4 Training and Validation (Secondary Dataset)

6.2.5 VGG16

In terms of deep learning, VGG16 is observed to limit itself in terms of working with deep networks, and as a result, even though we observed fluctuations in almost every 3 epochs, the model could not achieve training accuracy of more than 55%. The same goes for the validation accuracy. In terms of the training and validation loss, the model has performed nearly in a similar pattern and peaked at 0.69.

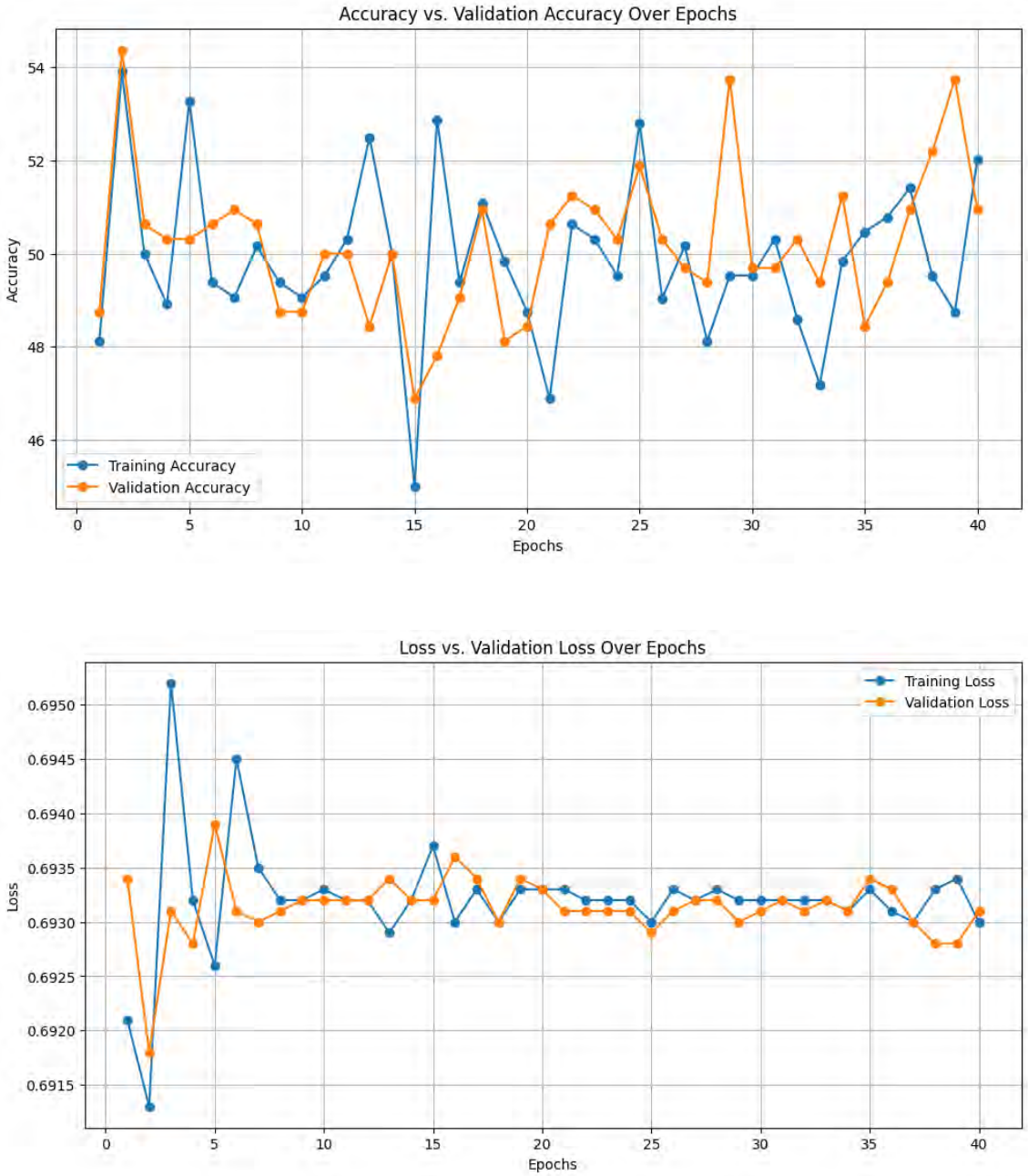


Figure 6.11: VGG16 Training and Validation (Main Dataset)

The VGG16 model exhibits fluctuations in training and validation accuracy throughout training. Starting with a validation accuracy of 51.62% and a

training accuracy of 51.42% in the first epoch, the model's performance undergoes variations but generally shows improvement. Training accuracy gradually increased to approximately 67.49%, while validation accuracy fluctuates around 50%-65%. These fluctuations suggest that the model is learning from the training data but may struggle to generalize well to the validation dataset. Overall, the VGG16 model achieves moderate performance, with the potential for further refinement to reduce over-fitting and improve validation accuracy. Further analysis and fine-tuning may be needed to enhance its generalization capabilities.

