CHAPTER 1
INTRODUCTION

This chapter presents an overview of the project tasks in shorts including the goal and potential applications of the project.

1.1 Introduction

The World Health Organization estimates that 347 million people have diabetes worldwide and the number will increase to 552 million by the year 2030. A diabetic person is at high risk of eye disease including diabetic retinopathy (DR), diabetic macular edema (DME), cataract, and glaucoma. The most common cause of vision loss, DR is caused by bleeding of the small blood vessels in the retina. If these bloody retina veins are untreated, it may cause varying degrees of vision loss and even blindness.

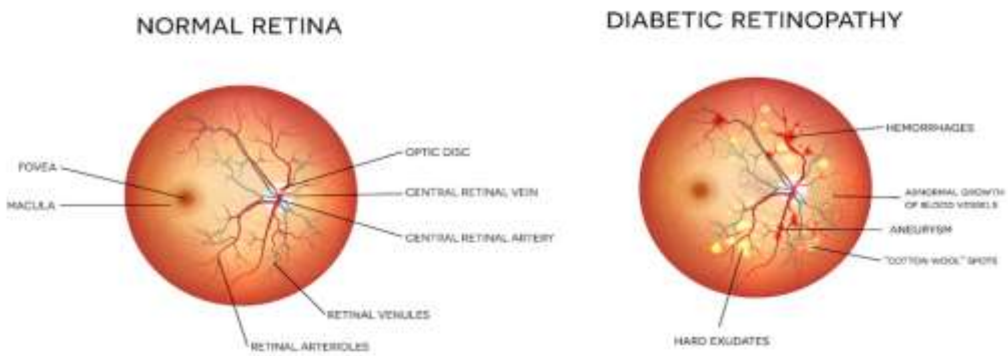


Figure 1: Normal Retina & Diabetic Retinopathy affected Retina [1]

The signs of DR can be listed as including but not limited to the existence of microaneurysms, hard exudates, vitreous hemorrhages, and retinal detachments. Figure 1 shows retina images with different DR levels such as (a) normal, (b) mild, (c) moderate, (d) severe, and (e) proliferative. As shown in Figure 1(ii), there is no abnormal lesions in the normal retina.

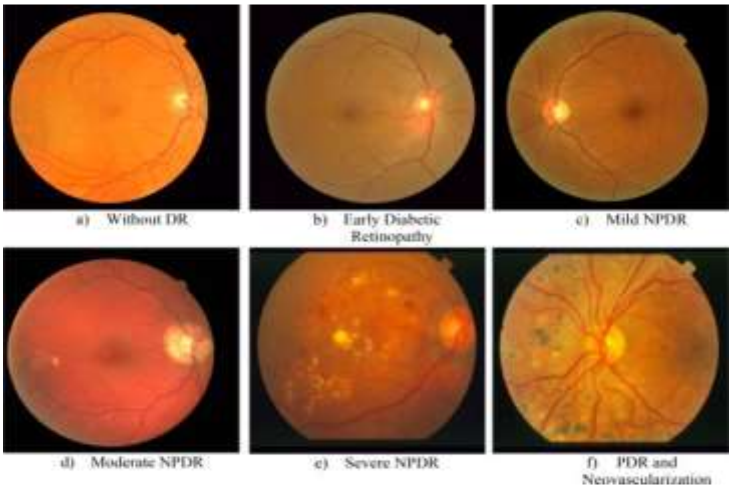


Figure 2: Stages of DR

In the retina examination, doctors use special optical devices such as an ophthalmoscope, a 20D (diopter) lens, and a fundus camera. Due to the easier storage, better image quality, and faster electronic transfer, fundus cameras are widely used to detect eye diseases with their digital imaging features. However, retinal imaging with a fundus camera is a time-consuming and manual process. Some certain level of expertise is required to capture a retinal image and an expert evaluation report may take a few days to submit. Since fundus cameras are also too big and heavy to be transported easily and too costly to be purchased.

Recent technological developments have enabled using of smartphones in designing small-sized, low-power, and affordable biomedical imaging devices. These systems are capable of imaging, onboard processing, and wireless communication. Therefore, smartphone-based systems are very popular in several applications ranging from health care to entertainment since they make existing systems small and portable.

