SMARTPHONE DIABETIC RETINOPATHY DETECTION USING RAYTRACING IMAGE AUGMENTATION TECHNIQUES.

Minor Project Report

Submitted in the partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER ENGINEERING

Submitted By:

Husain Shahid Rao (20BCS028)
Sparash Mahajan (20BCS050)

Under the supervision of:

Dr.Sarfaraz Masood
(Associate Professor)

DEPARTMENT OF COMPUTER ENGINEERING FACULTY OF ENGINEERING & TECHNOLOGY

JAMIA MILLIA ISLAMIA

NEW DELHI (110025)

(YEAR - 2023)

CERTIFICATE

This is to certify that the project entitled “Smartphone diabetic retinopathy detection using raytracing image augmentation techniques.” by Husain Shahid Rao (20BCS028) and Sparash Mahajan (20BCS050) is a record of bonafide work carried out by them, in the Department of Computer Engineering, Jamia Millia Islamia, New Delhi, under my supervision and guidance in partial fulfilment of requirements for the award of Bachelor Of Engineering in Computer Engineering, Jamia Millia Islamia in the academic year 2023 - 24

Prof. Bashir Alam
(Head of Department)
Department of Computer Engineering
Faculty of Engineering & Technology
JAMIA MILLIA ISLAMIA
New Delhi

Dr. Sarfaraz Masood
(Associate Professor)
Department of Computer Engineering
Faculty of Engineering & Technology
JAMIA MILLIA ISLAMIA
New Delhi

ACKNOWLEDGEMENT

A very sincere and honest acknowledgement to Mr. Mohammad Zeeshan Ansari, Assistant Professor, Department of Computer Engineering, Jamia Millia Islamia, New Delhi for his invaluable technical guidance, great innovative ideas and overwhelming support. We are very grateful to our HOD Prof. Bashir Alam for his valuable support throughout the project. We would also like to express our gratitude to the Department of Computer Engineering and entire faculty members, for their teaching, guidance and encouragement. We are also thankful to our classmates and friends for their valuable suggestions and support whenever required. We regret any inadvertent omissions

Husain Shahid Rao

(20BCS028)

Sparash Mahajan

(20BCS050)

Department of Computer Engineering

Faculty of Engineering & Technology

JAMIA MILLIA ISLAMIA

NEW DELHI

ABSTRACT

Diabetic retinopathy (DR) stands as the leading cause of vision loss, stemming from damage to the delicate blood vessels in the retina. Left untreated, it can lead to progressive vision impairment and, ultimately, blindness. The silent nature of DR, often devoid of noticeable symptoms, underscores the critical importance of annual eye exams for early detection—a key determinant for successful intervention. Traditional tools for retinal imaging, such as fundus cameras, are hampered by size, weight, and cost, making them impractical for widespread screening in health clinics.

Recent technological strides have paved the way for innovative solutions, leveraging the ubiquity and capabilities of smartphones to create compact, low-power, and cost-effective retinal imaging systems. In this investigation, we explore the realm of smartphone-based portable retinal imaging systems, employing a medical-grade 20 diopter lens. Notably, we identify challenges tied to reduced field of view, lower resolution, and the introduction of irregularities like distortion, warping, and glare in smartphone-based retinal systems, all of which contribute to a decline in the performance of diabetic retinopathy detection networks.

To address the absence of publicly available datasets for fundus images captured by smartphones, this project concentrates on the creation of a custom dataset. Additionally, we scrutinize the viability of this approach by employing a synthetic dataset in conjunction with convolutional neural networks (CNNs) to assess accuracy. The findings from this study provide insights into the potential of smartphone-based retinal imaging systems for diabetic retinopathy screening, bridging the gap between accessibility and effective early detection in diverse healthcare settings.

TABLE OF CONTETNS

CERTIFICATE **3**

ACKNOWLEDGEMENT **4**

ABSTRACT **5**

TABLE OF CONTETNS **6**

1. INTRODUCTION **8**

1.1 What is Diabetic Retinopathy Detection 8

1.2 How smartphones are used for fundus photography 9

1.3 Challenges faced to develop an AI model for smartphone images 10

1.4 Raytracing technology to simulate optics 10

1.5 Deep Learning for Diabetic Retinopathy classification 11

2. REVIEW OF LITERATURE **12**

3. THEORETICAL BACKGROUND **13**

3.1 Convolutional Neural Networks 13

3.2 Transfer Learning 13

3.3 Object Localization 14

3.4 Ray Tracing and optics simulation 15

3.5 ResNet34 17

4. PROJECT OBJECTIVES **18**

4.1 Prepare an artificial dataset resembling fundus images taken from a smartphone 18

4.2 Use an existing CNN architecture to build a classification model 18

4.3 Test in real environment 18

5. METHODOLOGY **19**

5.1 Preparing dataset images 19

5.1.1 Optical disk localization 20

5.1.2 Crop the images to match smartphone FOV 21

5.1.3 Render images in a virtual lens setup in blender 23

5.1.4 Cutout the circular retina image from the rendered image 25

5.2 Generate custom labels for the custom dataset 26

5.3 Making a classification model for detecting DR 27

6. CONCLUSION **29**

7. PROGRAMMING TOOLS.. 30

8. REFERENCES.. 31

1. INTRODUCTION

1.1 What is Diabetic Retinopathy Detection

Diabetic retinopathy detection refers to the identification and diagnosis of diabetic retinopathy (DR), a serious eye condition that can affect individuals with diabetes. DR is characterized by damage to the blood vessels in the retina, the light-sensitive tissue at the back of the eye. The condition often develops over time and can lead to vision impairment or even blindness if left untreated.

Detection of diabetic retinopathy is crucial for timely intervention and management. Physicians, ophthalmologists, and healthcare providers use various methods to detect and diagnose DR. Traditional methods involve comprehensive eye exams, including dilating the pupils to allow for a detailed examination of the retina.

In recent years, technological advancements, particularly in the field of medical imaging and artificial intelligence, have led to the development of automated systems for diabetic retinopathy detection. These systems utilize imaging techniques, such as retinal photography or fundus imaging, to capture detailed images of the retina. The images are then analyzed using computer algorithms, often based on machine learning and deep learning approaches, to identify signs of diabetic retinopathy.