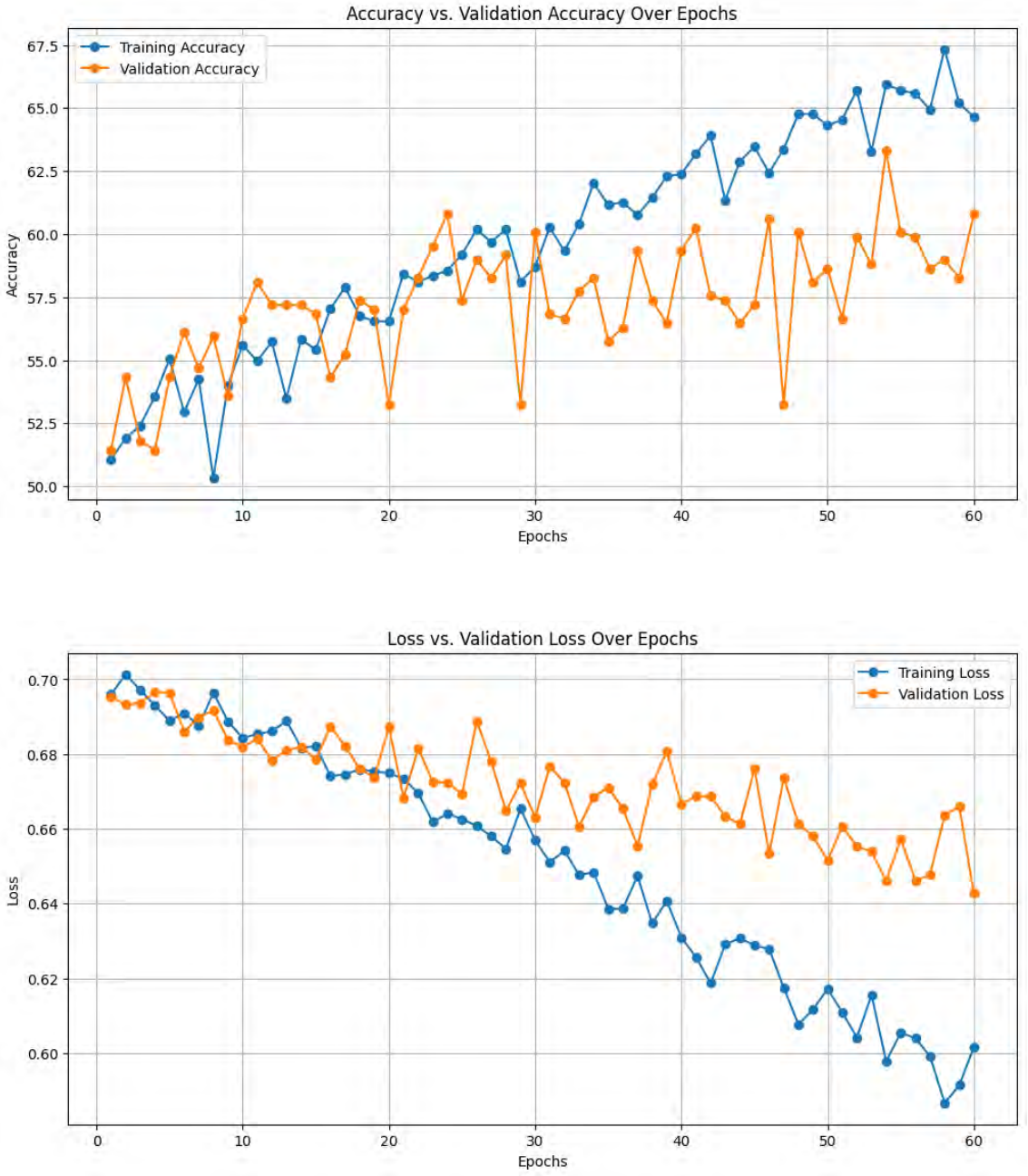


Figure 6.12: VGG16 Training and Validation (Secondary Dataset)

6.2.6 SqueezeNet

The Squeezenet was not able to present a variation of alterations throughout the 40 epochs in both the cases of training and validation accuracy. In the very beginning, the training accuracy peaked at 50.56% and by the end, it had dropped to less than 49% and remained constant in most of the epochs. Whereas, the validation accuracy seemed to be adamantly stable at the value of 50%. Change of picture in terms of the training loss where we observed some shifts in the plot but still not impressive. It peaked at 0.69 only in the 3rd epoch, where the validation loss here also remained constant throughout.