1.2 Motivation & Aim

Recent technological developments have enabled use of smartphones in designing small-sized, low-power, and affordable biomedical imaging devices. These systems are capable of imaging, onboard processing, and wireless communication. Therefore, smartphone-based systems are very popular in several applications ranging from health care to entertainment since they make existing systems small and portable. Since fundus cameras are large-size, heavy-weight, and high-price devices, they are good candidate to be transformed into a portable device to perform fast DR screening. Recently, new smartphone-based retinal imaging systems are released to the market including D-Eye, Peek Retina, and iNview. We will use deep learning method to determine the existence of the diabetic retinopathy. It is a cloud-based retinal image assessment tool to grade DR development using deep learning methods that were trained with Kaggle dataset [8] providing the EyePACS retinal images.

The primary aim is to increase accessibility to diabetic retinopathy screening by leveraging the widespread availability of smartphones. This can particularly benefit individuals in underserved or remote areas who may not have easy access to specialized eye care facilities. Finally, the research aims to validate the accuracy and clinical utility of smartphone-based retinal image analysis for diabetic retinopathy detection. This involves rigorous testing of deep learning algorithms on diverse datasets to ensure reliable performance across different population groups and varying levels of disease severity.

Overall, the aim of research on smartphone-based retinal image analysis for diabetic retinopathy detection using deep learning is to develop a scalable, cost-effective, and patient-centric solution that can revolutionize the way diabetic retinopathy is screened for and managed, ultimately improving outcomes for individuals living with diabetes.

1.3 Objective

The main objective of research on smartphone-based retinal image analysis for diabetic retinopathy detection using deep learning is to develop a reliable, accessible, and cost-effective screening solution that can accurately detect diabetic retinopathy from retinal images captured using smartphone cameras. This involves leveraging the power of deep learning algorithms to automatically analyze retinal images for signs of diabetic retinopathy, with the aim of enabling early detection, timely intervention, and improved management of this potentially blinding complication of diabetes. Ultimately, the goal is to democratize access to diabetic retinopathy screening, particularly in underserved or resource-limited settings, and to empower individuals with diabetes to take proactive control of their eye health using readily available technology. With the improvement of computational power and advances in neural networks, deep learning algorithms, especially Convolutional Neural Networks (CNNs), have been widely used in different applications including retinal imaging. Using Fundus on Phone (FOP) device to capture high-quality retina images compared with the traditional fundus devices. Compare the performance of fundus camera-based deep learning models with existing standard methods for diabetic retinopathy screening, such as manual grading by ophthalmologists or traditional automated retinal imaging systems, to assess superiority, equivalence, or non-inferiority. These objectives collectively aim to advance the field of fundus camera-based retinal image analysis for diabetic retinopathy detection using deep learning, with the overarching goal of improving early detection, timely intervention, and management of this sight-threatening complication of diabetes.

CHAPTER 2
LITERATURE REVIEW

This chapter shows the different literature or paper review which can be studied for the new work using the deep learning field.

2.1 Introduction

In this section, we introduce solutions for the detection of DR using conventional and deep learning methods. There exist several telemedicine solutions for retinal image analysis for DR screening. These solutions in the literature need manual grading. However, they have to be fully automated to accelerate the diagnosis of retinal diseases using predictive models, especially for patients in rural areas. Then some literature compare the result between smartphone based and fundus camera retinal images used by detect the diabetic retinopathy detection.

2.2 Review Literature

Many researchers have already used deep learning based approached to detect Diabetic Retinopathy(DR). Gulshan et al. used specific deep learning methods for automated DR detection [5]. They have used CNN methods for image classification and trained their algorithm with Inception-v3 architecture [6] with about 128175 images using EyePACS and Messidor-2 datasets. Based on the results, they suggest that such a CNN network gives the best results when the network trained with 60000 images in terms of sensitivity and specificity. For the sensitivity, they reached about 97%.

Abramoff et al. [7] developed Iowa detection program for detecting rDR and they have used their own DR database and publicly available Messidor-2 dataset for training and testing, respectively. Based on the test results on the Mesidor-2 dataset, they achieved 96.8% sensitivity and 59.4% specificity for detecting DR.

Gargeya et al. [8] used a customized CNN technique to detect DR. They trained their system with 75137 fundus images from their own dataset and tested with Messidor-2 and E-Optha datasets. They classified images into two categories, one with the healthy eyes, the other with any DR stage, in other words, mild and worse DR. They acquired 94% sensitivity and 98%

specificity from their own dataset. Also, they tested their model with Messidor-2 dataset, and they achieved 93% sensitivity and 87% specificity.

Philip et al. [9] developed a DR assessment system based on healthy and disease conditions, also known as mild DR and worse. They trained their algorithm with 1067 images and tested with 14406 images. The performance of their algorithm was 86.2% and 76.8% with sensitivity and specificity, respectively.

