



DEVELOPMENT OF GLAUCOMA DETECTION SYSTEM USING CNN AND SVM

N. S. Okomba¹, S. A. Adedayo¹, C. V. Aviara¹, A. O. Esan¹, B. Omodunbi¹, and
A. S. Chikezie²

¹Department of Computer Engineering, Federal University Oye-Ekiti, Ekiti State, Nigeria

²Department of Computer Engineering, University of Uyo, Akwa Ibom State, Nigeria

*Corresponding author's email address: nnamdi.okomba@fuoye.edu.ng

ARTICLE INFORMATION **ABSTRACT**

Submitted 11 November, 2023

Revised 23 February, 2024

Accepted 27 February, 2024

Keywords:

Neural Network

Features

Glaucoma

Vector Machine

Preprocessing

Convolution

Glaucoma is an eye illness that began as a result of high intraocular pressure and resulted in total blindness at its advanced stage. It is a chronic eye disease caused by the damage of the optic nerve found at the back of the eye and will lead to vision loss. Abnormality in the drainage system of the eye causes fluid buildup that excessively triggers pressure leading to optic nerve damage. The development of Glaucoma detection system using Convolutional neural network (CNN) and Support vector machine (SVM) has been extensively studied in recent years. Many studies have shown that CNNs can accurately detect glaucoma from fundus images, optic nerve head images, and other imaging modalities. To implement CNN and SVM techniques for glaucoma detection, the process typically involves data collection, preprocessing, feature extraction, model training, validation, and testing. MobileNetV2 model and combined SVM-CNN approach was introduced in the research. As the Glaucoma Detection System was constructed using the MobileNetV2 CNN architecture which integrates multiple architectural elements to achieve optimal classification performance. Pre-trained weights were employed from the image data set while transfer Learning with MobileNetV2 approach was applied to empower the model with strong feature extraction capabilities. Global Average Pooling layers were appended to the base architecture, followed by dense layers for classification. These dense layers utilized Rectified Linear Unit (ReLU) activation and Dropout regularization to enhance model generalization. As a result, a metric evaluation for detecting Glaucoma Eye disease using MobileNetV2 and SVM offered a mean sum of 75% accuracy, 92% precision, 55% recall, 84% AUC and 34% F1 score. In summary, while SVM excels in precision, MobileNetV2 demonstrates better recall, AUC, and overall balanced performance. The choice of model depends on the specific goals and priorities of the glaucoma detection application, such as whether avoiding false positives (precision) or detecting as many true positives as possible (recall) is more critical.

1.0 Introduction

Glaucoma is an eye optic nerve illness that begins as a result of high intraocular pressure which results to the destruction of the retina neural fiber layer causing total blindness at its advanced stage (Setina et al., 2021; Asaoka et al., 2016). It is classified into two main categories which are the open angle glaucoma and the angle closure glaucoma. Both categories are characterized by damage to the optic nerve through the iris, drainage angle, and the trabecular meshwork (Dinial et al., 2019). Open-angle glaucoma (OAG) is caused by poor outflow of the drainage

canals in the eye, resulting in increased pressure. OAG is the most common form of glaucoma which has no symptoms but develops over a number of years, and can cause gradual and permanent vision loss. Closed-angle glaucoma (CAG), the less common form of the two is caused by a mechanical obstruction in the drainage system inside of the eye, resulting in a sudden rise in pressure. This type of glaucoma can develop very quickly and is then referred to as acute narrow angle glaucoma. With this type of glaucoma there are noticeable symptoms of acute eye pain, nausea, decreased or blurry vision, headache, and/or eye redness. It can result in rapid damage to the eye from very high eye pressure and requires emergency medical attention in order to preserve sight in the eye. When the outer edge of the iris blocks fluid from draining out of the front of the eye, the fluid quickly builds up, causing sudden increase in eye pressure that pushes on the optic nerve leading to blindness. Glaucoma damages the optic nerve head thereby affecting the retina, the difference in the produced and drained range of intraocular fluid (IOF) of the eye results in IOP which in turn affects the nerve fibers (NF) (Thakur et al., 2021). The damaged NF disturbs the retinal nerve fiber layer (RNFL) and causes increase in the cup-to-disc ratio (CDR) and optic nerve head (ONH) (Ali and Sertan, 2019). While prompt glaucoma screening and treatment can prevent patients from losing all vision, screening steps depends highly on professionals who manually analyze the retinal samples, identifying glaucoma affected areas (Jipeng et al., 2020). Lack of professional resource persons and complex glaucoma analysis steps causes delay in predicting the illness, leading to high rate of vision loss.

Tham et al. (2014) provided an overview of the global prevalence of glaucoma and the projected burden of the disease through the year 2040. They discussed the different types of glaucoma and their prevalence in different regions of the world. Certain acknowledged image processing techniques for glaucoma detection were used in proposing location of fatal illness. The research presented the number of people affected by primary open angle glaucoma (POAG) and primary angle closure glaucoma (PACG). Prevalence population-based studies on glaucoma was studied up to 2023, and hierarchical Bayesian method was applied in estimating pooled glaucoma prevalence.

Annan et al. (2016) worked on automated glaucoma detection. This was achieved by using deep convolutional neural network gotten from large scale generic dataset to represent the visual appearance, while combining the holistic and local features to mitigate the influence of misalignment. An area under the receivers' operational characteristics curve of 0.8384 on the Origa dataset was realized by the proposed approach, which showed high efficiency.

Xiangyu et al. (2015) worked on the development of deep learning architecture with convolutional neural network for automated glaucoma diagnosis. The proposed deep learning architecture had six layers of learning, four convolution layers, and two fully connected layers. Glaucoma diagnosis performance was boosted by applying dropout and data augmentation techniques. Origa and SCES datasets were used for experimentation. Actualized results depicted area under curve of receiver operational characteristics curve in glaucoma detection at 0.831 and 0.887 in both datasets.

Kim et al. (2020) described the problem of glaucoma diagnosis when Optical Coherence Tomography (OCT) images are used, which is often challenging due to the subtle change occurring at retina nerve fiber layer and the optic nerve head. They noted that accurate and

timely diagnosis is crucial for preventing vision loss in glaucoma patients. A deep learning architecture was developed using inception v3 in order to predict vision fields with the aid of optical coherence tomography (OCT) imaging. Two OCT images were combined and a CNN architecture was created to predict vision fields with the aid of the combined OCT image. The performance evaluation of the system was carried out by calculating the root mean square error (RMSE) between the actual and predicted visual fields. The RMSE for all patients were 4.79 ± 2.56 dB, with 3.27 dB and 5.27 dB for the normal and glaucoma groups, respectively. The RMSE of the macular region (4.40 dB) was higher than that of the peripheral region (4.29 dB) for all subjects. In normal subjects, the RMSE of the macular region (2.45 dB) was significantly lower than that of the peripheral region (3.11 dB), whereas in glaucoma subjects, the RMSE was higher (5.62 dB versus 5.03 dB, respectively). The deep learning method effectively predicted the visual field 24–2 using the combined OCT image. This method may help clinicians determine visual fields, particularly for patients who are unable to undergo a physical visual field exam.

Ali and Sertan (2019) worked on early glaucoma detection using convolution neural network (CNN). Fundus pictures were used to demonstrate right on time recognition of glaucoma. They trained and furnished deep convolution neural network algorithms comprising of ResNet-50 and GoogleNet while applying transfer learning as classifier. Evaluation of the system performance showed that GoogleNet model performs better than the ResNet-50 in the early and advanced glaucoma detection.