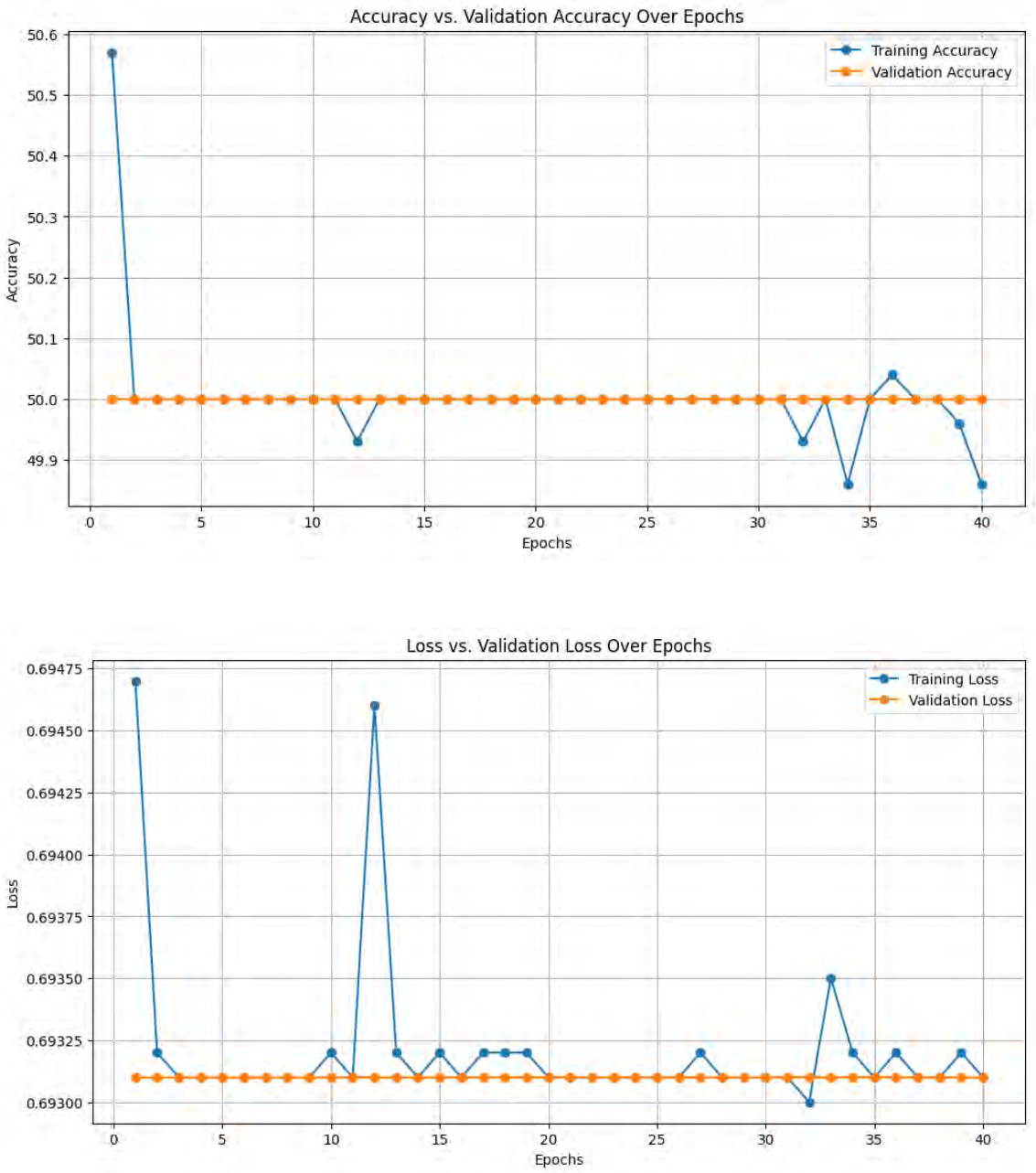


Figure 6.13: Squeezenet Training and Validation (Main Dataset)

The training process for the SqueezeNet model over 60 epochs resulted in limited progress, maintaining a training accuracy of around 52.09% and a stagnant validation accuracy of 51.44% throughout most of the training. This suggests that the model may not have been effectively learning from the data or that the model architecture and the hyperparameters need adjustment to improve performance. Further analysis and fine-tuning are recommended to enhance the model's learning capability and achieve better validation accuracy.

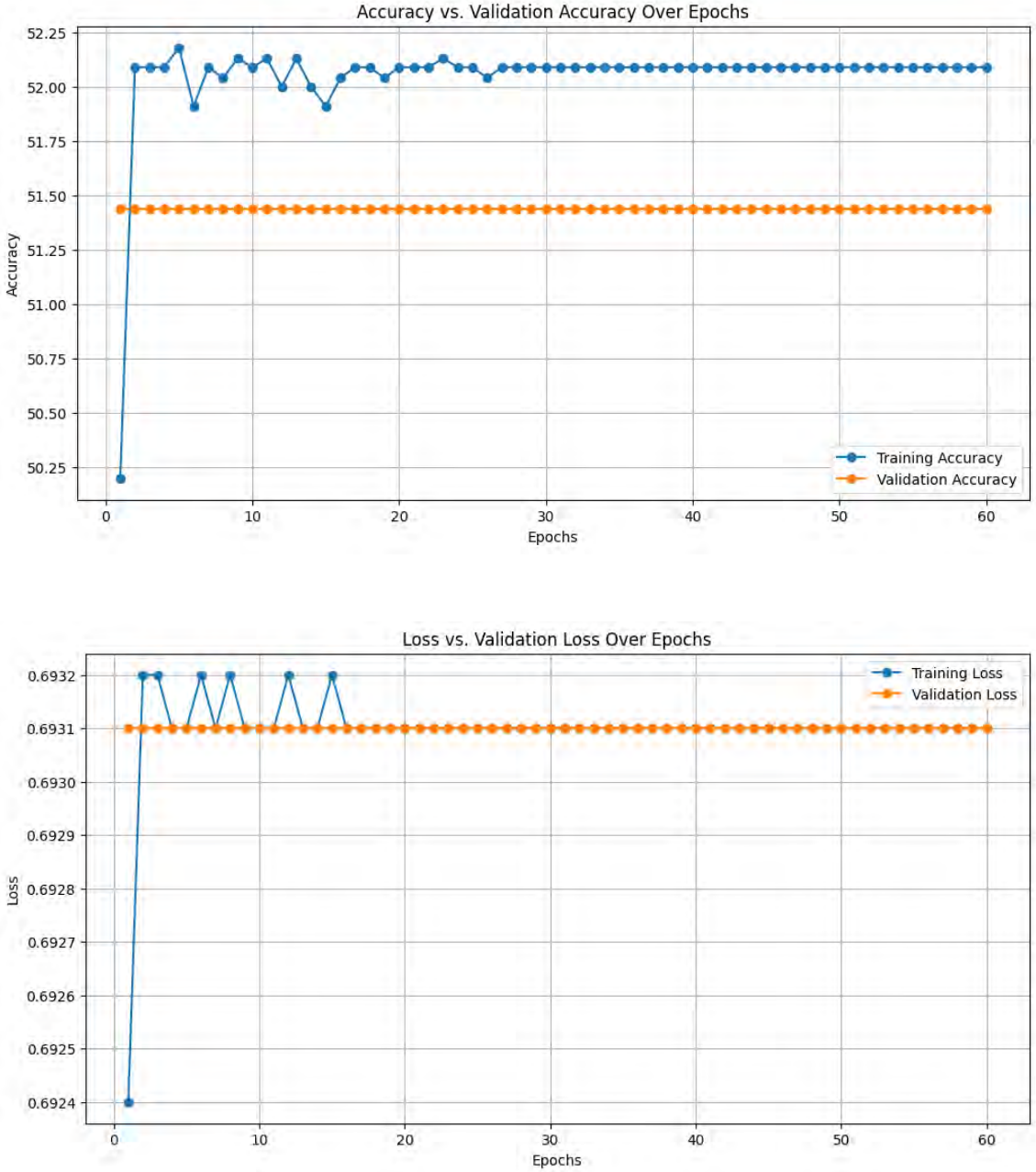


Figure 6.14: Squeezenet Training and Validation (Secondary Dataset)

6.2.7 AlexNet

When trained on the main dataset, Alexnet gave the constant value of 50% throughout the 40 epochs for validation accuracy but continuous variations were seen for the training accuracy but could only reach the height of 50.13%. The training and validation loss remained similar to the training and validation loss that we obtained in terms of the secondary dataset also, that is 0.693 and 0.694 respectively.