In another study, Pires et al. proposed a solution for detecting rDR using data-driven approaches [10]. They used transfer learning techniques by applying to CNN. They applied on their training stage to data augmentation, multi-resolution, feature extraction, per patient analysis, and testing their solution with cross dataset logic by using Kaggle EyePACS dataset as a training, Messidor-2 dataset for testing. Based on the results, they obtained a 98.2% Receiver Operating Characteristic (ROC) curve for predicting rDR.

Since these methods in the literature focus on fundus images, they cannot be easily applied to smartphone-based images. For this reason, we need to investigate and create our own synthetic DR dataset using the FoV approach.

CHAPTER 3

METHODOLOGY

In this chapter the implementation procedure of the smartphone based app , algorithm and model which are used for prediction of diabetic retinopathy.

3.1 Proposed Method

We first present the details of smartphone-based portable retinal imaging systems available on the market to compare their features and image qualities. Second, we introduce the Field of View (FoV) determination process of each smartphone-based retinal imaging system using a circular test pattern. Third, we introduce the layout of the adopted deep learning architecture for DR detection. The research Methodology for "Comparison of smartphone-based retinal imaging systems for diabetic retinopathy detection using deep learning." is mainly incorporating Dataset collecting & Building, Dataset ,Pre-processing, Training and class predict as DR or NO_DR. The subsequent phase involves to preprocessing comprehensive dateset. Gray scaling, thresholding, masking are used to crop the retinal images from eye images for building dataset. In training phase DenseNet169, MobileNet, ResNet152V2, VGG16, VGG19, Xception model is implemented to extract hierarchical features from images, including subtle patterns and textures indicative of diabetic retinopathy. ResNet152V2.h5 model is saved and used to calculate the test accuracy and predict the class as DR or NO_DR.

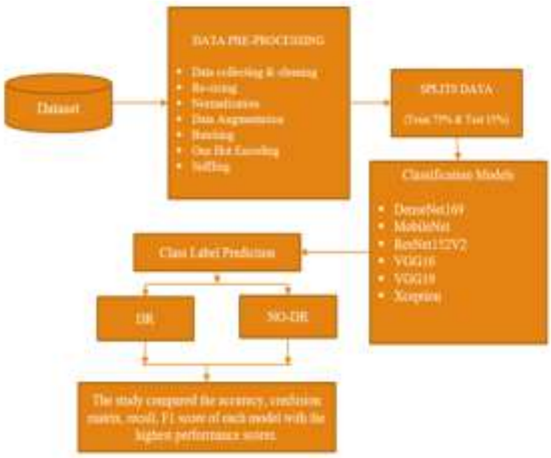


Figure 3: Proposed methodology

3.1.1 Data Collection

To investigate the automatic DR detection accuracy for smartphone-based retinal imaging systems and compare them with traditional fundus imagery, we set up two sets of experiments using synthetic and original retina images. For our experiments, we used several publicly available retina image datasets in gray scale image from kaggle. Then the original retina images in the dataset are originally color images and their resolutions vary since they captured by different fundus cameras. However, ResNet152V2, MobileNet, DenseNet, VGG19, VGG16, Exception Deep Learning framework requires the inputs to have 224x224x3 pixels as color images. Therefore, we first cropped pixels from the right and left sides of each original image to make it a square shape. In the other hand we collect image for create the second dataset, using smartphone and image capturing app. After collecting the image then generate 224x224 pixel image set.

Figure 4: Datasets

3.1.2 Data Pre-processing

Pre-processing of image data is crucial for improving the performance and efficiency of deep learning models, including convolutional neural network (CNN) classification models like ResNet152V2, VGG19, VGG16, MobileNet, DenseNet169, when applied to retinal image analysis for diabetic retinopathy detection. Here are some common preprocessing steps:

Data Collection and Cleaning: Gather your image dataset. Ensure that it's labeled correctly and that images are of consistent size and quality. Remove any corrupted or irrelevant images from the dataset.

Image Resizing: Resize retinal images to a standardized size suitable for the input dimensions of the deep learning model. This ensures consistency in image dimensions across the dataset and reduces computational complexity. Resize all images to the input size required by ResNet152V2. ResNet152V2 typically takes images of size 224x224 pixels. This can be done using libraries like OpenCV or PIL (Python Imaging Library).

Normalization: Normalize pixel values of retinal images to a common scale (e.g., [0, 1] or [-1, 1]) to improve convergence during model training. This involves dividing pixel values by the maximum intensity value (e.g., 255 for 8-bit images) or subtracting the mean and dividing by the standard deviation.