WeiLu et al. (2018) applied artificial intelligence in ophthalmology. They monitored the use of AI to diagnose visually impaired persons putting into consideration glaucoma, age related, and waterfall impairments. They made a presentation on the basics that contributed to the workflow for creating an artificial intelligent model and also carried out a systematic approach towards reviewing applications of AI in human eye ailment diagnosis.

Nooshin et al. (2019) worked on a profound multitasked learning system for interpreting glaucoma detection. The system comprises of segmentation and prediction modules. The optic disc and optic cup regions in a fundus image were located by the segmentation module in order to solve the problem of limitation in clinical interpretability. The prediction module improves segmentation task performance with the aid of large data set, thus lessening the gravity of limited label data in segmentation module. Both components were integrated into a multitask framework permitting end to end training. System evaluation showed the effectiveness of interpretable glaucoma detection in achieving great results in glaucoma screening.

Mijung et al. (2019) worked on computer aided diagnosis and glaucoma localization with the use of deep learning. They analyzed fundus images using convolutional neural network (CNN) and gradient-weighted class activation mapping (Grad-CAM), respectively. Different predictive models were built and evaluated using large set of fundus images, while presenting a web app for computer aided diagnosis and localization of glaucoma with the predictive model integrated at the back-end.

Hassan et al. (2017) worked on creamer significant learning on a single wide field optical adequacy tomography which intelligibly classified glaucoma partners. They worked on the performance of a hybrid deep learning method (HDLM) combine with a single wide-field OCT protocol on eyes initially classified as having mild glaucoma. CNN was used to capture features

from maps gotten from the scan, while a model was trained based on these features using random forest classifier in order to predict the presence glaucoma. The algorithm was compared with optical coherence tomographic (OCT) and visual fields (VFs) metrics. The HDLM protocol performed better than OCT and VF metrics, while distinguishing healthy suspect eye, and eye with early glaucoma. Diagnosing glaucoma using optical coherence tomography (OCT) images is often challenging due to the subtle structural changes in the optic nerve head and retinal nerve fiber layer. OCT utilizes light waves, of which the media opacities interfere with optimal imaging resulting in limitation of OCT performance (Kim et al., 2020). Support Vector Machine have shown good performance in the interpretation of OPT but have limitations when dealing with large data set (Abhishek and Samir, 2016). Studies have demonstrated that using DL for interpreting OCT is efficient, accurate, and with good performance for discriminating glaucomatous eyes from normal eyes, suggesting that incorporation of DL technology in OCT for glaucoma assessment could potentially address some gaps in the current practice and clinical workflow (An et al., 2020). In view of this, a glaucoma detection system using MobileNetV2 CNN and Support vector machine is developed simultaneously for evaluation and comparison

2. Materials and Methods

MobileNetV2 and SVM were used to separate highlights in the data with no complex pre handling, this is achieved using the Depthwise separable convolution technique of MobileNetV2 to reduce the computational cost of convolutions by splitting the computation into depthwise convolution and pointwise convolution. Depthwise convolution applies a single convolutional filter per each input channel and pointwise convolution is used to create a linear combination of the output of the depthwise convolution. The Radial Basis Function Kernel of the SVM is then used to take the data points to a higher dimension where they are linearly separable before classification. When this procedure is combined with move learning and tweaking parameters, then a quality cutting edge technique

is actualized. The developed system was designed using MobileNetV2, SVM, and CNN-SVM which are more adapting and used regularly. The design architecture of MobileNetV2 consists of a series of convolutional layers, followed by depthwise separable convolutions, inverted residuals, bottleneck design, linear bottlenecks, and squeeze-and-excitation (SE) blocks, while CNN requires hidden layers, max pooling layer, and kernel. Data acquisition was performed through the aid of Kaggle online data source, while depthwise separable convolution technique of MobileNetV2 and Radial Basis Function Kernel of the SVM serves as pre-processing tools. The mobileNetV2 was equipped with knowledge by preparing the data and loading a pre-trained model while stacking the classification layer on top. The model is trained, evaluated. And finetuned to increase accuracy. Training the CNN requires the neural network to be fed with large dataset labelled with their corresponding class label MobileNetV2 feature extraction module was applied and the evaluation of the system was carried out using standard metrics of Accuracy, Precision, Recall, AUC, and F1 Score. The flowchart diagram for glaucoma detection system is shown in Figure 1

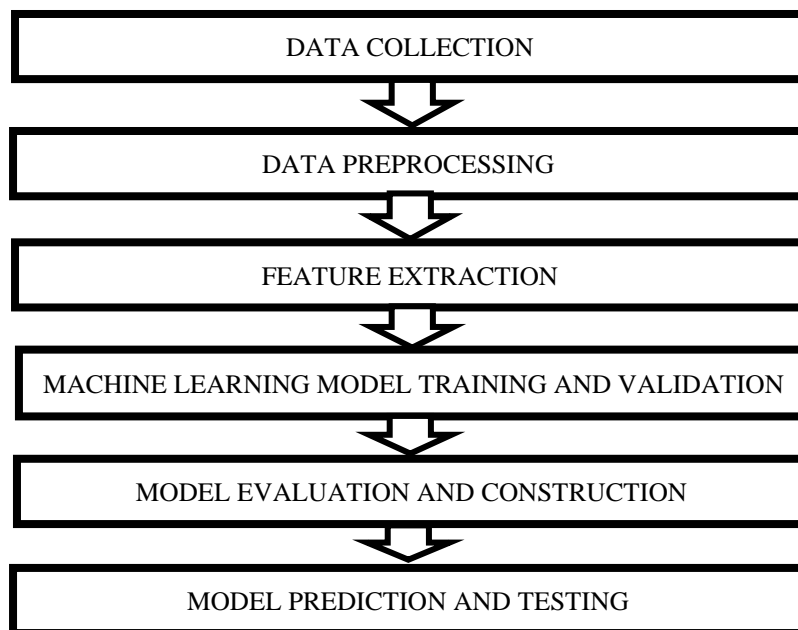


Figure 1: Flowchart Diagram for Glaucoma Detection

2.1 Methodology For building Glaucoma Detection System Using CNN Mobilenetv2

There are several approaches and methodologies that can be used for the detection of glaucoma using CNN. Here is a general outline of the research methodology for early detection of glaucoma which is essential to prevent irreversible vision loss. In this methodology, Specification of the steps taken to develop an accurate Glaucoma Detection System using the MobileNetV2 Convolutional Neural Network (CNN) architecture is outlined. MobileNetV2 is primarily designed for efficient image classification tasks. While it can be used as a backbone network for various computer vision tasks, including object detection, it is specifically tailored for detecting medical conditions like glaucoma. This approach involves data preprocessing, model construction, training, evaluation, and visualization. Figure 2 shows the Block diagram of Convolutional Neural Network (CNN) using MobileNetV2.