Automated diabetic retinopathy detection systems offer several advantages, including efficiency, consistency, and the potential for early detection. Early intervention and treatment can help prevent or slow down the progression of diabetic retinopathy, reducing the risk of vision loss in individuals with diabetes.

1.2 How smartphones are used for fundus photography

Smartphones are increasingly employed for fundus photography, a process involving the capture of detailed images of the back of the eye, including the retina. This trend is driven by the widespread accessibility of smartphones, as they are pervasive globally and owned by a significant portion of the population. Their cost-effectiveness compared to traditional fundus cameras makes them an attractive option, particularly in settings with financial constraints. The portability of smartphones is another significant advantage, allowing for flexibility in conducting retinal imaging in various healthcare settings, from clinics to remote or underserved areas. The integration of advanced cameras on smartphones, complete with features such as autofocus and image stabilization, enhances their capability for capturing high-quality fundus images. Moreover, smartphones facilitate telemedicine applications, enabling healthcare professionals to remotely assess and monitor patients' retinal health, especially valuable for individuals in remote locations. Patient engagement is enhanced as well, given the familiarity and comfort associated with using smartphones. Technological advancements, including add-on devices and artificial intelligence integration, further enhance smartphones' role in fundus photography. The method used in this project is the one which uses a 20-diopter lens. Examiner holds the smartphone in one hand while the 20D lens is held in another hand near the patient's eye.

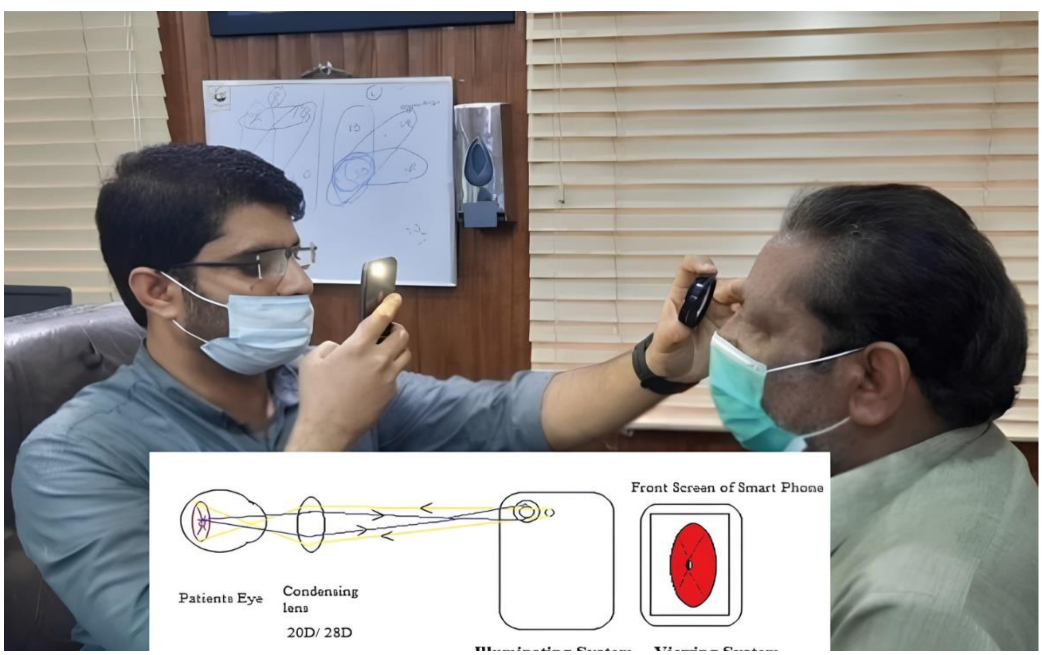

Figure 1: Positioning of examiner with 20D lens in one hand and smartphone with a flashlight on in another hand.

Examiner holds the lens between the thumb and index finger. The middle finger is used to elevate the eyelid while the little finger and ring finger stabilizes the hand on the patient's forehead. The filming distance is adjusted by the forward or backward movement of the lens or the smartphone, to the point the fundus is focused and fills the full area of condensing lens on the smartphone display screen.

1.3 Challenges faced to develop an AI model for smartphone images

Extensive research in diabetic retinopathy detection using deep learning has predominantly relied on high-resolution retinal image datasets captured by fundus cameras. However, the absence of a publicly available dataset containing retinal images from smartphones prompted our exploration. Collaborative efforts with ophthalmologists for a bespoke dataset were deemed impractical due to varied smartphone-related factors such as model differences, flashlight placement, environmental lighting, and examiner proficiency. To address this, we opted for a pragmatic approach, utilizing existing fundal image datasets and applying augmentation techniques. This methodology involves artificially expanding the dataset by simulating diverse scenarios, effectively preparing the model to handle the inherent noise and variations present in retinal images captured via smartphones. This strategy ensures the development of a robust model capable of generalizing across a spectrum of real-world conditions.

1.4 Raytracing technology to simulate optics

Ray tracing technology proves invaluable in simulating the behavior of light through lenses within 3D environment software. By leveraging advanced optical principles, ray tracing enables the accurate modeling of light paths as they interact with virtual lenses, mimicking the intricate play of reflections, refractions, and diffractions. In a 3D environment, the software, equipped with ray tracing capabilities, can replicate the optical characteristics of different lens types, including the refocusing effects achieved through varying focal lengths and apertures. This simulation provides a highly realistic portrayal of how light behaves within a lens system, offering a valuable tool for visualizing and analyzing optical phenomena in virtual spaces. Applying this technique by virtually constructing the smartphone fundus photography setup (Figure 1) can provide the required augmentations to the original image, making it appear as if taken from a smartphone. This process can be automated using a Python script to render all the images in different virtual environments. While this process will require moderate graphics processing power, it will result in much better augmentation than conventional methods.