Data Augmentation: Augment retinal image data through techniques such as random rotations, translations, flips, and zooms to increase dataset diversity and improve model generalization. Data augmentation helps prevent overfitting and enhances the robustness of the model to variations in retinal image acquisition.

Noise Reduction: Reduce noise and artifacts in retinal images using denoising techniques such as Gaussian smoothing, median filtering, or wavelet denoising. Noise reduction helps improve the signal-to-noise ratio and enhances the quality of input data for the deep learning model.

Region of Interest (ROI) Extraction: Identify and extract the region of interest (e.g., the optic disc or macula) from retinal images if necessary for focusing model attention on clinically relevant areas. ROI extraction can improve model performance by reducing irrelevant information and emphasizing diagnostically important features.

One-Hot Encoding (if using categorical labels):

If your labels are categorical (e.g., for classification tasks), convert them into one-hot encoded vectors. For example, if we have three classes, 'cat', 'dog', and 'bird', you would represent 'cat' as [1, 0, 0], 'dog' as [0, 1, 0], and 'bird' as [0, 0, 1].

Shuffling: Shuffle the dataset to ensure that the model does not learn any order-based patterns during training. This helps prevent the model from memorizing the order of the examples.

Splitting into Training, Validation, and Test Sets:
Split your dataset into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune hyper-parameters and monitor performance, and the test set is used to evaluate the final model. Typically, a common split ratio is 70% training, 15% validation, and 15% testing.

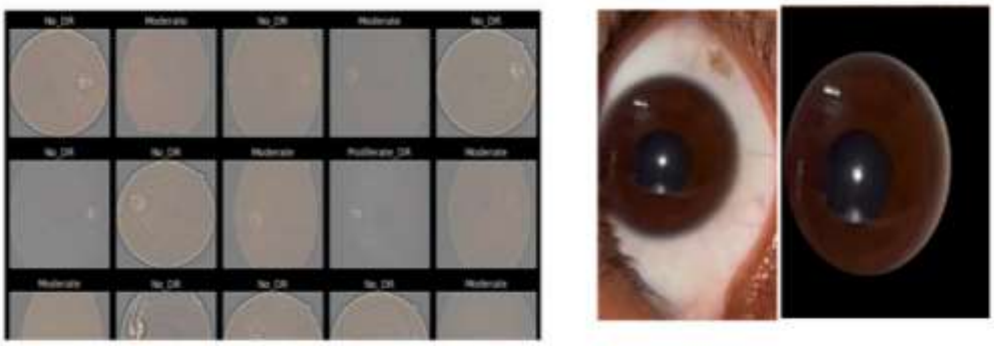


Figure5: After preprocessing of clinical images and captured image

Pre-trained Model Input Preprocessing:
If you're using a pre-trained ResNet152V2 model from a framework like TensorFlow or Keras, you might need to preprocess inputs according to the requirements of that framework. TensorFlow, for example, requires inputs to be in a specific range (typically [-1, 1] or [0, 1]).

3.1.3 Spliting Dataset

Here's an overview of the common data splitting techniques used in image dataset preparation:

Training Set: The training set is the largest subset of the dataset and is used to train the deep learning model. It consists of labeled examples (images) along with their corresponding ground truth annotations (e.g., diabetic retinopathy severity levels). The model learns from the training set to recognize patterns and features in the data. The training set is used to train the model.
Validation Set: The validation set is used to evaluate the performance of the model during training and to tune hyper-parameters. It provides an independent measure of the model's

performance on unseen data and helps prevent overfitting by monitoring performance metrics (e.g., accuracy, loss) on data not used for training.

Testing Set: The testing set is a completely independent subset of the dataset that is used to assess the final performance of the trained model. It provides an unbiased estimate of the model's generalization ability and is crucial for evaluating its effectiveness in real-world scenarios. The testing set should not be used for model training or parameter tuning to avoid data leakage and overfitting.

Common data splitting strategies for image datasets include:

Random Splitting, Stratified Splitting, Cross-Validation, Time-based Splitting. We used the Random Splitting where dataset is randomly partitioned into training, validation, and testing sets, typically with a predefined ratio (e.g., 70% training, 15% validation, 15% testing). Random splitting ensures that examples from each class are evenly distributed across subsets, minimizing bias in model training and evaluation.

3.1.4 Model Build

In the deep learning language CNN is commonly used for image data classification ,detection and segmentation .In CNN algorithm there are several classification models are used these are DenseNet169,MobileNet,VGG16,VGG19,ResNets152V2 , Exception etc.