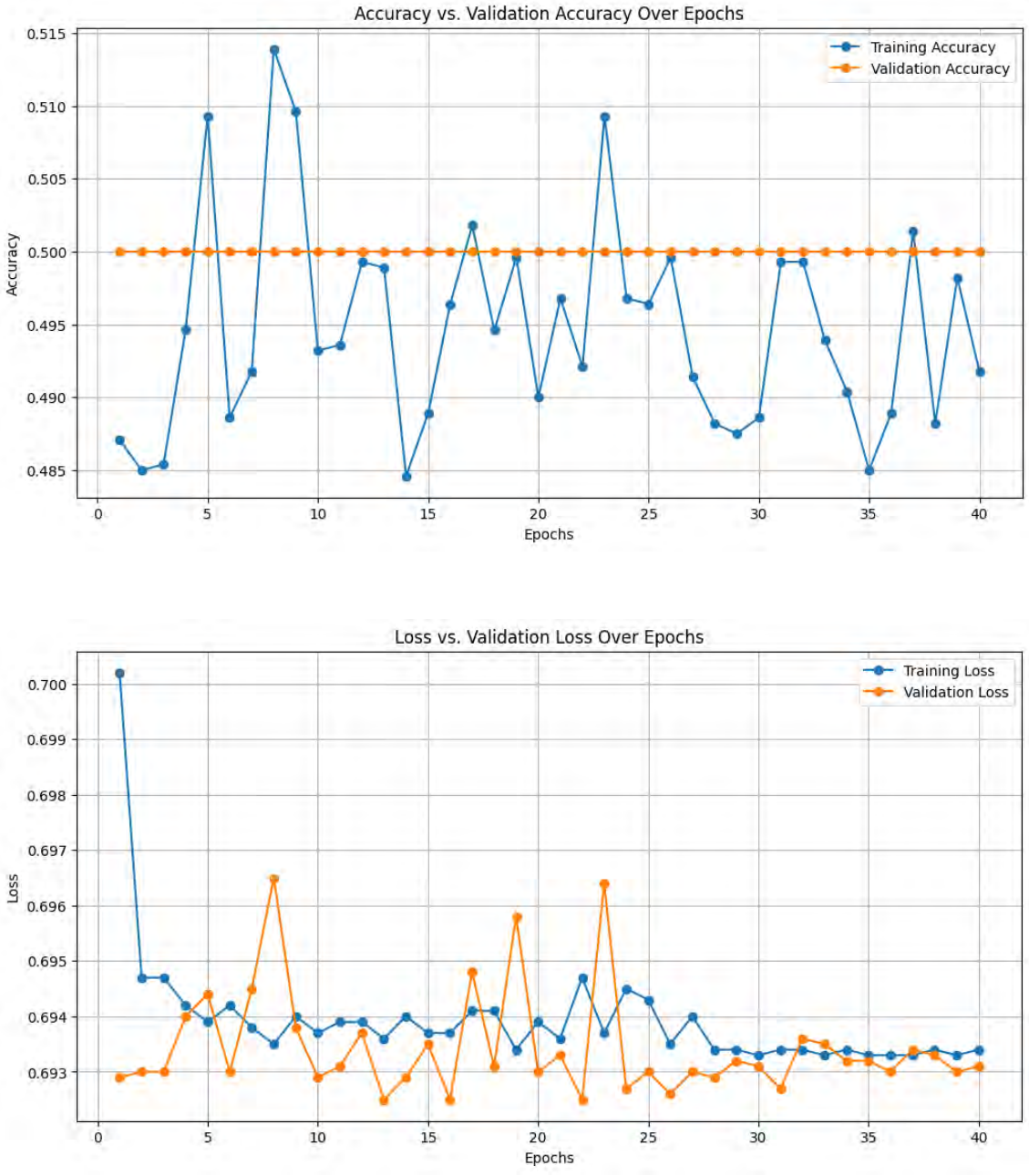


Figure 6.15: AlexNet Training and Validation (Main Dataset)

The training accuracy was able to show recurrent variations according to the graph including some portions where it gave constant values for straight 10 epochs. The accuracy could only reach 51.96% when trained on the secondary

dataset. Furthermore, the validation accuracy seemed to be constant most of the time at 50.16% with close to no fluctuations. On the contrary, The training and validation loss graphs obtained more fluctuations throughout the 40 epochs reaching 0.693 and 0.694 respectively.