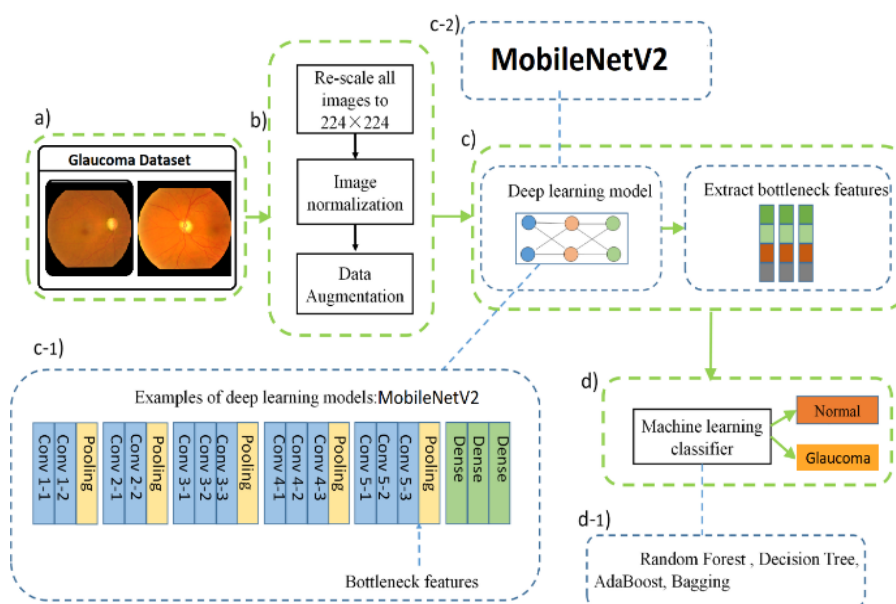


Figure 2: Block diagram CNN using MobileNetV2

2.1.1 Data Collection

Large datasets of digital images of the retina and optic nerve head from both healthy and glaucomatous eyes were collected from Kaggle online data source. These images can be obtained using optical coherence tomography (OCT) or fundus photography for data acquisition and preparation. A standardized multichannel dataset for glaucoma which is a collection of 19 public datasets comprising full fundus glaucoma images, and associated image metadata was used for the development of this system. The data set was retrieved from the specified link online; (<https://www.kaggle.com/datasets/deathtrooper/multichannel-glaucoma-benchmark-dataset/data>.)

2.1.2 Data Preprocessing

The dataset was loaded from a CSV file containing metadata. The metadata was extracted from the CSV file, providing essential information about each image. The dataset was cleaned by filtering out incomplete or irrelevant entries. To facilitate seamless integration with image files, a new column was created by appending '.png' extensions to image filenames. Data columns were appropriately converted to their required data types for further processing in order to correct variations in illumination, contrast, and other artifacts that can affect image quality. This step may include denoising, normalization, and segmentation of the retinal layers.

2.1.3 Feature Extraction

A balanced distribution of data across subsets was pivotal for effective model training and this is achieved by calculating the number of healthy eye images and glaucoma infected eye images in the dataset while MobileNetV2 was used as a feature extractor which composed of bottleneck layers, with additional convolutional layer used to deepen the neural network to increase detection accuracy. The preprocessed images were passed through the network while the output from the intermediate layer serves as the feature vector. These features would capture various patterns and structures present in the images.

2.1.4 Data Visualization:

For efficient model training and augmentation, the capabilities of Image Data Generators were harnessed. These generators preprocess and augment image data, while also preserving class balance. Moreover, dedicated visualization generators were developed to showcase sample images from the healthy and glaucoma categories. A custom visualization function facilitates the presentation of these samples for insightful analysis. It permits creation of own data visual logic to draw results not covered by built in library of standard charts. The improvise visualization tool was applied for this purpose. Improvise is a visualization system that mainly supports coordinated visualizations. It provides primitive and specialized properties whose visual properties can show data using expressions. The expressions can be conditional, logical, or mathematical. Improvise provides specialized objects that support complex layouts such as trees. It was used in the accomplishment of visual mappings by navigating from panel to panel in a development environment to create a visualization. Since each panel has a distinct purpose, one panel shows the available visual objects and their properties, another panel shows the variables that can be used in expressions. x-y graph was defined using Plane View 2D object from the list of visual objects. visual mappings for the visual object were created using Layer Projection from the properties list, this leads to a new panel where expressions are defined. Improvise shows the result as a conditional expression tree with default. This expression was built step-by-step using combo boxes that provide the available expression elements,

Conditional statement part was manipulated by clicking the tree nodes. Function from the Category combo box were used to create a comparison and conditional part of the expression. Improve displays the available fields to enable selection of options. Visual mappings rely heavily on dialogues. The environment enables the use of combo-boxes that have the expression elements (Kostas et al., (2013).

2.2 Construction of Mobilenetv2 CNN

The Glaucoma Detection System was constructed using the MobileNetV2 CNN architecture. The MobileNetV2 model integrates multiple architectural elements to achieve optimal classification performance. The architecture of MobileNetV2 consists of a series of convolutional layers, followed by depthwise separable convolutions, inverted residuals, bottleneck design, linear bottlenecks, and squeeze-and-excitation (SE) blocks. These components work together to reduce the number of parameters and computations required while maintaining the model's ability to capture complex features. Depthwise separable convolution was used to reduce the computational cost of convolutions. It separates the standard convolution into two separate operations: depthwise convolution and pointwise convolution. This separation significantly reduces the number of computations required, making the model more efficient. Inverted residuals were used to improve the model's accuracy by introducing a bottleneck structure that expands the number of channels before applying depthwise separable convolutions. This expansion allows the model to capture more complex features and enhance its representation power. The bottleneck design was applied to further reduce the computational cost by using 1×1 convolutions to reduce the number of channels before applying depthwise separable convolutions. This helps maintain a good balance between model size and accuracy. Linear bottlenecks are introduced to address the issue of information loss during the bottleneck process. By using linear activations instead of non-linear activations, the model preserves more information and improves its ability to capture fine-grained details. Squeeze-and-excitation (SE) blocks are added to MobileNetV2 to enhance its feature representation capabilities. These blocks adaptively recalibrate the channel-wise feature responses, allowing the model to focus on more informative features and suppress less relevant ones.

2.2.1 Transfer Learning with MobileNetV2: Leveraging the MobileNetV2 architecture, pre-trained weights were employed from the ImageNet dataset. This transfer learning approach empowered the model with powerful feature extraction capabilities. This process was carried out by creating a dataset from a directory, preprocess and augment data using the Sequential API, adapt a pretrained model to new data and train a classifier using the Functional API and MobileNet, then fine-tune the classifier's final layers to improve accuracy. The pre-trained model is a network that's already been trained on a large dataset and saved, which allows to be used to customize an own model cheaply and efficiently. MobileNetV2, was designed to provide fast and computationally efficient performance. It's been pre-trained on ImageNet, which is a dataset containing over 14 million images and 1000 classes.

2.2.3 Global Average Pooling and Dense Layers: Global Average Pooling layers were appended to the base architecture, followed by dense layers for classification. These dense layers utilized ReLU activation and Dropout regularization to enhance model generalization, they are responsible for reducing the spatial dimensions of the input data, in terms of width and height, while retaining the most important information. This enables the generation of one

feature map for each corresponding category of classification task in the last mlpconv layer. Instead of adding fully connected layers on top of the feature maps, the average of each feature map is taken, and the resulting vector is fed directly into the softmax layer.

2.3 Model Compilation, Training, And Evaluation

The performance of the trained machine learning models was evaluated on a separate test dataset using metrics such as sensitivity, specificity, and accuracy (Rarasmaya et al., 2022). This step helped to assess the model's ability to detect glaucoma accurately. Creating an effective MobileNetV2-based Glaucoma Detection System requires meticulous training and evaluation procedures. MobileNetV2 was trained on ImageNet and is optimized to run on mobile and other low-power applications. It is 155 layers deep and very efficient for object detection and image segmentation tasks, as well as classification tasks. The architecture has three defining characteristics which are depthwise separable convolutions, thin input and output bottlenecks between layers, and Shortcut connections between bottleneck layers. MobileNetV2 uses depthwise separable convolutions as efficient building blocks. Traditional convolutions are often very resource-intensive, and depthwise separable convolutions are able to reduce the number of trainable parameters and operations and also speed up convolutions in two steps, the first step calculates an intermediate result by convolving on each of the channels independently. This is the depthwise convolution, secondly, another convolution merges the outputs of the previous step into one. This gets a single result from a single feature at a time, and then is applied to all the filters in the output layer. This is the pointwise convolution.