1.5 Deep Learning for Diabetic Retinopathy classification

Deep learning for diabetic retinopathy classification involves the application of advanced neural network models to analyze and classify fundus images for signs of diabetic retinopathy. These models, often based on Convolutional Neural Networks (CNNs), are trained on large datasets of annotated retinal images, learning to automatically identify patterns associated with different stages of diabetic retinopathy. Through features learned from the data, deep learning models can distinguish between normal and pathological conditions, providing a valuable tool for early detection and classification of diabetic retinopathy. The use of transfer learning, data augmentation, and interpretable techniques enhances the model's performance, making it a promising approach for efficient and accurate screening in clinical settings. Deep learning's ability to handle complex visual information makes it a key player in the advancement of automated diabetic retinopathy diagnosis, contributing to timely intervention and improved patient outcomes.

2. REVIEW OF LITERATURE

Karakaya and Hacisoftaoglu , in their paper, "Comparison of smartphone-based retinal imaging systems for diabetic retinopathy detection using deep learning", explore diverse methods for capturing fundus images with smartphones and associated add-ons. Their comparative study involves utilizing different tools for smartphone-based fundus image capture. Additionally, they propose a method to convert an existing EyePACS dataset into one resembling smartphone-captured images. This involves segmenting the original retina image boundary, fitting a circle using circular Hough transform, detecting the optic nerve, and generating circular masks for each device.

The authors of the paper “Diabetic Retinopathy Classification Using CNN and Hybrid Deep Convolutional Neural Networks” proposed to use CNN and hybrid deep convolutional neural networks to detect and grade DR. They used a large dataset of 3662 train images and 1928 test images, and performed image preprocessing and augmentation to enhance the quality and diversity of the images14. They compared three models: CNN, hybrid CNN with ResNet, and hybrid CNN with DenseNet. They found that the hybrid CNN with DenseNet model outperformed the other models, achieving an accuracy of 96.22%15. They also compared their model with the existing works and showed that it had a higher accuracy and a lower complexity. They concluded that their model was a simple, accurate, and efficient method for automated DR detection and grading.

3. THEORETICAL BACKGROUND

3.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of deep learning network that can learn features from data, especially images, and perform various tasks such as classification, segmentation, detection, and generation. CNNs are inspired by the biological vision system, where neurons in the visual cortex respond to different stimuli in the visual field. CNNs consist of multiple layers, each of which performs a specific operation on the input data.

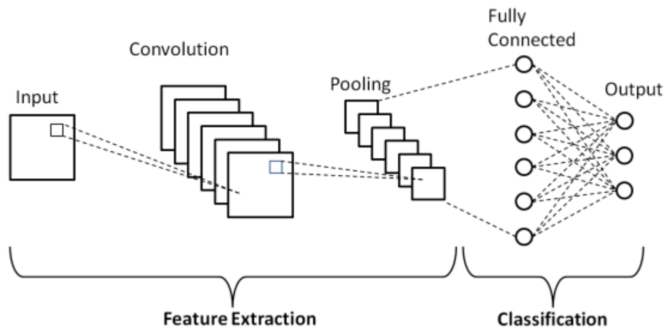

Figure 2: Schematic diagram of a basic convolutional neural network (CNN) architecture.

3.2 Transfer Learning

Transfer learning is a technique in machine learning where a model trained on one task is used as the starting point for a model on a second task. This can be useful when the second task is similar to the first task, or when there is limited data available for the second task. By using the learned features from the first task as a starting point, the model can learn more quickly and effectively on the second task. This can also help to prevent overfitting, as the model will have already learned general features that are likely to be useful in the second task

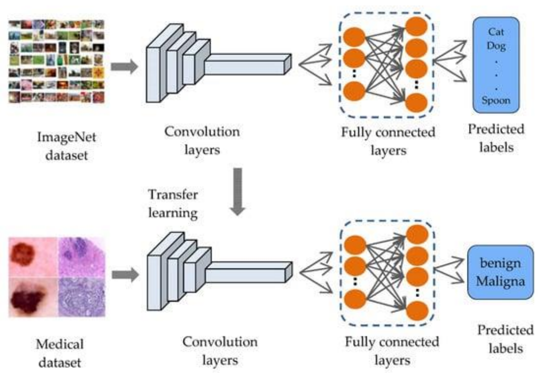

Figure 3: Transfer learning from ImageNet.

Transfer learning for medical imaging is to use a model that was trained on a large natural image dataset, such as ImageNet, and fine-tune it on a medical image dataset. ImageNet is a dataset of over 14 million images with 1000 classes, covering a variety of objects, scenes, and animals. By using a model that was trained on ImageNet, such as AlexNet, VGGNet, ResNet, or Inception, we can leverage the general features that the model has learned, such as edges, shapes, colors, and textures, and adapt them to the medical domain. This can improve the performance and speed of the model, as well as reduce the risk of overfitting.

Another way to use transfer learning for medical imaging is to use a model that was trained on a large medical image dataset, and fine-tune it on a smaller or different medical image dataset. This can be more effective than using ImageNet, as the model can learn more domain-specific features.

3.3 Object Localization

Object localization is a task in computer vision that aims to locate and identify objects in an image or a video. It usually involves two steps: object detection and object classification. Object detection is the process of finding regions of interest (ROIs) that contain potential objects.

Faster R-CNN is a deep learning model that can perform object localization using CNNs. It consists of two modules: a region proposal network (RPN) and a fast R-CNN network. The RPN is a fully convolutional network that generates ROIs from the input image using anchor boxes and sliding windows. The fast R-CNN network is a CNN that takes the ROIs as input and outputs the bounding box coordinates and the class probabilities for each ROI.

One way to implement object localization using transfer learning and faster R-CNN is to use a pre-trained convolutional neural network (CNN) as the backbone of the faster R-CNN model and fine-tune it on a new dataset.