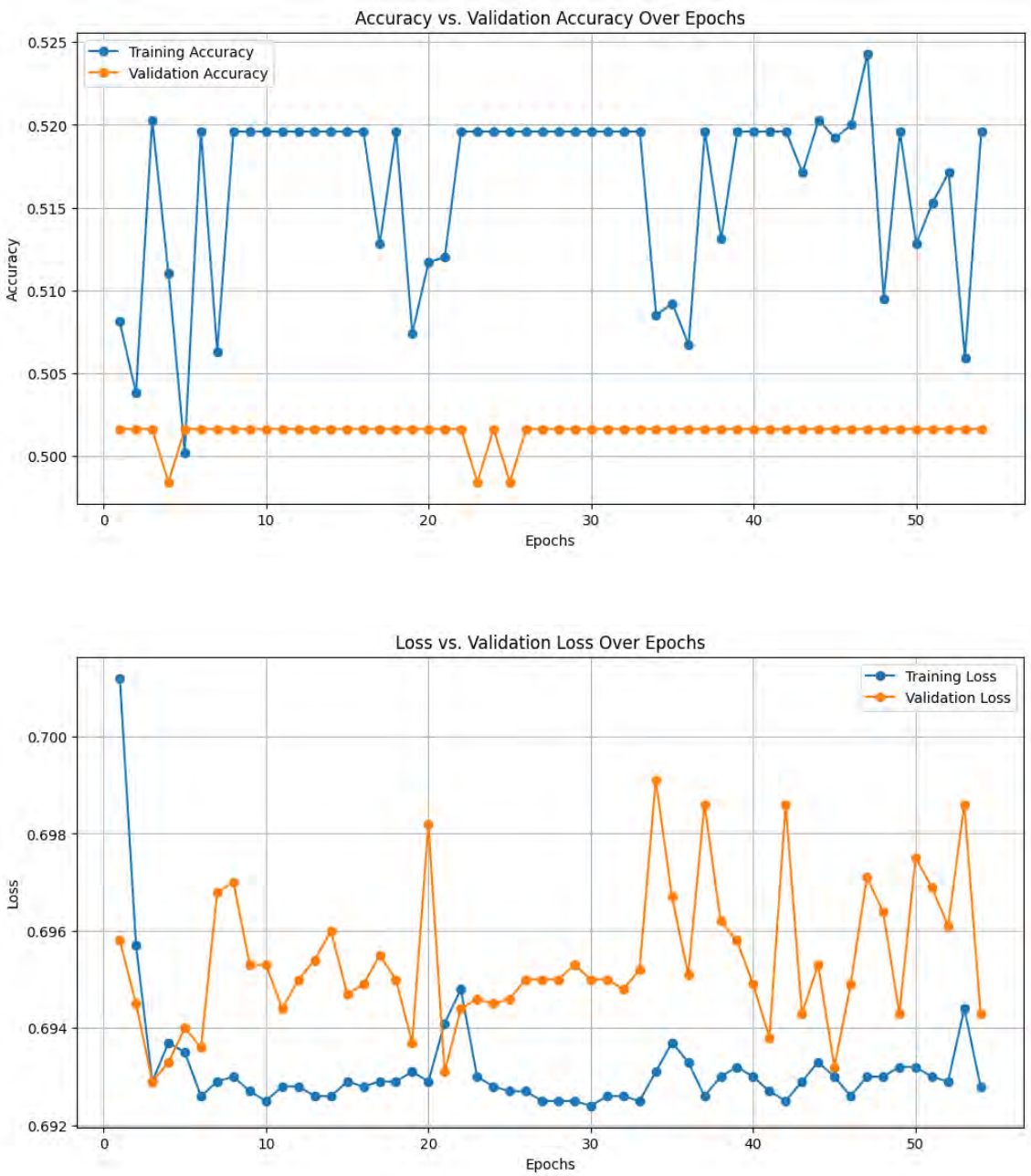


Figure 6.16: AlexNet Training and Validation (Secondary Dataset)

6.3 Model Comparison with Related Works

As a crucial part of our research methodology and performance analysis of our proposed custom CNN model, we have compared our result with the related works we have presented in the section Literature Review. One of our main motives is obtaining a better result in the scale of accuracy in comparison to the relevant work that has been done until now on detecting early Glaucoma. The combined accuracy of the algorithms HCDR and VCDR is obtained to be 74.2% in the paper [2]. In the paper [4], with the concept of automated feather learning, a model has been proposed called ALADDIN. The accuracy using two different datasets by this model is 83.8%. The accuracy in predicting the incidence of Glaucoma by the diagnostic algorithm model DiagnoseNet afterward improved to 90% [12]. The CNN model in [1] has been built in a more traditional manner and with the least amount of layers which rather was a simpler approach with a very low amount of images. This paper achieved an accuracy of 92.2%.. Lastly, the paper [7] has only displayed the combined accuracy of the pre-trained models VGGNet, GoogLeNet and ResNet which is 88.96%.

Approach	**Method**	**Accuracy**
This Paper	Custom CNN	98.71%
Almazroa et al., 2018 [2]	HCDR and VCDR	74.2%
Chen et al., 2015 [4]	ALADDIN	83.8%
Li et al., 2022 [12]	DignoseNet	90%
Ahn et al., 2018 [1]	Traditional Simplified CNN	92.2%
Joshi et al., 2022 [7]	VGGNet, GoogLeNet and ResNet	88.96%

