

# Automatic Classification of Preliminary Diabetic Retinopathy Stages using CNN

Omar Khaled<sup>1</sup>, Mahmoud ElSahhar<sup>2</sup>, Mohamed Alaa El-Dine<sup>3</sup>, Youssef Talaat<sup>4</sup>, Yomna M. I. Hassan<sup>5</sup>, Alaa Hamdy<sup>6</sup>  
Faculty of Computer Science,  
Misr International University,  
Cairo, Egypt

**Abstract**—Diabetes Mellitus is one of the modern world's most prominent and dominant maladies. This condition later on leads to a menacing eye disease called Diabetic Retinopathy (DR). Diabetic Retinopathy is a retinal disease that is caused by high blood sugar levels in the retina, and can naturally progress to irreversible vision loss (blindness). The primary purpose of this imperative research is the early detection and classification of this hazardous condition, to try and prevent any threatening complications in the future. In the course of recent years, Convolutional Neural Networks (CNNs) turned out to be exceptionally famous and fruitful in solving and unraveling image processing and object detection problems for enormous datasets. Throughout this pivotal research, a model was proposed to detect the presence of (DR) and classify it into 5 distinct stages, factoring in an immense and substantial dataset. The model starts by applying preprocessing techniques such as normalization, to maintain the same dimensions for all the images before proceeding to the main processing stage. Furthermore, diverse sampling methods such as “Resize & Crop”, “Rotation”, and “Flipping” have been tested out, so as to pinpoint the best augmentation technique. Finally, the normalized images were fed into a Convolutional Neural Network (CNN), to predict whether a person suffers from DR or not, and classify the level/stage of the disease. The proposed method was utilized on 88,700 retinal fundus images, which are a parcel of the full (EyePACS) dataset, and finally achieved 81.12%, 89.16%, and 84.16% for sensitivity, specificity, and accuracy, respectively.

**Keywords**—Diabetes mellitus; diabetic retinopathy; DR; convolutional neural networks (CNNs); image processing

## I. INTRODUCTION

### A. History and Background

As indicated by the World Health Organization (WHO) [1] around 422 million individuals worldwide have been determined to have Diabetes Mellitus. These cases were especially in low and middle income nations, and the expressed numbers are only expected to increment with time. As indicated by Lee's et al. [2] studies, 33% of individuals experiencing Diabetes Mellitus are likewise determined to have other eye maladies, such as Diabetic Retinopathy. This implies that around 147 million individuals are at risk.

The correlation between Diabetes and retinal complications has been first found and presented in 1856, however, it was not until the second half of the 20<sup>th</sup> century that this eminent work gave more proof that proposed that Retinopathy really was an entanglement of diabetes.

In recent years a new approach to accurately diagnose

and detect the presence of Diabetic Retinopathy has been introduced. The approach mainly replaces the old-fashioned manual diagnosis of (DR), with a modern automated method. Automatic classification and analysis of retinal fundus images is materializing as one of the most significant screening tools for the early detection of (DR). This new approach not only provides more reliable and accurate results, but it also saves a lot of time and money.

Diabetic Retinopathy is an illness that causes retina irregularity from the norm, and in extreme conditions can without a doubt lead to total blindness. A classification technique was suggested that interprets and extracts features and aspects from retinal fundus images, and determines whether an individual experiences (DR) or not, and what level or stage is he/she currently at. This research is additionally centered around distinguishing and immediately perceiving the characteristics and qualities of (DR) for ideal precision during the classification operation.

### B. Motivation

Around 39 million individuals in the MENA region (Middle East and North Africa) experience the ill effects of Diabetes Mellitus, and it is without a doubt expected that by 2045 this number will ascend to 67 million. The inspiration and motivation to tirelessly seek after this particular issue was that out of these alarming numbers, 8.2 million cases were in Egypt in 2017 as indicated by the “Worldwide Diabetes Federation”[3]. Which further implies that third of this number is undoubtedly at risk of experiencing the ill effects of (DR).

After thoroughly investigating and breaking down the market, it was found that the preliminary phases or stages of Diabetic Retinopathy and other eye ailments were not identified precisely manually. Moreover, two principle disadvantages that were without a doubt pivotal factors in precisely identifying (DR) were likewise revealed; the datasets utilized in the process were surprisingly little, which obviously prompted the second potential downside that being, low classification and accuracy rates. So based on this critical information, the primary point becomes to devise a successful method for classifying and identifying the preliminary phases of Diabetic Retinopathy, for possible clinical advantages.

1) *Problem Analysis*: The main issue is that very high blood sugar levels over a broad period of time, cause harm throughout the entire human body. This case occurs when the blood vessels behind the retina get weakened over time and get

damaged, which eventually causes new abnormal blood vessels to grow at the back of the eye. Moreover, these abnormal blood vessels not only leak fluids into the eye, but they can also cause more serious complications such as vision loss (blindness) or glaucoma.

Diabetic Retinopathy is viewed as one of the deadliest infections around the globe, since one probably won't have any visible symptoms of (DR) in the beginning phases, yet as the ailment advances or normally progresses, Diabetic Retinopathy side effects and symptoms might occur. (DR) symptoms and side effects can include Blurred vision, impaired color vision, or spots and dark strings floating in one's vision (floaters).

## II. RELATED WORK

The following section introduces the most relevant published work, that primarily depicts and represents the proposed research in terms of applied algorithms, datasets used, number of classified stages, and the overall achieved accuracy.

### A. SVM Classifier

Bhattacharjee et al.[4] using Random Forest classifier, classified Diabetic Retinopathy based on three features, which are the area of microaneurysm, the area of blood vessels, and the area of exudates. And using these features, they have classified them into five stages: normal, mild, moderate, severe and Proliferative using Kaggle resized images which are about 10052 images for training, and 3350 images for testing to achieve an accuracy of 76.5%.