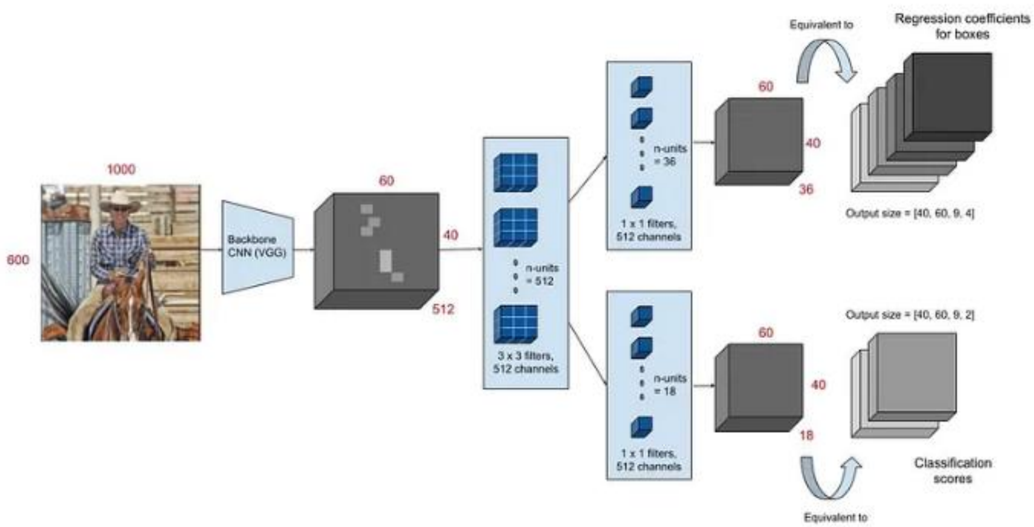

Figure 4:The architecture of the region proposal network or RPN

3.4 Ray Tracing and optics simulation

Ray tracing is a rendering technique for generating an image by tracing the path of light as pixels in an image plane and simulating the effects of its encounters with virtual objects. Ray tracing can be used in various 3D design software like blender, POV-Ray, etc.

Blender is a 3D creation suite that supports the entire 3D pipeline, from modeling to rendering. Blender’s Cycles is a ray-trace based production render engine that can be used for ray tracing.

Blender has an add-on called OptiCore that provides accurate and well-defined models of optical elements such as lenses, mirrors, and optomechanical components like optical bench and posts. OptiCore can help you create renderings of optical laboratory experiments and stunning images involving caustics.

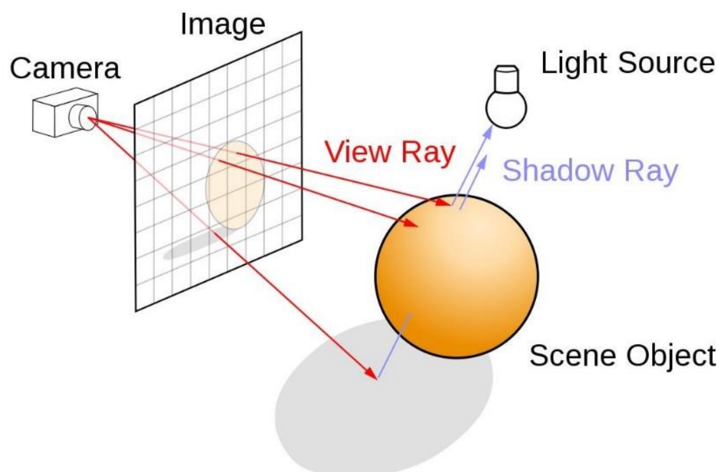

Figure 5: Ray tracing basics

Figure 6: Blender layout 3D view

3.5 ResNet34

ResNet-34, short for Residual Network with 34 layers, is a variant of the ResNet architecture designed for deep learning tasks, particularly image classification. Introduced by Microsoft Research, ResNet-34 is characterized by its distinctive residual blocks, which address the challenge of training very deep neural networks by mitigating the vanishing gradient problem. The architecture consists of 34 layers, including multiple residual blocks with skip connections, allowing for the smooth flow of information during both forward and backward passes. ResNet-34 has proven effective in various computer vision tasks due to its ability to capture intricate features from images while avoiding the degradation problem associated with increasing network depth. It strikes a balance between model complexity and computational efficiency, making it a popular choice for applications where a moderate-sized deep network is desirable.

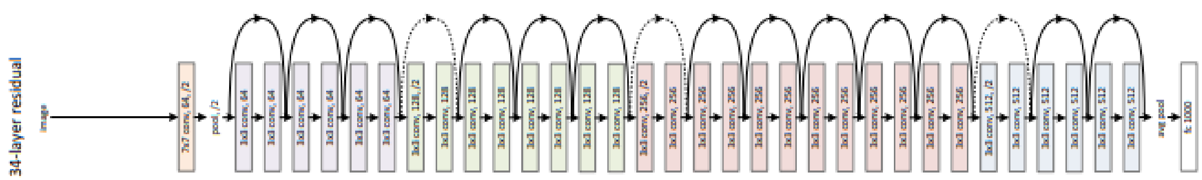

Figure 7: A plain network with 34 parameter layers (3.6 billion FLOPs).

4. PROJECT OBJECTIVES

4.1 Prepare an artificial dataset resembling fundus images taken from a smartphone

☑ Analyze an existing dataset of fundus images like IDRiD or EyePacs

☑ Apply augmentations and transformations to the images

☑ Change disease grading labels to just two: DR and No DR

4.2 Use an existing CNN architecture to build a classification model

☑ Implement an existing CNN architecture for image classification

☑ Apply transfer learning using imagenet-v1 as initial weights

☑ Train the model on the artificial dataset and compare results with the original dataset

4.3 Test in real environment

☐ Get small data of real smartphone fundus images with DR labels

☐ Test the model with these samples

5. METHODOLOGY

We will be using the Indian Diabetic Retinopathy Imaging Dataset (IDRID) in the entire project. This dataset is divided in three parts:

A. Segmentation

   Consists of original color fundus images and masks for optical disks. Total of 81 images divided in train and test set.

B. Disease Grading

   Original color fundus images (516 images divided into train set (413 images) and test set (103 images) with diabetic retinopathy and macular edema severity grade. Diabetic retinopathy has five labels -

   0: No DR, 1: Mild DR, 2: Moderate DR, 3: Severe DR and 4: Proliferate DR

C. Localization

   Original color fundus images (516 images divided into train set (413 images) and test set (103 images) with labels for optic disk center location and fovea center location.