Table 6.3: Table for Comparison with Related Works

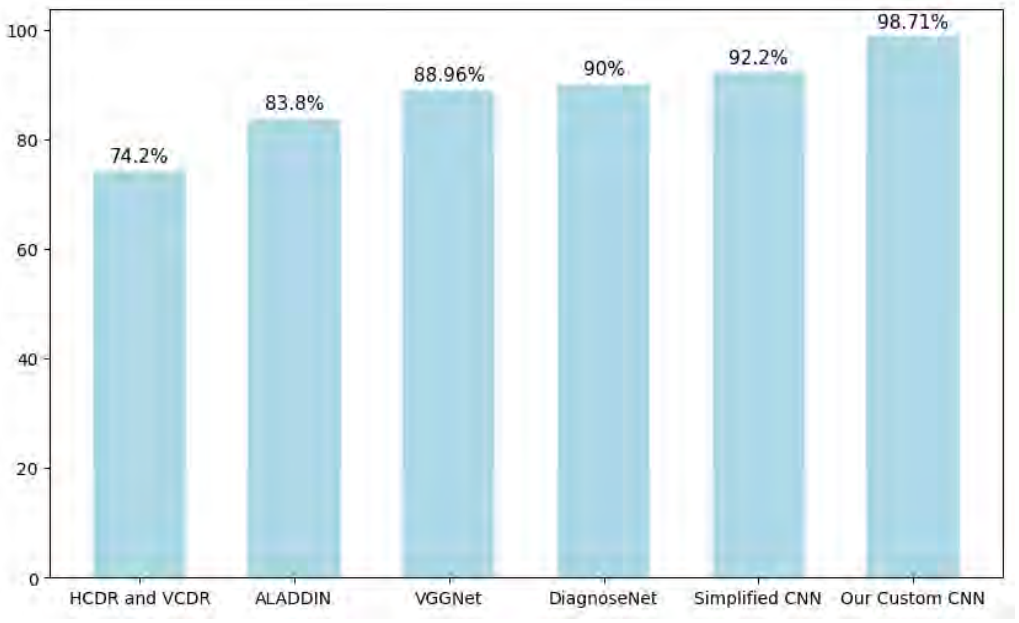


Figure 6.17: Bar Graph for Accuracy of Related Works

Chapter 7

Performance Evaluation

In this chapter, we have tried to analyze the performance of our custom CNN Model by evaluating the classification report, confusion matrix and ROC curve respectively.

7.1 Performance Metrics

The equations of the performance metrics used are specified as follows:

- **Accuracy Formula** :

$$\text{Accuracy} = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.1}$$

- **Precision Formula** :

$$\text{PPV} = \frac{TP}{TP + FP} = 1 - \text{FDR} \tag{7.2}$$

- **Recall Formula** :

$$\text{PPV} = \frac{TP}{TP + FN} = 1 - \text{FDR} \tag{7.3}$$

- **F1-Score Formula** :

$$\text{F1-Score} = 2 \times \frac{\text{PPV} \times \text{TPR}}{P + N} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{7.4}$$

Here, the respective abbreviations are: TP = True Positive, TN = True Negative, P = Positive Case, N = Negative Case, FP = False Positive, FN = False Negative, PPV = Positive Predictive Value, TPR = True Positive Rate, FDR = False Discovery Rate.

Data	**Result Type**	**Class 0**	**Class 1**	**Accuracy**
Test	Precision	0.9870	0.9541	0.97
	Recall	0.9525	0.9875	
	F1-Score	0.9695	0.9705	
Validation	Precision	0.9894	0.9384	0.9625
	Recall	0.9350	0.9900	
	F1-Score	0.9614	0.9635	

Table 7.1: Classification Report of the Custom CNN Model

7.2 Confusion Matrix

The confusion matrix provides a comprehensive summary of the performance of the custom CNN model. We have constructed the confusion matrix heat map for the test and validation data.

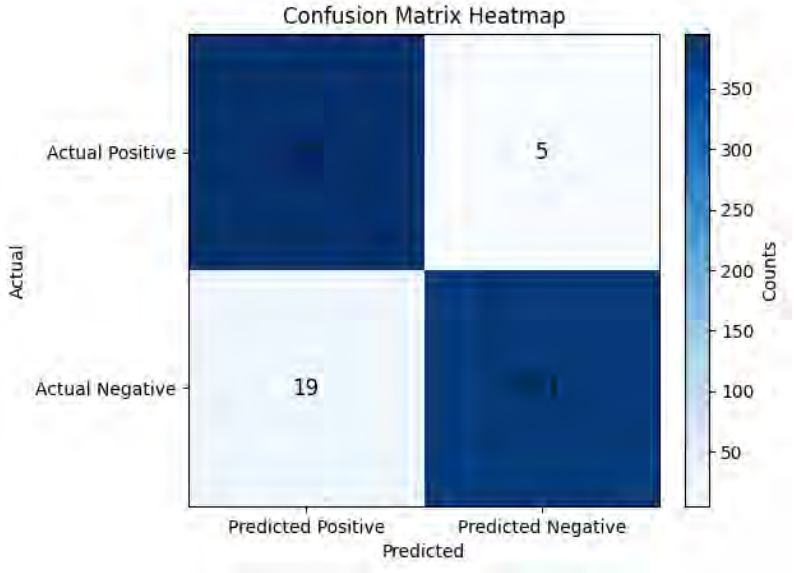


Figure 7.1: Confusion Matrix for Test data

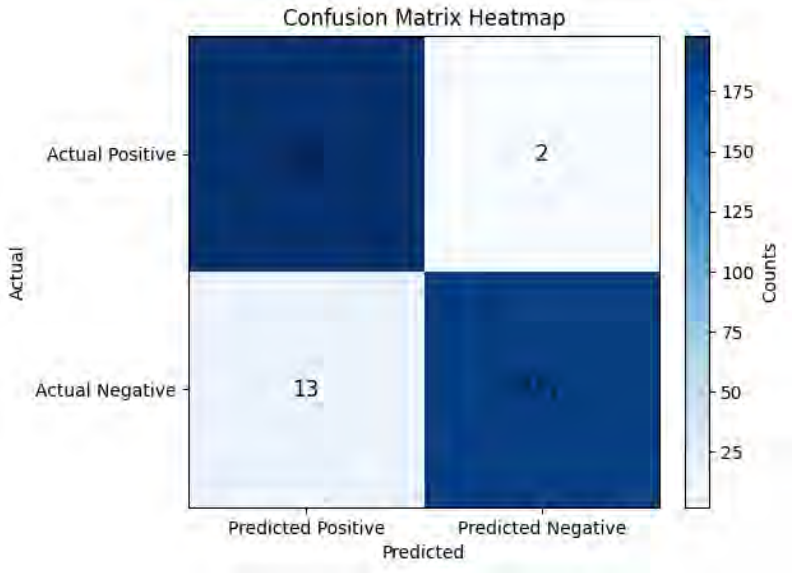


Figure 7.2: Confusion Matrix for Validation Data

7.3 AUC-ROC

AUC-ROC uses a single value to analyze how accurately a binary classification model separates the classes. If the score is closer to 1, it means that the model can identify the classes accurately.