3.1.4.1 DenseNet

DenseNet-169 is a dense convolutional neural network architecture that belongs to the DenseNet family. It is an extension of DenseNet-121 and is known for its deep and densely connected structure. DenseNet-169 has gained popularity for its remarkable performance in image classification and other computer vision tasks.The key building blocks of DenseNet-169 are dense blocks, similar to DenseNet-121. Dense blocks consist of multiple layers, where each layer is connected to every other layer within the block. This dense connectivity facilitates feature reuse and information flow across layers, enhancing gradient propagation and aiding in the extraction of rich and informative features.

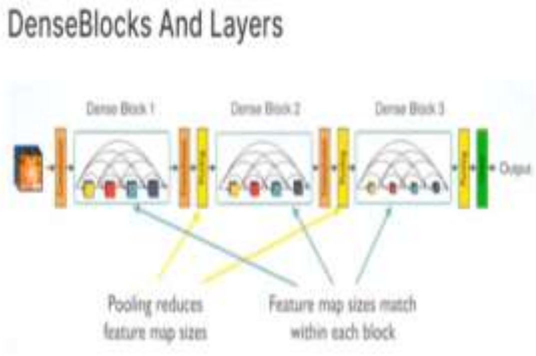


Figure 6: DenseNet169 model

3.1.4.2 MobileNet

Mobilenet is a model which does the same convolution as done by CNN to filter images but in a different way than those done by the previous CNN. It uses the idea of Depth convolution and point convolution which is different from the normal convolution as done by normal CNNs. This increases the efficiency of CNN to predict images and hence they can be able to compete in the mobile systems as well. Since these ways of convolution reduce the comparison and recognition time a lot, so it provides a better response in a very short time and hence we are using them as our image recognition model.

3.1.4.3 ResNet152V2

ResNet-152v2 is a variant of the Residual Network (ResNet) architecture, which is a deep convolutional neural network (CNN) designed for image classification tasks. ResNet-152v2 is an extension of the original ResNet architecture proposed by Microsoft Research in the seminal paper "Deep Residual Learning for Image Recognition.ResNet-152v2 is a very deep neural network, consisting of 152 layers. The depth of the network allows it to capture intricate hierarchical features in input images, which is beneficial for tasks requiring fine-grained discrimination, such as object recognition and scene understanding.ResNet-152v2 is known for its exceptional performance on benchmark image classification datasets such as ImageNet. It achieves state-of-the-art results in terms of classification accuracy and generalization ability, demonstrating the effectiveness of deep residual learning for image recognition tasks.

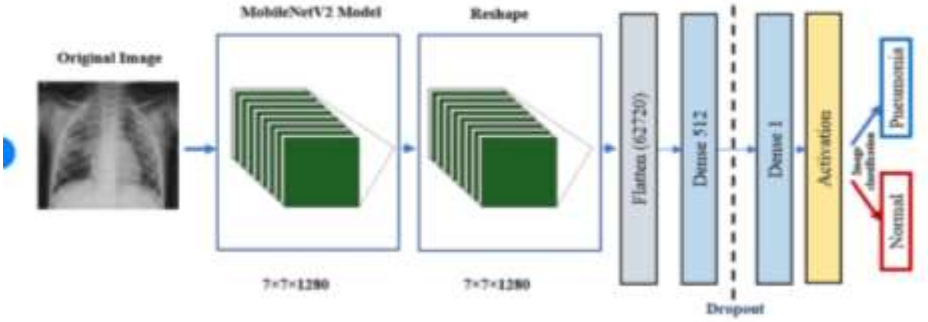


Figure 7: ResNet152V2 model

3.1.4.4 VGG16

A Convolutional Neural Network (CNN) architecture is a deep learning model designed for processing structured grid-like data, such as images. It consists of multiple layers, including convolutional, pooling, and fully connected layers. CNNs are highly effective for tasks like image classification, object detection, and image segmentation due to their hierarchical feature extraction capabilities. VGG16 is characterized by its depth, consisting of 16 layers, including 13 convolutional layers and 3 fully connected layers. VGG-16 is renowned for its simplicity and effectiveness, as well as its ability to achieve strong performance on various computer vision tasks, including image classification and object recognition. The model's architecture features a stack of convolutional layers followed by max-pooling layers, with progressively increasing depth. This design enables the model to learn intricate hierarchical representations of visual features, leading to robust and accurate predictions.