5.1 Preparing dataset images

Preparation of the dataset will be a multistep process. The steps are improvement to some steps mention in the paper "Comparison of smartphone-based retinal imaging systems for diabetic retinopathy detection using deep learning"

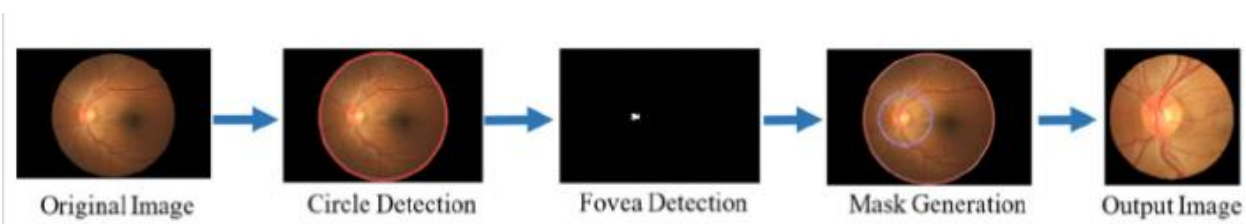

Figure 8: Steps used in the previous research papers

We will initially develop a model for localization of fovea, instead of segmentation, then crop the image to match the smartphone FOV, then we will use a 3D rendering engine to simulate how retina would look through a lens in a virtual environment, and then finally we will use circle detection to crop out the final output image. All the steps are discussed in detail below.

5.1.1 Optical disk localization

The preferred method for optical disk localization was the implementation of Faster R-CNN, chosen for its robust performance in object detection tasks. Specifically, the Faster R-CNN ResNet50 FPN model from the `torchvision.models.detection` library was utilized. Background weights were set to None, and the number of classes was configured to 2, designating one class for the background and the other for the optical disk. This model configuration proved effective in accurately localizing optical disks within images.

Prior to training the model, it is imperative to adjust the labels, considering that Faster R-CNN requires bounding box positions while our current labels only provide center positions for the optical disk. To address this, a bounding box size needs to be defined, and subsequently, bounding box coordinates must be generated in accordance with this size. This process ensures compatibility with the Faster R-CNN model, enabling it to effectively learn and localize optical disks based on bounding box information during the training phase.

$$boundingBox = [X - boxSize, Y - boxSize, X + boxSize, Y + boxSize]$$

Train the model for approximately 10 epochs on the training dataset and save the model weights in a file. When testing on random fundus images, you should observe bounding boxes near the optical disk.

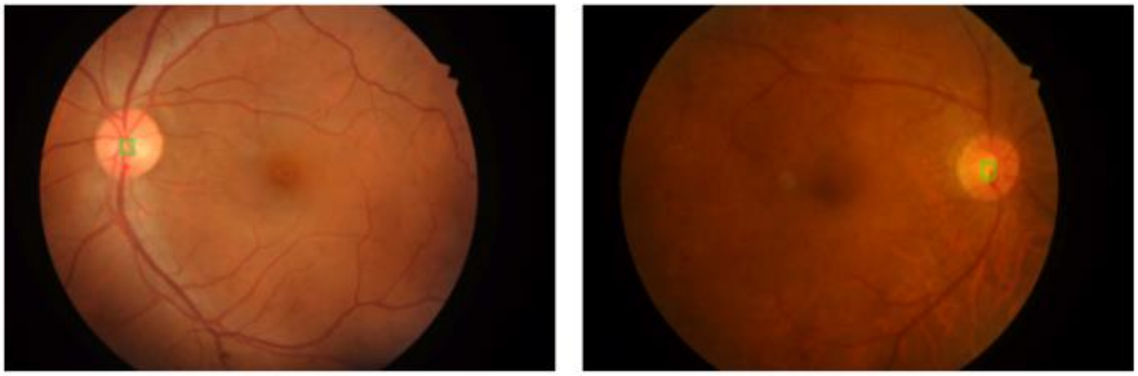

Figure 9: Bounding box on optical disk. Box size can be changed by changing boxSize variable

Once the model has been successfully trained, proceed to employ it for predicting optical disk locations in any diabetic retinopathy classification dataset. A larger dataset would be advantageous for enhanced generalization. Here, for the sake of simplicity, we will utilize the model in the Disease Grading section of the IDRID. Predict the optical disk locations and save the results in a CSV file.

5.1.2 Crop the images to match smartphone FOV

Now that we have obtained the locations of optical disks, the next step involves cropping images to match the field of view (FOV) of a smartphone. Drawing upon insights from previous research, a comparison of FOV using various tools for capturing fundus images has been documented. This comparative analysis serves as a reference to guide the cropping process, ensuring that the resultant images align with the FOV specifications deemed suitable for smartphone-based fundus imaging. This step is integral to maintaining consistency and relevance across the dataset, aligning with established standards in the field.

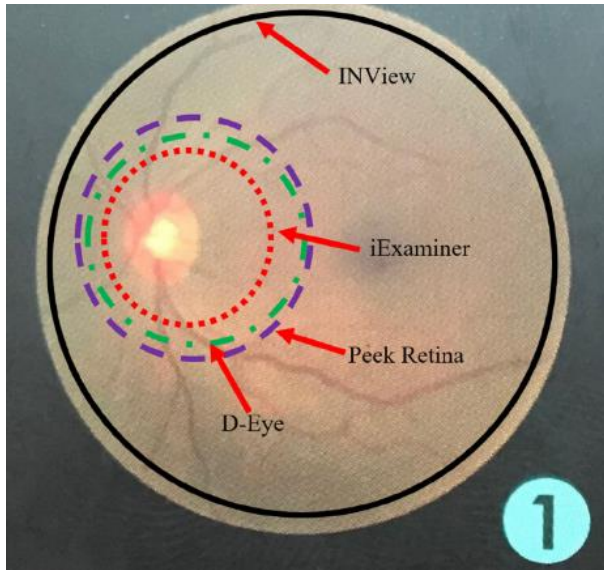

Figure 10: Comparison of the field of view of each smartphone-based retinal imaging system

We will set max image size to 3000 px and then crop the retina images such that optical disk is try to be placed at center. Calculate the top left corner of the image using center of the optical disk result image size.

$$[topX, topY] = \left[\max\left(\text{opticalDiskX} - \frac{\text{imgSize}}{2}, 0\right), \max\left(\text{opticalDiskY} - \frac{\text{imgSize}}{2}, 0\right)\right]$$

Save the resultant images as they will be used in the next stage of the processing pipeline.