2.3.1 Optimizers and Loss Function

The model was compiled with the 'adam' optimizer and binary cross-entropy loss. These choices aligned with the binary classification nature of our problem Adaptive Moment Estimation is an iterative optimization algorithm which was used to minimize the loss function during the training of mobileNetV2 model. It is a combination of RMSprop and Stochastic Gradient Descent with momentum (Diederik and Jimmy, 2015).

2.3.2 Standard Metrics

Standard metrics was introduced, including precision, recall, and F1 score, to comprehensively evaluate model performance. Precision and recall are two evaluation metrics used to measure the performance of a classifier, precision measures the accuracy of positive predictions and can be seen as a measure of quality. Higher precision means that the algorithm returns more relevant results than irrelevant ones, it is used for machine learning models where low False Positive (FP) is important such as spam detection. Recall measures how often the model correctly identifies positive instances from all the actual positive samples in the dataset, it shows the ability of a model to find all the relevant cases within a data set. Mathematically, it is the number of true positives divided by the number of true positives plus the number of false negatives. It provides an overall measure of how often the model is correct, regardless of whether the instance is positive or negative. The F1-score specifies the weighted average of both precision and recall depending on the weight function. It shows the harmonic mean between precision and recall, and uses recall to get the fraction of true positive records among the total of actual positive records. Early Stopping and Tensor Board callbacks were integrated to optimize training efficiency and monitor progress (Hossin and Sulaiman, 2015).

2.3.3 Training and Validation

model training was initiated using the training data generator, with simultaneous validation using the validation generator. This iterative process fine-tunes the model's parameters for optimal performance. The system was implemented using python programming language, and the model trained with PyTorch framework. The data was downloaded using Roboflow and converted into a Tensorflow ImageFolder Format. The pre-trained model is loaded and the classification layers stacked on top. The model was trained using the PyTorch framework on a Colab GPU interface, which allocated a 12GB NVIDIA Tesla K80 GPU. The pre-trained MobileNetv2 used to train a model for glaucoma detection was modified by the replacement of the number of output classes in the last fully connected linear block with the number of classes in the image dataset. The model was fine tuned to increase accuracy after convergence (Pan and Yang, 2010).

2.3.4 Evaluation

The model's effectiveness was gauged using the validation data. Metrics such as accuracy, precision, recall, Specificity, Receiver Operating Characteristics Area Under Curve (ROC-AUC) were computed, offering a comprehensive view of its capabilities.

2.3.5 Accuracy (A)

This metric was used to generally describe how the model performs in different predictions of Glaucoma detection. Convolutional Neural Networks tend to achieve high accuracy rates due to their ability to learn complex features from images. They can capture intricate patterns in retinal images, leading to accurate glaucoma detection. SVMs also offer good accuracy, but they may not perform as well as CNNs when dealing with complex and highly dimensional data like medical images (Hossin and Sulaiman, 2015).

$$A = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

The measure was computed using the following confusion matrix parameters:

True Positive (TP): the number of infected images classified correctly.

True Negative (TN): the number of images correctly predicted as negative.

False Positive (FP): The number of positive images incorrectly predicted as negative.

False Negative (FN): The number of negative images incorrectly predicted as positive

CNNs can be fine-tuned to optimize sensitivity and specificity based on chosen threshold. This flexibility allowed for adjusting the model's performance to meet specific clinical requirements. SVMs typically provide a good balance between sensitivity and specificity. However, fine-tuning might require more manual intervention compared to CNNs. Sensitivity metrics measures the fraction of positive patterns that are correctly classified. Sensitivity analysis determines how different values of an independent variable affect a particular dependent variable under a given set of assumptions. In other words, sensitivity analyses study how various sources of uncertainty in a mathematical model contribute to the model's overall uncertainty. (Hossin and Sulaiman, 2015)

$$S = \frac{TP}{TP+FN} \quad (2)$$

The measure was computed using the following confusion matrix parameters:

True Positive (TP): the number of infected images classified correctly.

False Negative (FN): The number of negative images incorrectly predicted as positive.

2.3.6 Precision (P)

This was placed as the ratio of number of positively correctly classified samples to number of samples classified as positive. When the system presents more of the incorrect glaucoma detection, the denominator increases and the precision gets small. The reverse is the case when more correct glaucoma is detected.

$$P = \frac{TP}{TP+FP} \quad (3)$$

The measure was computed using the following confusion matrix parameters:

True Positive (TP): the number of infected images classified correctly.

False Positive (FP): The number of positive images incorrectly predicted as negative.

2.3.7 Recall (R)

This was calculated as the ratio of number of positively correct classified glaucoma prediction to the entire glaucoma detection samples. More positive samples are detected at higher recall level.

$$R = \frac{TP}{TP+FN} \quad (4)$$

The measure was computed using the following confusion matrix parameters:

True Positive (TP): the number of infected images classified correctly.

False Negative (FN): The number of negative images incorrectly predicted as positive.

2.3.8 F1 Score (F)

This was calculated using the realized precision and recall from the system. It is the harmonic mean value of the precision and recall.

$$F = \frac{2 \times R \times P}{R+P} \quad (5)$$

The measure was computed using the following parameters:

Precision (P): a measure of the exactness of the model which is the percentage of the positive predicted cases that are true,

Recall (R): a measure of the completeness of the model which is the percentage of positive case correctly identified to all the cases in a class,

2.3.9 AUC-ROC (Area Under the Receiver Operating Characteristic Curve)

CNNs often exhibit high AUC-ROC values, indicating their ability to discriminate between glaucomatous and healthy cases effectively. This is crucial for diagnostic accuracy. SVMs can also achieve good AUC-ROC values but may not consistently outperform CNNs when it comes to capturing complex image patterns. To plot the ROC curve, the TPR and FPR for many different thresholds were calculated. For each threshold, the FPR value in the x-axis and the TPR value in the y-axis were plotted and the dots joined with a line. The True Positive Rate (TPR) and False Positive Rate (FPR) values for each of the thresholds were computed and then plot TPR against FPR. AUC (area under the ROC curve) measures the area lying below the entire ROC curve. It represents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. AUC becomes useful for knowing the average concentration over a time interval, AUC/t . The true positive rate (TPR, also called sensitivity) is calculated as $TP/TP+FN$. TPR is the probability that an actual positive will test positive. The true negative rate (also called specificity), which is the probability that an actual negative will test negative. It is calculated as $TN/TN+FP$.

2.4 Visualization of Training Progress

Visualizing the training process was essential for grasping model dynamics and improvements. Once PyTorch is set up, bring in tensorboard pytorch. Install TensorBoard by using pip, Python's package installer. This step ensures access to all the visualization capabilities offered by TensorBoard during the model training journey. These visualizations provide insights into data flow, activation functions, and how the different model components are interconnected (Dongyu et al, 2018).

2.4.1 Plotting Training Metrics

Matplotlib was employed to generate insightful plots that illustrate the evolution of training and validation accuracy and loss over epochs (Dongyu et al, 2018).

2.4.2 Model Prediction and Testing

The culmination of my effort's manifests in model prediction and rigorous testing.

2.4.3 Generating Predictions

Deploying the trained MobileNetV2 model, generated predictions on the test data. This step unveils the model's real-world performance on unseen instances.