Figure 8 : VGG16 model

3.1.4.5 VGG19

VGG-19 Architecture VGG has an architecture of a CNN network, and VGG-19 is one of the VGG-based architectures. The VGG-19 is a deep-learning neural network with 19 connection layers, including 16 convolution layers and 3 fully connected layers. The convolution layers will extract features of the input images, and the fully connected layers will classify the leaf images for those features. In addition, the max-pooling layers will reduce the features and avoid overfitting, a. VGG-19 has 16 convolution layers grouped into 5 blocks. After every block, there is a Maxpool layer that decreases the size of the input image by 2 and increases the number of filters of the convolution layer also by 2.

3.1.4.6 Exception

Xception is an extension of the Inception architecture which replaces the standard Inception modules with depth wise separable convolutions. Exception classification model in CNNs focuses on identifying and classifying exceptions or anomalies in the input data. Using an exception classification model in image datasets within deep learning frameworks enhances data quality, improves model robustness, and enables the detection of anomalies or outliers, thereby enhancing the performance and reliability of various image-based applications.

CHAPTER 4
RESULT ANALYSIS

In this chapter describes the result of fundus camera image and smartphone based image results.

4.1 Diabetic Retinopathy Detection Using Single Dataset

To compare the smartphone-based retinal imaging systems, we set up two sets of experiments using smartphone captured iris images and real fundus camera gray scale images. We first collected retina images from kaggle. To make the fair comparison between different retinal imaging systems, we captured retinal images with iPhone6, iPhone15 pro, Vivo, Redmi iPhone7 using their compatible adapters and bumpers. In order to capture an image of the dark retina, we first needed to use a light source for illumination. For this purpose, we reflected the smartphone flashlight to the retina for use its own light source on capture the iris image. The smart phone's camera and using an app which was used to capture retina images and images were saved to the smartphone's memory.

4.2 Result Analysis For Fundus Camera Images

First, DenseNet, ResNet152V2, VGG16, VGG19 and Exception models were trained on the merged dataset (EyePACS, Messidor) dataset for DR detection. Table shows the overall train accuracy, validation accuracy, and specificity of our proposed networks from the validation and testing and compared with similar existing works in the literature. ResNet152V2 showed better performance for validation compared with DenseNet169, VGG19, VGG16, Execption. However, its accuracy for test images dropped and became better than MobileNet because MobileNet is the shallowest network among others. Besides, ResNet152V2 reached the highest accuracy of 94.6%, the validation accuracy is 96.2% for test images since it is the deepest network with a larger number of layers than others.

	Model	Train Accuracy	Validation Accuracy	Training Time (sec)
0	Xception	0.9231	0.9673	44.26
1	DenseNet169	0.9118	0.9618	38.96
2	ResNet152V2	0.9446	0.9618	50.84
3	MobileNet	0.9313	0.9564	22.25
4	VGG16	0.8517	0.9218	41.90
5	VGG19	0.8509	0.9073	21.57

Figure 9: Classification accuracy of deep learning frameworks

MobileNet provide better accuracy but works in small dataset but ResNet trained their CNNs from scratch using a very large dataset used transfer learning to retrain MobileNet and VGG19,VGG16 .The ResNet152V2 models overall took the longest time to train while the MobileNet models took the shortest time to train.

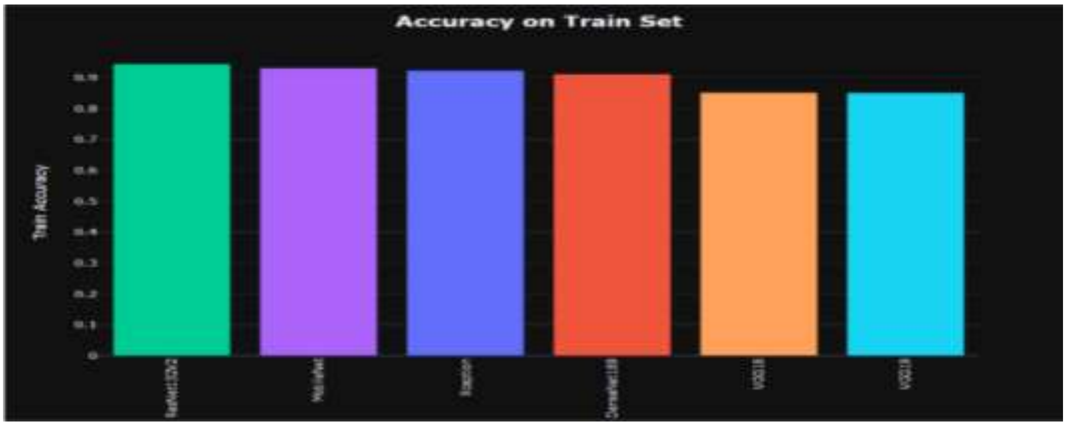


Figure 10: Classification accuracy comparison

The training and validation accuracy increasing along with the training loss and validation loss. Decreasing the training loss and validation loss indicates that the CNN's training session was a success.