5.1.3 Render images in a virtual lens setup in blender

This is one of the most computationally expensive step in the process as it requires large GPU power to process big datasets. Firstly, create a lens, preferably using OptiCore addon for blender, for realistic optic simulations. Set the focal length of lens to 50mm since we are simulating a 20D lens. Now, add other objects in the scene including a placeholder plane where we will place our images from the dataset prepared in the last step. A camera with a focal length of 28mm, mimicking that of an iPhone camera, and a spot light, mimicking flashlight of the smartphone. Now place the lens at a distance of about 50mm from the retinal plane and the adjust the camera position such that the image is in the frame.

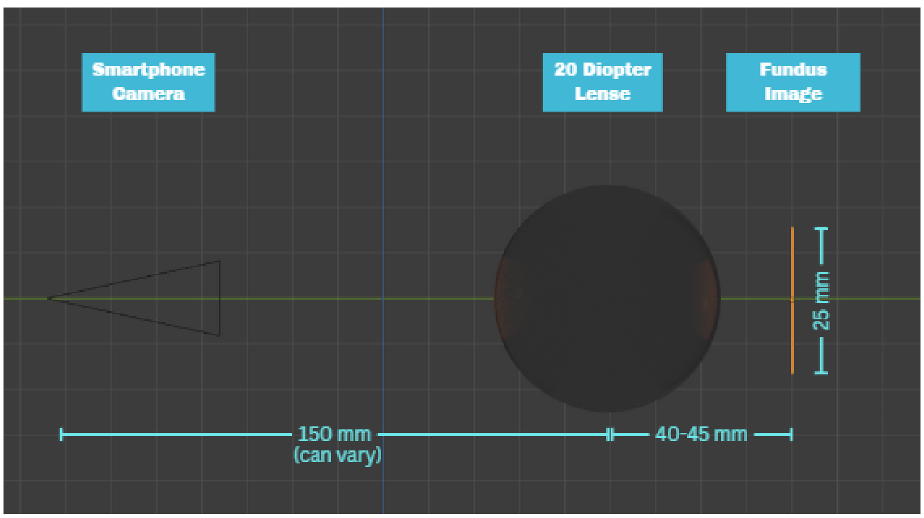

Figure 11: Orthogonal view of the scene

We can make multiple scenes and can change variables like distance of lens from the retina while automating the rendering using a python script in the scripting section of blender.

If you are using native blender glass material as the shader, instead of the lens provided by OptiCore addon, make sure the shader has Glass BSDF selected with IOR 1.2 and roughness between 0.03 to 0.05.

In scripting tab, write a script to vary rotation and distance of lens by different values for each image, and render each image in the dataset folder. So if we have **rotations = [0, 1.2]** and **distances = [85, 90]**, for **n** number of images, we will get **4*n** output images.

Running this script for a dataset of 413 images, on high end PC with 12GB RTX 3060 GPU, Ryzen 5 3600X CPU and 16 GB of RAM it took about 4 hours to render all the images.

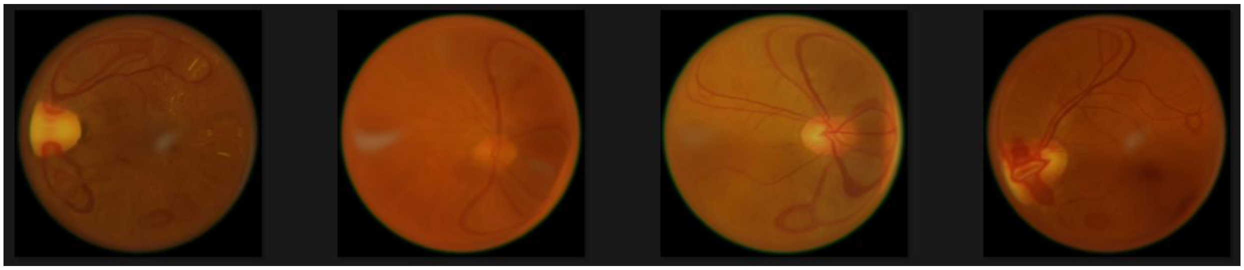

Figure 14: Rendered images with lens distortion and chromatic aberration

Figure 13: Lens shader

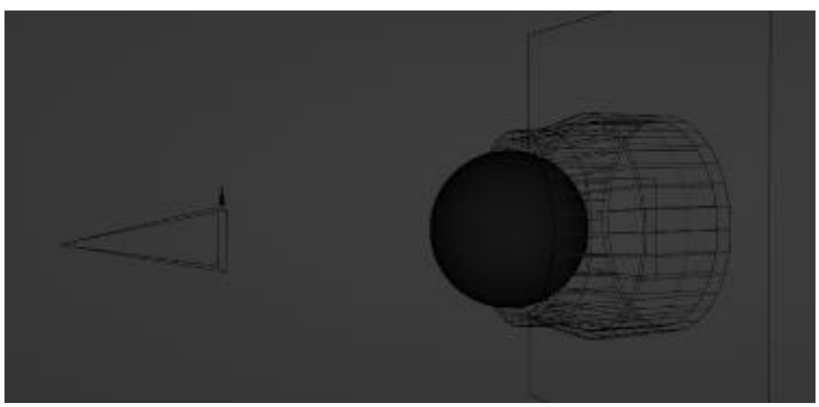

Figure 12: Wireframe view of the scene with an ambient light blocking casing around the lens

5.1.4 Cutout the circular retina image from the rendered image

In the culminating stage of the image preparation phase, the critical task involves meticulously cropping out the circular fundus image from the originally rendered rectangular image. This intricate process is executed with precision through the application of the HOUGH_GRADIENT method from the renowned computer vision library, OpenCV. Employing the HOUGH_GRADIENT method enables the algorithm to adeptly detect and isolate the circular boundaries of the fundus image, utilizing a technique rooted in gradient information. This method proves instrumental in extracting the relevant anatomical details present in the circular fundus image, a pivotal step that contributes to the overall accuracy and effectiveness of subsequent image analysis and processing tasks. By leveraging the capabilities of OpenCV's HOUGH_GRADIENT method, the image preparation phase attains a level of sophistication that ensures the extraction of the fundus image with optimal precision, laying a solid foundation for subsequent stages in the computational pipeline.