2.5 Methodology for Building Glaucoma Detection System Using SVM

Various methods could be applied in realizing an accurate and timely diagnosis of glaucoma, a leading cause of irreversible blindness, is of paramount importance in preserving vision. Support Vector Machines (SVMs) was used for the detection of glaucoma from medical images, including retinal scans and optic nerve images. SVMs are a type of machine learning algorithm that can be employed for binary classification tasks like distinguishing between healthy and glaucomatous eyes. In this methodology, the steps taken to create an effective Glaucoma Detection System using the Support Vector Machine (SVM) algorithm was outlined. The approach applied involves data preprocessing, model construction, training, and thorough evaluation. Figure 3 shows the Block Diagram Model of Support Vector Machine (SVM)

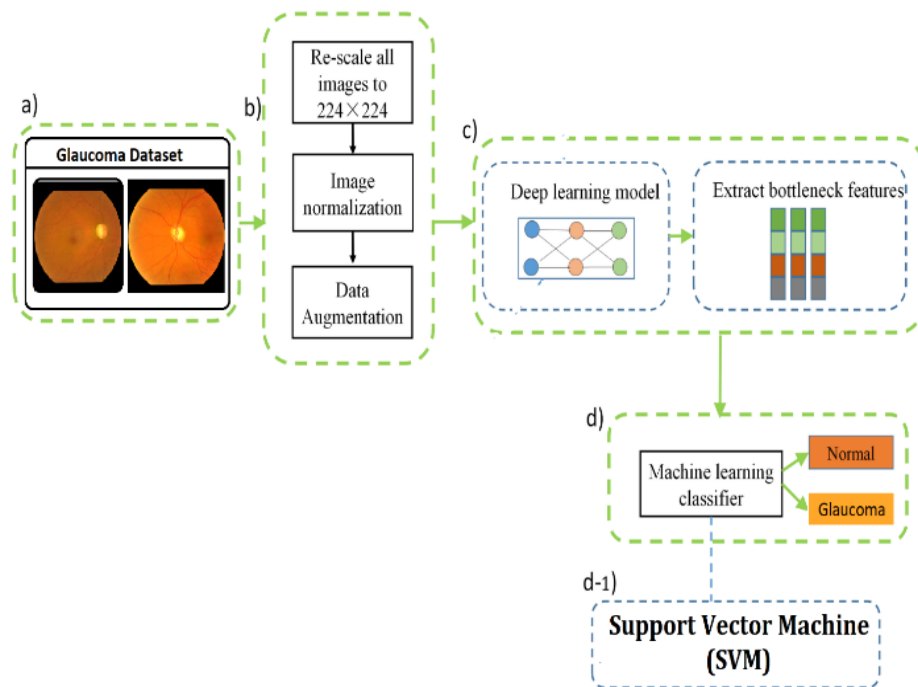


Figure 3: Block Diagram Model of Support Vector Machine (SVM)

2.5.1 Constructing the SVM Model

The SVM-based Glaucoma Detection System was realized through the construction of a sequential model. This model incorporates several crucial components to maximize classification performance. Multiple convolutional layers were implemented, followed by max-pooling operations. These layers were instrumental in extracting discriminative features from fundus images. Batch Normalization was applied to normalize activations, aiding in training stability. The output from the convolutional layers was flattened into a one-dimensional vector, suitable for feeding into dense layers. Dense layers with ReLU activation were introduced to further refine features and enable sophisticated decision making. Dropout layers were strategically placed to counter overfitting concerns. The final output layer was tailored for SVM-based binary classification. L2 regularization was employed to enhance generalization capabilities. Some preprocessing tasks were done on the input images, this includes normalizing the images, colour conversion from RGB to grayscale, resizing the images, removal of noise from the images and contrast adjustment for improving image quality Feature Extraction Using Principal Component Analysis Method was then carried out, followed by Classification using Support Vector Machine (SVM) (Abhishek and Samir, 2016).

2.5.2 Model Training and Evaluation

Creating an effective SVM model involves meticulous training and thorough evaluation. load in the data and separate into training and test sets. The training set will help find a line to separate the people with and without glaucoma, and the test set will say how well the model works on people it hasn't seen before. With the data loaded, one can prepare the model to be fit to the data. SVMs are in the svm module of scikit-learn in the SVC class. SVC stands for Support Vector Classifier and is a close relative to the SVM. After bringing in the SVC class, the model is fit using the age and chol columns from the training set. Using the fit method builds the line that separates those with glaucoma from those without. Once the model has been fit, next is

to predict the glaucoma disease status in the test group, and can compare the model predictions to the actual observations in the test data.

In the pursuit of comprehensive evaluation, custom metrics, and F1 score were defined. Early Stopping and Tensor Board callbacks were integrated to optimize training efficiency and monitor model progress. Early stopping was used as a regularization to avoid overfitting when training a with an iterative method, such as gradient descent. Such methods update the learner so as to make it better to fit the training data with each iteration (Abhishek and Samir, 2016). A callback is passed into the model in the call to *fit* and is called by the method at various points in the training. The callback object can access the model at various time states and with it take action; such as interrupt training, save a model, load a different weight set or alter the state of the Model. TensorBoard allows viewing of the models at a high level, which can be important in debugging and optimizing the model using features like Live graphs of Training and Validation Metrics. Tensorboard, enables visually monitoring metrics during training, visualizing the model architecture, visualizing histograms of activations and gradient, and-exploring embeddings in 3D (Dongyu et al., 2018).

Training was initiated, utilizing the training data generator, while simultaneously validating using the validation generator. This iterative process refines our model's weights and biases. To train the SVM, certain decisions were applied starting from Importing the dataset, Exploring the data to figure out what they look like, Pre-processing the data, Splitting the data into attributes and labels, Dividing the data into training and testing sets, Training the SVM algorithm, make some predictions, and evaluate the results of the algorithm (Abhishek and Samir, 2016).

To assess the model's performance, evaluation was carried out on the validation data. Metrics such as accuracy, precision, recall, AUC, and F1 score were computed and analyzed. The effectiveness of SVM depends on the selection of kernel, kernel's parameters and soft margin parameter C. Each pair of parameters is checked using cross validation, and the parameters with best cross validation accuracy are picked. Another important step in evaluating an SVM model is choosing the right type of model and the appropriate parameters for the problem. There are different types of SVM models, such as linear, polynomial, radial basis function (RBF), and sigmoid, that have different assumptions and characteristics. The SVM classification score for classifying observation x is the signed distance from x to the decision boundary ranging from $-\infty$ to $+\infty$. A positive score for a class indicates that x is predicted to be in that class. A negative score indicates otherwise (Komal et al, 2019).

2.5.3 Visualizing Training Progress

A visual representation of training progress is essential for understanding model dynamics. The visualization process was carried out by importing necessary libraries and loading the dataset, training SVM with linear kernel, creating meshgrid for decision boundary, plotting decision boundary of linear SVM, defining gamma values, and plotting decision boundaries for each gamma value. Matplotlib was employed to generate insightful plots depicting the progression of training and validation accuracy and loss across epochs. The training and validation loss values provide important information because they give us a better insight into how the learning performance changes over the number of epochs and help us diagnose any problems with learning that can lead to an underfit or an overfit model. They will also inform us about the epoch with which to use the trained model weights at the inferencing stage. Computationally,

the training loss is calculated by taking the sum of errors for each example in the training set. It is also important to note that the training loss is measured after each batch. This is usually visualized by plotting a curve of the training loss. This is carried out by importing digit dataset and necessary libraries, importing learning curve function for visualization, splitting dataset into train and test, and plotting graphs using matplotlib to analyze the learning curve (Abhishek and Samir, 2016).

The culmination of our effort lies in the generation of predictions and model testing. Prediction is done with a mapping function which maps independent variables to dependent variable. The mapping function for SVM is a decision boundary which makes the distinction between two or more classes. The model-building steps in prediction that are used iteratively until a desired model has been constructed are data preparation, model selection and data fitting, and model validation. The accuracy of the classification with SVM can be calculated by adding up all the data that are True Positive (TP) and True Negative (TN), then dividing the result of the summation by the total amount of data. A support vector machine (SVM) is a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane that maximizes the distance between each class in an N-dimensional space (Hossin and Sulaiman, 2015).