	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Epoch
0	0.900078	0.354912	0.507273	0.707760	1
1	0.927400	0.188577	0.601818	0.623062	2
2	0.948087	0.136776	0.918182	0.398194	3
3	0.955894	0.138565	0.796364	0.438284	4
4	0.964091	0.097315	0.741818	0.479196	5
5	0.980484	0.050104	0.936364	0.168177	6
6	0.987119	0.031525	0.929091	0.271077	7
7	0.989852	0.026580	0.943636	0.345348	8
8	0.991413	0.019933	0.930909	0.391569	9
9	0.991803	0.019832	0.843636	1.197694	10

Figure 11: Classification accuracy comparison with Epoch

```
print(classification_report(test_batch.classes, y_pred, target_names = classes))

              precision    recall  f1-score   support

          DR       0.97      0.91      0.94       279
       No_DR       0.92      0.97      0.94       271

    accuracy                           0.94       550
   macro avg       0.94      0.94      0.94       550
weighted avg       0.94      0.94      0.94       550
```

Figure 12: Precision, recall, f1-score, accuracy

The high classification accuracy indicates that the model correctly predicted DR at a high rate.

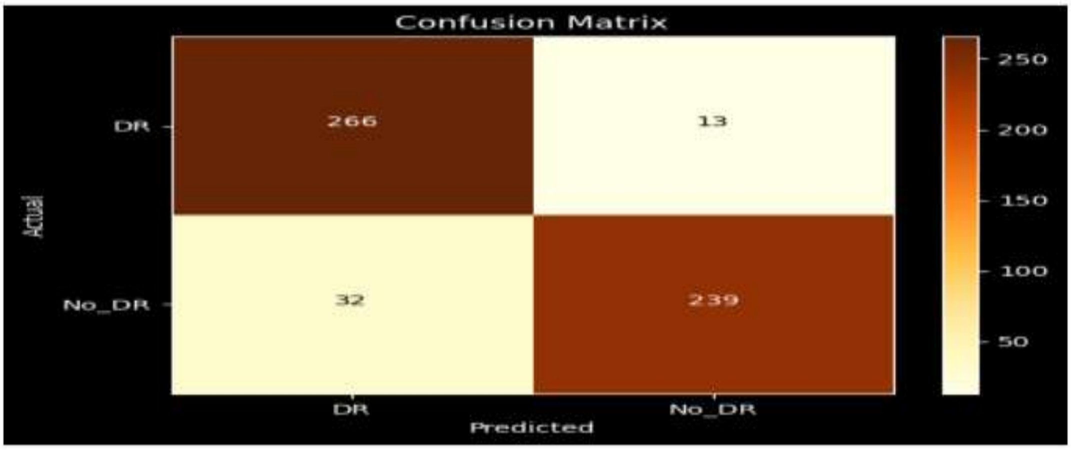
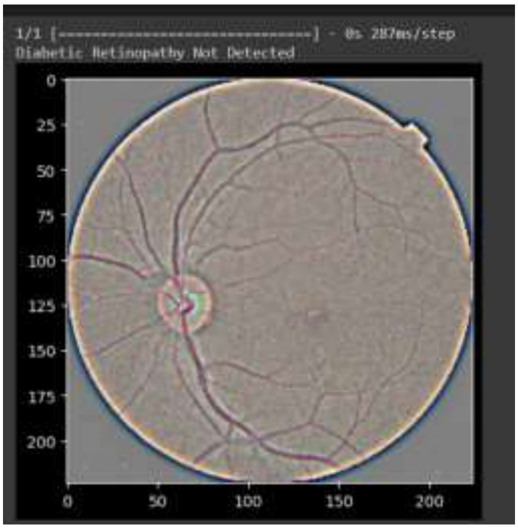


Figure 13: Confusion matrix using predict DR & No-DR

We developed a smartphone-based retinal image analysis system targeting at resource-limited setting. We used the ResNet152V2.h5 model to calculate the test accuracy of our capture image dataset.