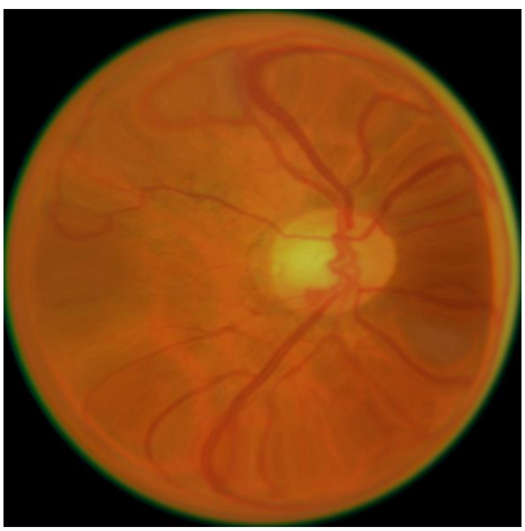

Figure 15: A No DR sample image after final step

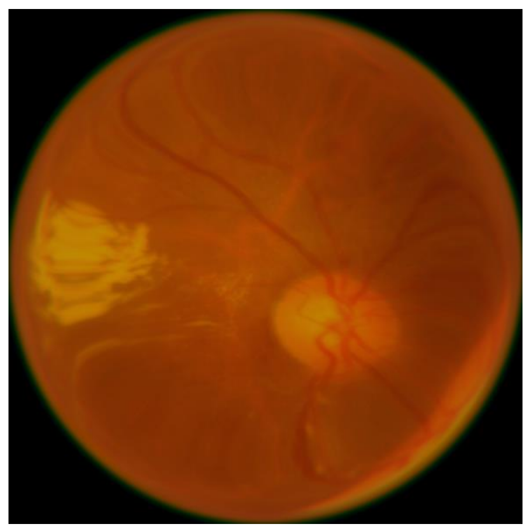

Figure 16: A DR sample image after final step

5.2 Generate custom labels for the custom dataset

After applying above techniques to the original dataset, we have considerably reduced the amount of information contained in an image. Therefore, it will be difficult to differentiate between different levels severity of diabetic retinopathy. To simplify our dataset, we will reduce our labels to just two- 0: DR, 1: No DR.

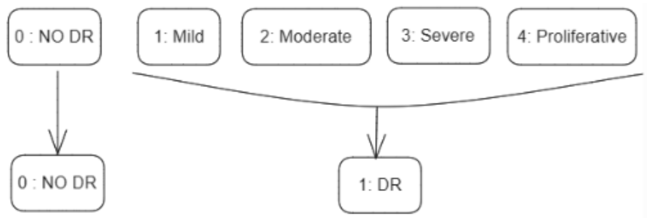

Figure 19: Reducing five labels to just two

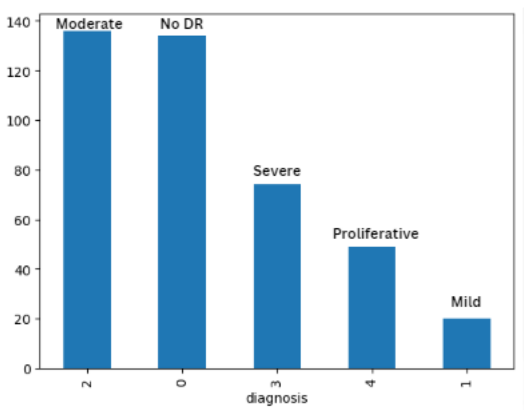

Figure 17: Label distribution before applying transformation

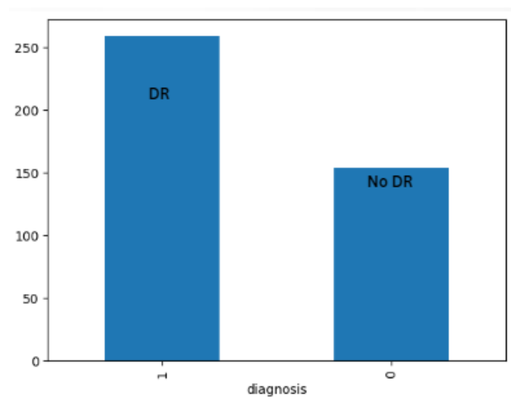

Figure 18: Label distribution after applying transformation

To rectify irregularities in our label distribution, we used scikit-learn's **class_weight** function during model training. This dynamic adjustment of class weights mitigates imbalanced distributions, preventing bias towards majority classes and improving generalization, especially in datasets with rare or less prevalent classes.

5.3 Making a classification model for detecting DR

Numerous studies have explored various CNN architectures for classifying diabetic retinopathy images in the literature. Due to our limited dataset, consisting of only 413 images, we opted for transfer learning in our implementation. As a benchmarking model to evaluate the performance of our dataset, we selected ResNet34. This architecture is pretrained on the Imagenet1k_v1 dataset, which comprises 14 million images across 1000 object classes.

```
model = models.resnet34(weights=models.ResNet34_Weights.IMAGENET1K_V1)
```

We intend to modify the architecture of the pretrained model by substituting its final layer with a set of four new layers. This modification is undertaken to tailor the model's output structure to the specific requirements of our task or dataset. This process of layer replacement is a common practice in transfer learning, allowing us to leverage the learned features of the pretrained model while customizing the model's output to suit the problem at hand.

```
model.fc = nn.Sequential(nn.Linear(512,256),
                         nn.ReLU(inplace=True),
                         nn.Linear(256,128),
                         nn.ReLU(inplace=True),
                         nn.Linear(128,64),
                         nn.ReLU(inplace=True),
                         nn.Linear(64,2),
                    )
```

With the finalization of our model architecture, the subsequent step involves training the model on the processed dataset. The training process encompasses iterating through the dataset for 60 epochs, allowing the model to adapt and optimize its parameters. Throughout this training phase, we closely monitor both training and validation losses. Tracking these losses is essential for assessing the model's performance and ensuring that it is effectively learning from the dataset. By observing the trends in training and validation losses, we gain insights into the model's convergence and generalization capabilities.