The trained SVM model was deployed to predict on the test data. This step illuminated the model's actual performance on unseen data. In the realm of medical image analysis, the Glaucoma Detection System using SVM stands as a testament to the synergy of advanced algorithms and meticulous methodologies. To build the predictive model, dataset is collected and organized, it is then cleaned and made available for an algorithm to build the model. Generating the model Firstly, imports the SVM module and create support vector classifier object by passing argument kernel as the linear kernel in SVC () function. Then, fit the model on train set using fit () and perform prediction on the test set using predict (). The support vector uses a mathematical function, often called a kernel function which is a math function that matches the new data to the best image from the training data in order to predict the unknown image label Through systematic data preprocessing, model construction, and intensive training, a model has been built, capable of distinguishing between healthy and glaucoma fundus images.

2.6 Methodology for Building Glaucoma Detection System Using Mobilenetv2 and SVM (CNN-SVM)

There are several approaches and methodologies that can be used for the detection of glaucoma using CNN-SVM. Here is a general outline of the research methodology for Early detection of glaucoma is essential to prevent irreversible vision loss. In this methodology, we outline the steps taken to develop an accurate Glaucoma Detection System using the MobileNetV2 Convolutional Neural Network (CNN) and SVM architecture. MobileNetV2 is primarily excellent at feature extraction from medical images, while SVMs are effective at classification tasks. While it can be used as a backbone network for various computer vision tasks, including object detection, it is specifically tailored for detecting medical conditions like glaucoma. Our approach encompasses data preprocessing, model construction, training, evaluation, and visualization but during the process SVM was used as an output layer from

CNN thereby adding Regularization for SVM. Because when we use Support Vector Machine for binary classification, we use something called Linear SVM.

2.6.1 Construction of MobileNetV2 CNN-SVM

The Glaucoma Detection System was constructed using the MobileNetV2 CNN architecture and SVM architecture as base output layer. This model integrates multiple architectural elements to achieve optimal classification performance. Leveraging the MobileNetV2 architecture, pre-trained weights were employed from the ImageNet dataset. This transfer learning approach empowered my model with powerful feature extraction capabilities. Global Average Pooling layers was appended to the base architecture, followed by dense layers for classification. These dense layers utilized ReLU activation and Dropout regularization to enhance model generalization.

2.6.2 Model Compilation, Training, And Evaluation

The performance of the trained machine learning models was evaluated on a separate test dataset using metrics such as sensitivity, specificity, and accuracy. This step helps to assess the model's ability to detect glaucoma accurately. Creating an effective MobileNetV2-based Glaucoma Detection System requires meticulous training and evaluation procedures. The model was compiled with the 'Adam' optimizer and binary cross-entropy loss. These choices aligned with the binary classification nature of our problem. Custom metrics was introduced, including precision, recall, and F1 score, to comprehensively evaluate model performance. Early Stopping and Tensor Board callbacks were integrated to optimize training efficiency and monitor progress. Model training was initiated using the training data generator, with simultaneous validation using the validation generator. This iterative process fine-tunes the model's parameters for optimal performance. Added output layer model and added (tf.keras.layers.Dense(1, kernel_regularizer=l2(0.01)). Added regularizer for SVM, because when using Support Vector Machine for binary classification, Linear SVM activation='linear')) was applied. softmax was used as activation in the output layer model. The model's effectiveness was gauged using the validation data. Metrics such as accuracy, precision, recall, F1 score, specificity, and Receiver Operating Characteristic Area Under the Curve (ROC-AUC) were computed, offering a comprehensive view of its capabilities.

3. Results and Discussion

The impact of classification model training results showing the comparison of all the chosen evaluation metrics (Accuracy, Precision, Recall, AUC, and F1 Score) are presented in Table 1. Mobile-netV2, and SVM were the Robust deep learning models that were employed in this work. A batch size of 64 and an epoch of 10 were used to train each model. The architectures were fine-tuned specifically for Glaucoma detection by adding additional layers. The models were trained on the training set and evaluated on the validation set. The loss function employed during training was the Adam Optimizer and binary cross-entropy loss function. In the realm of medical image analysis, the MobileNetV2 CNN-based Glaucoma Detection System underscores the potent fusion of advanced neural networks and meticulous methodologies. Our systematic approach to data preprocessing, model construction, intensive training, and holistic evaluation results in a model capable of discerning between healthy and glaucoma fundus images. This journey serves as a testament to the transformative power of deep learning in addressing pressing healthcare challenges.

Table 1: Comparison of the used algorithms

Evaluation Metrics	MobileNetV2 (CNN)	SVM	CNN-SVM
Accuracy	0.7790	0.7090	0.6310
Precision	0.8532	0.9906	0.9507
Recall	0.6740	0.4220	0.2700
AUC	0.8609	0.8081	0.6838
F1 Score	0.3977	0.2721	0.2063

The evaluation metrics used include accuracy, precision and recall, F1 score, and the area under the receiver operating characteristic (AUC-ROC) curve. The metric evaluation for detecting Glaucoma Eye disease using Mobile_NetV2, SVM and CNN-SVM offered a mean sum of 71% accuracy, 93% precision, 46% recall, 78% AUC and 29% F1 score.

Comparison:

Accuracy: MobileNetV2 out performs SVM and CNN-SVM in terms of accuracy, indicating a better overall predictive performance.

Precision: SVM as well as CNN-SVM had higher precision, suggesting that it has a lower false positive rate compared to MobileNetV2.

Recall: MobileNetV2 has higher recall, implying that it is better at identifying actual positive instances compared to SVM and CNN-SVM was lower as well.

AUC: MobileNetV2 has a higher AUC, indicating a better ability to discriminate between the three classes.

F1 Score: MobileNetV2 has a higher F1 score, indicating a better balance between precision and recall compared to SVM and CNN-SVM.

In summary, while SVM and CNN-SVM excels in precision, MobileNetV2 demonstrates better recall, AUC, and overall balanced performance. The choice of model depends on the specific goals and priorities of the glaucoma detection application, such as whether avoiding false positives (precision) or detecting as many true positives as possible (recall) is more critical. The combination of CNN and SVM offers a robust and accurate solution for glaucoma detection. CNNs are excellent at feature extraction from medical images, while SVMs are effective at classification tasks. This dual approach can enhance the reliability of the system. Figure 4 shows the Training and validation loss curves of Mobile-NetV2.



Figure 4: Training and validation loss curves of Mobile-NetV2

The plots on Figure 4 shows the plotting of Loss against Epoch. Epoch means the number of times the dataset training and validation loss curves was done for Mobile-Net V2. The Green color indicates the Training Loss and the Red color indicates Validation Loss. From the plot, it was discovered that while training and validating, the training loss went downwards from epoch 2 to epoch 10. (which means., the training loss was almost zero on epoch 10). While the Validation loss was ranging within sinusoidal waveform between 0.55 to 0.45 loss on the plot. The training loss was better on each epoch than validation loss. Figure 5 shows the Training and validation Accuracy curves of Mobile-NetV2.