4.3 Result Analysis For Prediction Of DR And No-DR

.We used the ResNet152V2.h5 model to calculate the train accuracy, validation accuracy, then generate a confusion matrix to calculate the precision,recall,f1-score .the confusion matrix provide high accuracy rate that proved the prediction of diabetic retinopathy detection is truly give the correct answer .After that we use the smartphone based iris image set to calculate the test accuracy of our capture image dataset ,then predict the diabetic retinopathy detection .



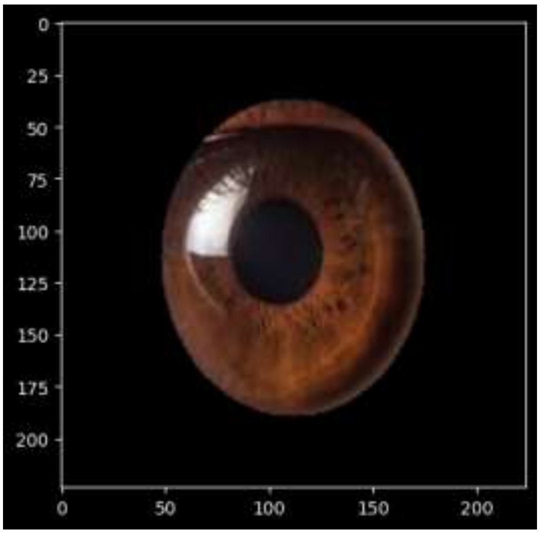


Figure14: fundus image DR prediction

Figure 15: Captured image DR prediction

CHAPTER 5

CONCLUSION

This thesis first investigated the fundus cameras images & smartphone-based portable retinal imaging system. Then, we adapted deep learning frameworks using transfer learning. Using smartphones is an emerging research area in designing small-sized, low-power, and affordable retinal imaging systems to perform accuracy calculation then based on the result then detecting the DR screening and automated DR detection due to the size, weight, and price of fundus cameras. Using smartphones is an emerging research area in designing small-sized, low-power, and affordable retinal imaging systems to perform DR detection due to the size, weight, and price of fundus cameras. This paper presented the utility of CNN-based MobileNet, DenseNet169, VGG19, VGG16 and ResNet152v2 frameworks to improve the performance of DR detection in smartphone-based and traditional fundus camera retina images. This study allowed us to compare the deep learning frameworks and to study the effect of FoVs in smartphone-based retinal imaging systems on their DR detection accuracy. Based on our results, the proposed ResNet152V2 approach showed the highest accuracy, sensitivity, and specificity for validation and test images compared with other frameworks and recently published related works.

References

[1] Voets, M., Møllersen, K., & Bongo, L. A. (2018). Replication study: Development and validation of deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. arXiv preprint arXiv:1803.04337.

[2] Karakaya, M., & Hacisoftaoglu, R. E. (2020). Comparison of smartphone-based retinal imaging systems for diabetic retinopathy detection using deep learning. BMC bioinformatics, 21(4), 1-18.

[3] Al-Smadi, M., Hammad, M., Baker, Q. B., & Sa'ad, A. (2021). A transfer learning with deep neural network approach for diabetic retinopathy classification. International Journal of Electrical and Computer Engineering, 11(4), 3492.

*[4]*https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-224x224-gaussian-filtered/code

*[5]*Gulshan, V., Peng, L., Coram, M., Stumpe, M.C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J. and Kim, R., 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. Jama, 316(22), pp.2402-2410

[6] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[7] Abràmoff, M.D., Folk, J.C., Han, D.P., Walker, J.D., Williams, D.F., Russell, S.R., Massin, P., Cochener, B., Gain, P., Tang, L. and Lamard, M., 2013. Automated analysis of retinal images for detection of referable diabetic retinopathy. JAMA ophthalmology, 131(3), pp.351-357.

*[9]*Gargeya, R. and Leng, T., 2017. Automated identification of diabetic retinopathy using deep learning. Ophthalmology, 124(7), pp.962-969.

*[10]*Philip, S., Fleming, A.D., Goatman, K.A., Fonseca, S., Mcnamee, P., Scotland, G.S., Prescott, G.J., Sharp, P.F. and Olson, J.A., 2007. The efficacy of automated “disease/no disease” grading for diabetic retinopathy in a systematic screening programme. British Journal of Ophthalmology, 91(11), pp.1512-1517.

*[11]*Pires, R., Avila, S., Wainer, J., Valle, E., Abramoff, M.D. and Rocha, A., 2019. A data-driven approach to referable diabetic retinopathy detection. Artificial intelligence in medicine, 96, pp.93-106.