At the conclusion of the training process, our model achieved an accuracy of 75.15% on the training set and 75.90% on the validation set, reflecting its proficiency in learning from the provided dataset. Subsequently, when subjected to evaluation on an unseen test set, the model demonstrated an accuracy of 70.66%

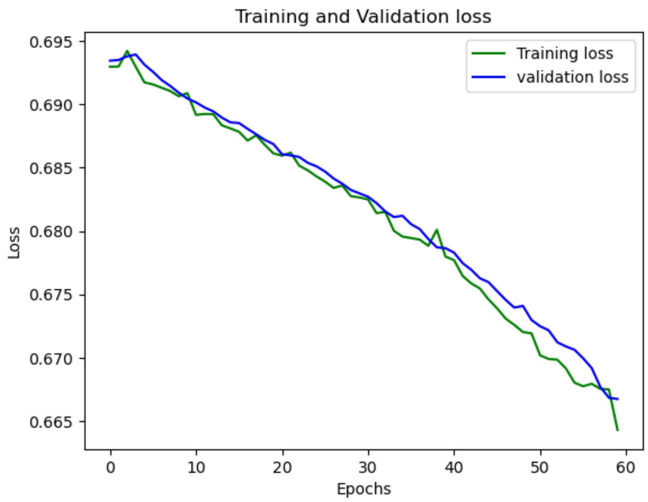

Figure 21: Training and Validation loss on custom dataset

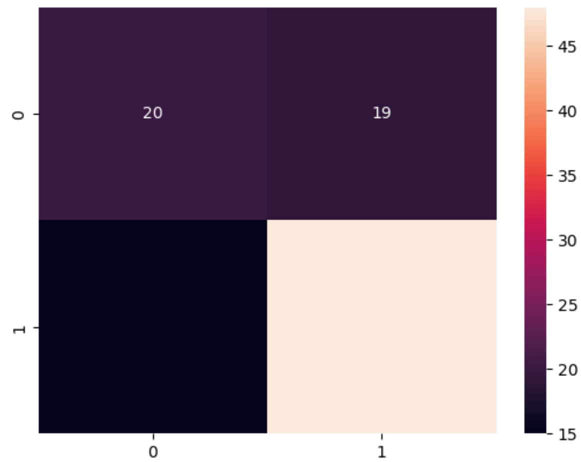

Figure 20: Confusion matrix for the test set

6. CONCLUSION

The observation from the confusion matrix indicates a higher occurrence of false positives in the model's predictions. This pattern may be attributed to the aggregation of five severity levels into just two, suggesting a potential source of confusion for the model. The simplification of severity levels can lead to a loss of nuanced information, making it challenging for the model to discern between different degrees of diabetic retinopathy accurately. To address this, reevaluating the classification approach or considering a more fine-grained representation of severity levels might be beneficial. Adjusting the model architecture, revisiting the dataset preprocessing, or exploring alternative classification strategies could be avenues for improving the model's performance in distinguishing between the nuanced severity levels of diabetic retinopathy.

To have better insights about there is an issue with the model architecture or the data preprocessing steps, we train the same model with original IDRiD dataset and compare the results. Large improvement will indicate that we are loosing valuable information in data preprocessing, moderate improvement will indicate that there is a need to improve our CNN architecture or use different architecture, which is more fit for DR classification.

Metric	Original Dataset	Custom Dataset
Epochs	30	30
Training Set Accuracy	72.12%	73.63%
Validation Set Accuracy	71.08%	73.49%

From the above for both the original dataset and the custom dataset, it becomes apparent that the initial approach utilizing resnet34 may not be optimal. Despite achieving reasonable accuracy, there is room for improvement. Exploring more advanced architectures, such as deep ResNet variants like resnet50, or considering alternative CNN architectures specifically tailored for diabetic retinopathy classification. This strategic shift in model architecture may lead to improved accuracy and effectiveness in the classification task.

7. PROGRAMMING TOOLS

Languages:

Python - https://www.python.org/

Frameworks:

pyTorch - https://pytorch.org/

Libraries:

Pandas - https://pandas.pydata.org/

Numpy - https://numpy.org/

OpenCV - https://opencv.org/

OptiCore - https://github.com/CodeFHD/OptiCore

Other tools:

Anaconda environment manager - https://www.anaconda.com/

Blender Open source 3D designing software - https://www.blender.org/

Jupyter Notebook - https://jupyter.org/

VS code - https://code.visualstudio.com/

8. REFERENCES

[1] Iqbal, U. Smartphone fundus photography: a narrative review. Int J Retin Vitr 7, 44 (2021)

[2] Karakaya M, Hacisoftaoglu RE. Comparison of smartphone-based retinal imaging systems for diabetic retinopathy detection using deep learning. BMC Bioinformatics. 2020

[3] R. Y, Raja Sarobin M. V, Panjanathan R, S. GJ, L. JA. Diabetic Retinopathy Classification Using CNN and Hybrid Deep Convolutional Neural Networks. Symmetry. 2022; 14(9):1932.

[4] Ren S, Kaiming H, Girshick R and Sun J 2016 Faster R-CNN: towards real-time object detection with region proposal networks

[5] He K, Zhang X, Ren S and Sun J 2016 Deep Residual Learning for Image Recognition (IEEE) 770–8

[6] Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, Fabrice Meriaudeau, April 24, 2018, "Indian Diabetic Retinopathy Image Dataset (IDRiD)", IEEE Dataport, doi: https://dx.doi.org/10.21227/H25W98.

[7] Lord RK, Shah VA, San Filippo AN, Krishna R. Novel uses of smartphones in ophthalmology. Ophthalmology. 2010;117(6):1274-1274e1273