In our custom model, the AUC-ROC score for the test data is 0.9857 and the score for the validation data is 0.9940.

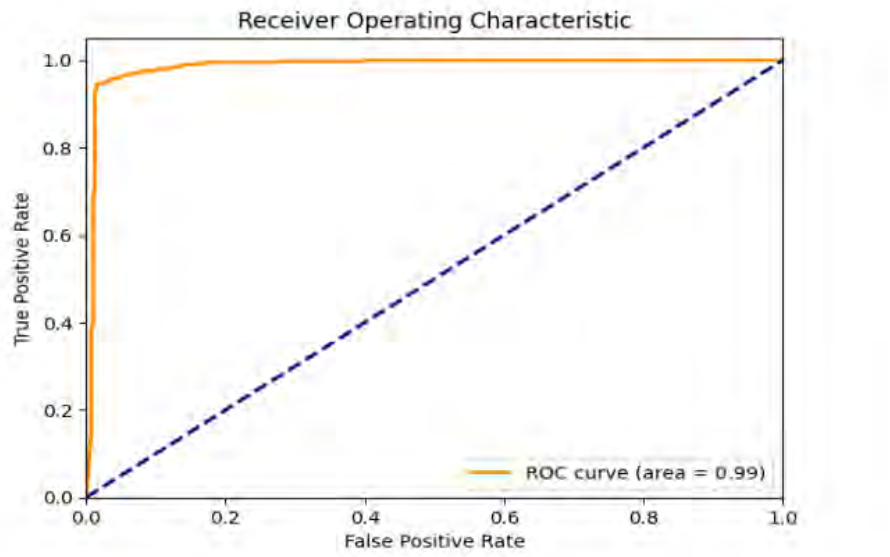


Figure 7.3: ROC for Test data

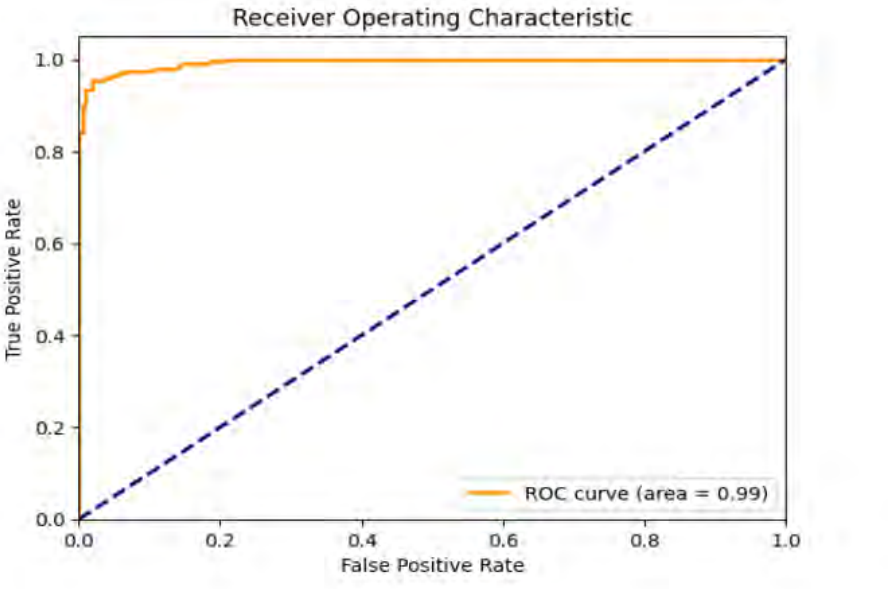


Figure 7.4: ROC for Validation Data

Chapter 8

Limitation and Future Work

In this research, we have proposed a custom model for glaucoma detection using Deep Learning. While the results demonstrate promising advancements in the detection of the disease, there are several areas for further improvement and exploration.

Through the course of our data collection and the development of our model, we faced some constraints we want to look into in the future.

- **Insufficient Dataset:** The availability of a limited amount of data can be a hindrance to the effective performance of the model. Due to privacy and ethical concerns, patient records are highly sensitive. In addition to that, there are institutional barriers that limit the collection of sufficient datasets.
- **Resource Limitations:** The lack of access to high-performance computing resources limits the complexity of the model and impacts its performance.
- **Model Complexity:** The inability to add more parameters and layers may have limited the capacity of the model to capture intricate patterns present in the data.

To address these limitations and pave the way for future enhancements, we need:

- **Collect Larger Datasets from Medical Centers:** For the insufficiency of data, we will be collecting larger and more diverse datasets from different medical centers. Through establishing ethical approval and collaboration with medical professionals for responsible acquisition of dataset.
- **Vision Transformers (ViT):** ViT models have demonstrated effectiveness, particularly with large datasets. ViT is capable of enhancing scalability and overcoming the challenges associated with using merged datasets.
- **Website Implementation for Medical Centers:** We are planning on developing a website that will help medical professionals leverage the capabilities of our model for diagnostic purposes.