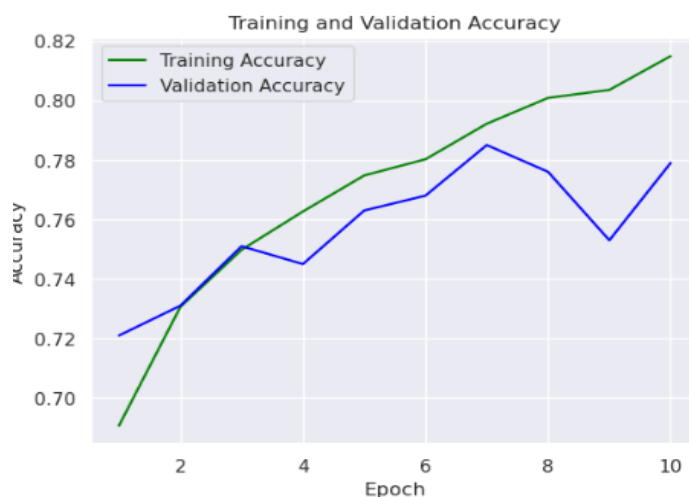


Figure 5: Training and validation Accuracy curves of Mobile-NetV2

This plot on Figure 5 shows the plotting of Accuracy against Epoch, which shows the number of times the dataset training and validation Accuracy curves was done for Mobile-Net V2. The Green color indicates the Training Accuracy and the Blue color indicates Validation Accuracy. From the plot, it was discovered that while training and validating, the training accuracy went upwards from epoch 2 to epoch 10., (high and better accuracy simultaneously on each epoch). While the Validation Accuracy also went upward little bit on each epoch, compared to Training Accuracy. Training Accuracy has a better output than the Validation Accuracy as stated from the plot above. Figure 6 shows the Training and validation loss curves of SVM



Figure 6: Training and validation loss curves of SVM

The plot on Figure 6 shows the plotting of Loss against Epoch, indicating the number of times the dataset training and validation loss curves was done for SVM. The Green color indicates the Training loss and the red color indicates Validation loss. From the plot, it was discovered that while training and validating, the Training loss went downwards from epoch 2 to epoch 10., (the training loss was almost zero on epoch 10). While the Validation loss was ranging within sinusoidal waveform between 0.87 to 0.60 loss on the plot. The training loss was better on each epoch than validation loss. Figure 7 shows the Training and validation accuracy curves of SVM

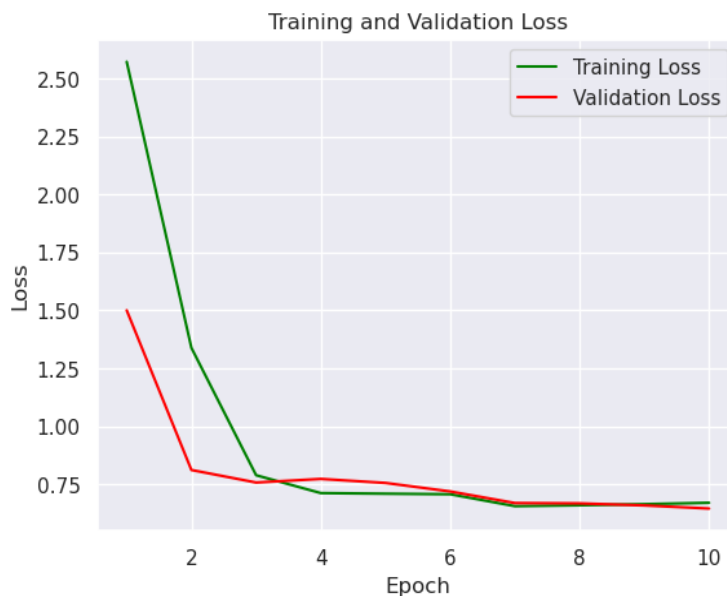


Figure 8: Training and validation loss curves of CNN-SVM

Figure 8 shows a plotting of Loss against Epoch, indicating the number of times the dataset Training and validation loss curves was done for CNN-SVM. The Green color indicates the Training loss and the red color indicates Validation loss. From the plot, it was discovered that while training and validating, the training loss went downwards from epoch 2 to epoch 10., (the training loss was almost zero on epoch 10). While the Validation loss was ranging between 0.82 on epoch 2 and training loss ranging between 1.32 loss on the plot. The training loss was better on each epoch than validation loss but the validation has consistency this time around. Figure 9 shows the Training and validation Accuracy curves of CNN-SVM.

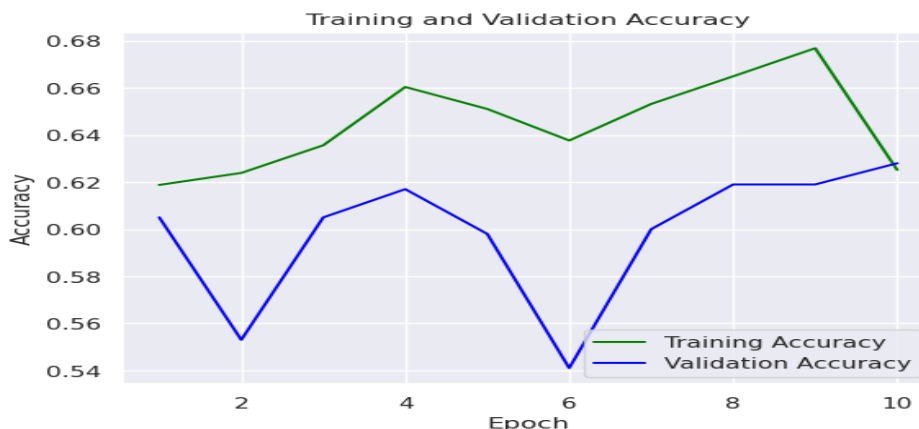


Figure 9: Training and validation Accuracy curves of CNN-SVM

Figure 9 shows the plotting of Accuracy against Epoch, illustrating the number of times the dataset training and validation accuracy curves was done for CNN-SVM. The Green color indicates the Training Accuracy and the Blue color indicates Validation Accuracy. From the plotting, it was discovered that while training and validating, the training accuracy went upwards from epoch 2 to epoch 10, but came down to 62% on the 10th epoch., (good accuracy but not consistent enough). While the Validation Accuracy also went upward little bit on each epoch, came down to 54% accuracy on the 6th epoch and as well got same accuracy percentage of 62%. Training Accuracy has a better output than the Validation Accuracy as stated from the plot above with a 68% Accuracy curve for CNN-SVM on the 9th epoch.

3D BAR CHART COMPARISON BETWEEN MOBILE-NETV2 (CNN), SVM AND CNN-SVM

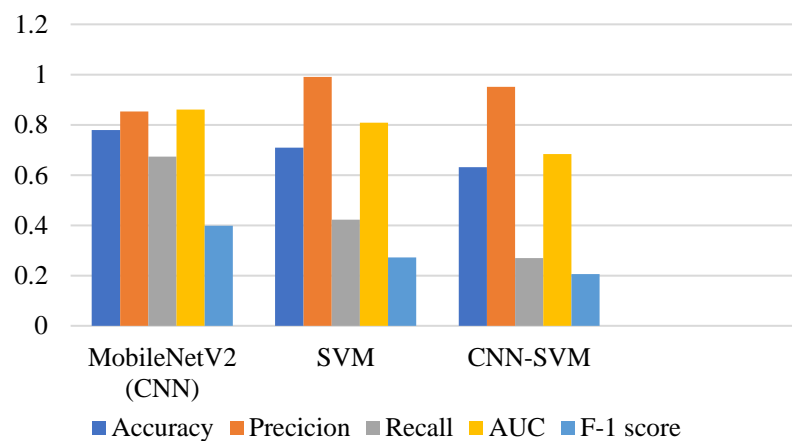


Figure 10: 3D Bar Chart Comparison Between Mobile-Netv2 (CNN), SVM and CNN-SVM

Figure 10 shows a comprehensive bar chart depicting the performance of the MobileNetV2, SVM, and CNN-SVM algorithms used in the glaucoma detection system. This analysis was carried out with the aid of accuracy, precision, recall, AUC, and F1 score as evaluation metrics.

4. Conclusion

In conclusion, the development of a glaucoma detection system using MobileNetV2, SVM and CNN-SVM is a significant advancement in the field of ophthalmology. It has the potential to enhance early detection, reduce human error, and improve patient outcomes. However, it is essential to address challenges related to data quality, interpretability, and integration into clinical practice to realize the full potential of such a system in improving the diagnosis and management of glaucoma. The performance analysis of classification and detective accuracy between MobileNetV2, SVM and CNN-SVM techniques were examined and compared. Both MobileNetV2 and SVMs have their strengths and weaknesses in the context of glaucoma detection; MobileNetV2 excels generally in terms of accuracy and other applied metrics, particularly when ample data is available, and they can adapt to different clinical requirements. SVMs are computationally efficient and offer a reasonable result in precision. CNN-SVM performs poorly considering SVMs deficiency in large data content. MobileNetV2 tends to thrive when trained on large datasets, as they can learn a wide range of features and patterns. With limited data, they may be prone to overfitting. SVMs can perform well with smaller datasets and are often used when data availability is constrained. Finally, the development of a glaucoma detection system using MobileNetV2 and SVMs is a promising approach, and the

choice between these models should be based on specific data, computational resources, and interpretability needs. Combining the strengths of both models and following best practices in data collection, preprocessing, and evaluation can lead to a more robust and effective glaucoma detection system.

References

Ali, S. and Sertan S. 2019. Transfer Learning for Early and Advanced Glaucoma Detection with Convolutional Neural Networks. Medical Technologies National Conference (TIPTEKNO), DOI:10.1109/TIPTEKNO.2019.8894965.

WeiLu, YT., Yue, Yu., Yiqiao, X., Changzheng, C. and Yin, S. 2018. Applications of Artificial Intelligence in Ophthalmology: General Overview. Hindawi Journal of Ophthalmology, 2018(6): 1-15. DOI: 10.1155/2018/5278196

Nooshin, M., Vahid, N., Philip, Y. and Joelle, AH. 2019. Deep Multi-task Learning for Interpretable Glaucoma Detection. IEEE 20th international conference on Reuse and Integration of Data Science, 167-174. doi.org /10.1109/IRI.2019.00037.

Mijung, K., Jong, C., Seung, H., Olivier, J., Sofie, VH., Changwon, K. and Wesley, DN. 2019. Medinoid: Computer-Aided Diagnosis and Localization of Glaucoma Using Deep Learning. Article in MDIP Journal of Applied Science, 9(15): 1-19. <https://doi.org/10.3390/app9153064>

Hassan, M., Thomas, JF., Nicole, C., Carlos, GM., Dana, MB., Jeffrey, ML., Robert, R. and Donald CH. 2017. Hybrid deep learning on single wide-field optical coherence tomography scans accurately classifies glaucoma suspects. National Centre of Biotechnology Information Journal of Glaucoma, 26(12): 1086–1094. DOI: 10.1097/IJG.0000000000000765

Asaoka, R., Murata, A., Iwase, and Araie, M. 2016. Detecting pre-perimetric glaucoma with standard automated perimetry using a deep learning classifier. Ophthalmology, 123(9): 1974–1980. DOI: 10.1016/j.ophtha.2016.05.029

Annan, L., Jun, C., Damon, WK. and Jiang, Liu. 2016. Integrating holistic and local deep features for glaucoma classification. Proceedings of 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC): 1328-1331. DOI:10.1109/EMBC.2016.7590952

Xiangyu, C., Yanwu, X., Damon, WK., Tien, YW. and Jiang, Liu. 2015. Glaucoma detection based on deep convolutional neural network. Proceedings of 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milano, Italy. DOI: 10.1109/EMBC.2015.7318462

Dinial, UN., Handayani, T. and Chastine, F. 2019. Classification of Diabetic Retinopathy and Normal Retinal Images using CNN and SVM. 12th International Conference on Information & Communication Technology and System (ICTS): 152-157. DOI:10.1109/ICTS.2019.8850940

Tham, YC., Li, X., Wong, TY., Quigley, HA., Aung, T. and Cheng, CY. 2014. Global prevalence of glaucoma and projections of glaucoma burden through 2040: a systematic review and meta-analysis. Ophthalmology, 121(11): 2081-2090. DOI: 10.1016/j.ophtha.2014.05.013

Thakur, S., Kumar, P., Kumar, V. and Singh, V. 2021. Glaucoma Detection using Deep Learning Techniques: A Review. International Journal of Engineering and Advanced Technology (IJEAT), 10(1): 784–792.

Kim, J., Lee, J., Sung, K., Lee, K. and Park, J. 2020. A Deep Learning Model to Predict the Presence of Glaucoma Using Optical Coherence Tomography Images. Scientific Reports, 10(1): 1–10.

Setina, J., Khawaja, A. and Weizer, J. 2021. Glaucoma in Adults Screening, Diagnosis, and Management: A Review. Journal of the American Medical Association (JAMA), 325(2): 164–174. DOI: 10.1001/jama.2020.21899.

Jipeng, T., Suma, P. and Manjunath, TC. 2020. Use of Artificial Intelligence & Machine Learning with Deep Learning for Glaucoma Detection in Human Eyes & its Real Time Hardware Implementation. European Journal of Electrical Engineering and Computer Science, 4(2): 1-7. DOI: <https://doi.org/10.24018/ejece.2020.4.2.204>.

Kostas, P., Mohammad, AK., Soren, L. and Shangjin, X. 2013. uVis Studio: An Integrated Development Environment for Visualization, Conference Paper in Proceedings of SPIE - The International Society for Optical Engineering, DOI: 10.1117/12.2003067

Rarasmaya, I., Rika, R. and Wiwiet, H. 2022. Melanoma image classification based on MobileNetV2 network. Sixth Information Systems International Conference (ISICO 2021) Procedia Computer Science, 197(2022): 198–207.

Diederik, PK. and Jimmy, LB. 2015. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. <https://arxiv.org/pdf/1412.6980>

An, R., Clement, CT., Poemen, PC., Ching-Yu, C., Yih-Chung, T., Tyler, HR. and Carol, YC. 2020. Deep learning in glaucoma with optical coherence tomography: A Review. The Royal College of Ophthalmologist, 35: 188–201. <https://doi.org/10.1038/s41433-020-01191-5>

Abhishek, D. and Samir, KB. 2016. Automated Glaucoma Detection Using Support Vector Machine Classification Method. British Journal of Medicine & Medical Research, 11(12): 1-12. DOI: 10.9734/BJMMR/2016/19617

Hossin, M. and Sulaiman, MN. 2015. A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION EVALUATIONS. International Journal of Data Mining & Knowledge Management Process (IJDKP), 5(2): 1-11. DOI: 10.5121/ijdkp.2015.5201

Pan, SJ. and Yang, Q. 2010. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10): 1345–1359, DOI:10.1109/TKDE.2009.191.

Dongyu, L., Weiwei, C., Kai, J., Yuxiao, G. and Huamin, Q. 2018. DeepTracker: Visualizing the Training Process of Convolutional Neural Networks. ACM Transactions on Intelligent Systems and Technology, 10(1): 1–25. DOI: 10.1145/3200489.

Komal, SK., Puneeth, C., Rekha, BS. and Srinivasan, GN. 2019. Performance Evaluation of Support Vector Machines (SVM) and Convolution Neural Networks (CNN) for Video Tampering Classification. *International Journal of Soft Computing*, 14(3): 53-60. DOI: 10.36478/ijscmp.2019.53.60