Chapter 9

Conclusion

Glaucoma is a chronic eye condition that can lead to complete vision loss if left untreated. To prevent this, our study was directed towards the automated detection of this defect in the early stages through the development and implementation of deep learning models. We tried to come up with an efficient framework to differentiate between images with and without glaucoma and detect them. Furthermore, we tried to come up with a Custom CNN model which would be more feasible and accurate. To that extent, we compared different research papers, pre-trained models and datasets. However, the model must be tested on a larger dataset with more diversity, and evaluated in a clinical setting. The model may further combine with other techniques to achieve even better results. Our study aims to showcase the potential of our approach and it will progressively improve the chances of diagnosis and treatment. Our work focuses on decreasing labor and time consumption while also increasing accuracy as the patterns may be too complex for manual detection. In conclusion, we aim to reduce the development of glaucoma by early detection and treatment.

Bibliography

[1] J. M. Ahn, S. Kim, K.-S. Ahn, S.-H. Cho, K. B. Lee, and U. S. Kim, "A deep learning model for the detection of both advanced and early glaucoma using fundus photography," *PLOS ONE*, vol. 14, 2018. DOI: https://doi.org/10.1371/journal.pone.0207982.

[2] A. Almazroa, S. Alodhayb, K. Raahemifar, and V. Lakshminarayanan, "An automatic image processing system for glaucoma screening," *International Journal of Biomedical Imaging*, e4826385 2018. DOI: https://doi.org/10.1155/2017/4826385.

[3] M. Bajwa, G. Amrit, P. Singh, *et al.*, "G1020: A benchmark retinal fundus image dataset for computer-aided glaucoma detection," *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, Retrieved May 19, 2023. DOI: https://arxiv.org/pdf/2006.09158.pdf.

[4] X. Chen, Y. Xu, S. Yan, D. W. K. Wong, T. Y. Wong, and J. Liu, "Automatic feature learning for glaucoma detection based on deep learning," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham: Springer International Publishing, 2015, pp. 669–677, ISBN: 978-3-319-24574-4.

[5] X. Chen, Y. Xu, D. W. Kee Wong, T. Y. Wong, and J. Liu, "Glaucoma detection based on deep convolutional neural network," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 715–718. DOI: 10.1109/EMBC.2015.7318462.

[6] G. Huang, Z. Liu, L. v. d. Maaten, and K. Weinberger, "Densely connected convolutional networks," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708. DOI: https://doi.org/10.48550/arXiv.1608.06993.

[7] S. Joshi, B. Partibane, W. A. Hatamleh, H. Tarazi, C. S. Yadav, and D. Krah, "Glaucoma detection using image processing and supervised learning for classification," *Journal of Healthcare Engineering*, e2988262 2022. DOI: https://doi.org/10.1155/2022/2988262.

[8] Kaggle, *Glaucoma fundus imaging datasets*. [Online]. Available: https://www.kaggle.com/datasets/arnavjain1/glaucoma-datasets.

[9] R. Kashyap, R. Nair, S. M. P. Gangadharan, M. Botto-Tobar, S. Farooq, and A. Rizwan, "Glaucoma detection and classification using improved u-net deep learning model," *Healthcare*, vol. 10, 2022. DOI: https://doi.org/10.3390/healthcare10122497.

[10] K. E. Kim, J. M. Kim, J. E. Song, C. Kee, J. C. Han, and S. H. Hyun, "Development and validation of a deep learning system for diagnosing glaucoma using optical coherence tomography," *Journal of Clinical Medicine*, vol. 9, 7 2020. DOI: https://doi.org/10.3390/jcm9072167.

[11] Ş. S. Kucur, G. Holló, and R. Sznitman, "A deep learning approach to automatic detection of early glaucoma from visual fields," *PLOS ONE*, vol. 13, 11 2018, e0206081. DOI: https://doi.org/10.1371/journal.pone.0206081Œ.

[12] F. Li, Y. Su, F. Lin, *et al.*, "A deep-learning system predicts glaucoma incidence and progression using retinal photographs," *The Journal of Clinical Investigation*, vol. 132, 11 2022. DOI: https://doi.org/10.1172/JCI157968.

[13] F. Li, Z. Wang, G. Qu, *et al.*, "Automatic differentiation of glaucoma visual field from non-glaucoma visual field using deep convolutional neural network," *BMC Medical Imaging*, vol. 18, 1 2018. DOI: https://doi.org/10.1186/s12880-018-0273-5.

[14] P. Mehta, C. A. Petersen, J. C. Wen, *et al.*, "Automated detection of glaucoma with interpretable machine learning using clinical data and multi-modal retinal images," *American Journal of Ophthalmology*, 2020. DOI: https://doi.org/10.1101/2020.02.26.967208.

[15] A. Saxena, A. Vyas, L. Parashar, and U. Singh, "A glaucoma detection using convolutional neural network," *IEEE Xplore*, 2020, July 1. DOI: https://doi.org/10.1109/ICESC48915.2020.9155930.

[16] S. Serte and A. Serener, "A generalized deep learning model for glaucoma detection," *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2019. DOI: https://doi.org/10.1109/ismsit.2019.8932753Œ.

[17] L. Singh, G. H. Pooja, and et al., "An enhanced deep image model for glaucoma diagnosis using feature-based detection in retinal fundus," *Med Biol Eng Comput*, vol. 59, 2021. DOI: https://doi.org/10.1007/s11517-020-02307-5.

[18] S. Sreng, N. Maneerat, K. Hamamoto, and K. Y. Win, "Deep learning for optic disc segmentation and glaucoma diagnosis on retinal images," *Applied Sciences*, vol. 10, 14 2020, 4916. DOI: https://doi.org/10.3390/app10144916.

[19] C. Szegedy, V. Vanhoucke, J. Ioffe Sergey andShlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1096–1105. DOI: https://doi.org/10.48550/arXiv.1608.06993.

[20] N. Tahira, I. Aun, and S. Valery, "Optic disc and optic cup segmentation for glaucoma detection from blur retinal images using improved mask-rcnn," *International Journal of Optics*, vol. 2021, ID-6641980 2021. DOI: https://doi.org/10.1155/2021/6641980.