Kumar et al. [5] classified 89 images from DIARETDB1 dataset into two stages after removing the noise on the images by using (CALHE) histogram equalization. They also extracted hard exudates, Blood vessels, the area of MA, and the number of MA. Outputting a result for sensitivity and specificity of 96% and 92%, respectively.

Cisneros et al. [6] reached an accuracy between 84.6% and 87.3% by using 413 images for training and 130 for testing, to extract the hard exudates. They also segmented the blood vessels and other properties.

Tjandrasa et al. [7] used soft margin SVM on 149 images from the Messidor dataset, extracting from it the features and properties such as area, perimeter, standard deviation and energy of each exudated image during the feature extraction process, to finally classify between Moderate and Severe cases. They reached an accuracy of 90.54%.

Carrera et al. [8] used 400 images from the Messidor dataset which contains four different stages: Normal, Mild, Moderate and Severe. They then extracted the features of the images by detecting blood vessels, microaneurysms, hard exudates, and other features. They finally reached an accuracy of 85%.

Sangwan [9] trained their system on 96 images while using 54 images to classify three different stages: Mild, Moderate and Proliferative. After performing histogram equalisation on the dataset and turning it into grey-scale images, they reached an overall accuracy of 92.6%.

1) *Convolution Neural Networks Algorithms*: Lian et al. [10] explored three neural network architectures: AlexNet, ResNet-50 and VGG-16 on a dataset that was provided by EyePACS via kaggle. The dataset consists of 35,126 fundus images and distributed to five classes: normal, mild NPDR, moderate NPDR, severe NPDR and severe PDR. The classes have a Proportion of 73.46% ,6.69 % , 15.06 % , 2.50% and 2.02%, respectively. They then normalized all the images from their original size into 256x256 pixels. Also, they re-sampled the images for the over represented classes, and randomly sub-sampled the underrepresented classes. Finally, they used spatial translation with one pixel in both left and right horizontal directions, to increase the number of images and avoid bias. The three models achieved accuracy rates of 73.19% for the AlexNet model, 76.41% for the ResNet-50, while VGG-16 achieved the best accuracy which was 79.04%.

Harun et al. [11] classified two stages of Diabetic Retinopathy, using 1,151 fundus images divided into 70:30 data proportion, in which 806 images were used for training, while 345 images were used as testing images. They then achieved an overall accuracy of 67.47% for classifying the two classes, 66.4% and 64.48% for DR and No DR, respectively. They finally used a Multi-layer Perceptron (MLP), trained by Binary relevance for classification with 50 training epochs and 20 hidden layers.

Li et al. [12] proposed a system to classify two stages of Diabetic Retinopathy, and its five different stages using Deep Convolutional Neural Networks. They used a Kaggle dataset divided into 34,124 for training, 1,000 for validation, and 53,572 for testing. They finally reached an accuracy for the five-class classification of 86.17%, while the accuracy for the binary class was 91.05%.

Challa et al. [13] detected and classified the five different stages of Diabetic Retinopathy using an All-CNN architecture that has ten convolutional layers and a Softmax layer. They used a Kaggle dataset divided into 30,000 images for training, and 3,000 images for testing. They then applied some preprocessing techniques on the dataset such as removing black Boundaries, and data augmentation such as vertical and horizontal flipping; rotation in different angles between 45° and 180° to make sure that all the images in stages 1,2,3 and 4 are equal to the images in level 0. They finally achieved an accuracy of 86.64% for classifying the five stages of Diabetic Retinopathy, but from observing the the percentage of Recall, Precision and F1 score in the five classes, class (0) had the highest percentage compared to other classes that didn't exceed 60%.

Junjun et al. [14] applied the Residual Network (ResNet) approach on the EyePACS dataset that contains about 35,126 images, using about 30,000 images for training, and around 5,000 images for testing. The images were then resized to 256 × 256 pixels, and due to the large number of images from class 0, they augmented the images of the other classes to avoid over-fitting, by flipping the images and rotating randomly between 0° and 360° to classify 5 stages with an accuracy of 78.4%.

Jain et al. [15] detected Diabetic Retinopathy and evaluated its severity through using different Convolutional Neural Network (CNN) Architectures such as VGG-16, VGG-19 and

Inception v3 architectures. They divided the 35,126 images from the EyePACS dataset into 60% for training, 20% for validation, and the last 20% to classify 5 stages. The images have been passed on preparation techniques such as Normalization and Data Augmentation, by rotating the images for training by 90° and 270°. As for their results they reached an accuracy of 71.7%, 76.9% and 70.2%, respectively.

Kwasigroch et al.[16] classified 5 stages of Diabetic Retinopathy by using VGG-D architecture on 37,000 images after scaling and cropping the images to 224 x 224 pixels. Data augmentation method was performed on the images, such as horizontal and vertical shifts & flips, rotations, and zooming. They finally obtained an accuracy of 81.7%.

kajan et al.[17] designed a Diabetic Retinopathy classifier to detect the degree of the presence of the disease in the eye using different pre-trained deep neural networks models, such as: VGG-16, ResNet-50, and Inception-v3 on the EyePacs dataset. They used 75% of the dataset randomly for training and the rest for testing, they then created two models, the first model classified the presence of Diabetic Retinopathy in the fundus images, while the second model classified the degree of this disease into four different stages. They achieved their best results of the average classification accuracy of the first classification model using ResNet50 model, and achieved an accuracy of 92.64%. While the other model using InceptionV3 reached an average of 70.29% among the four stages.

Suriyal et al. [18] used MobileNets model for classifying two stages (non-presence & presence) of Diabetic Retinopathy on 16,798 resized images, and used 1000 images for testing from Kaggle dataset. The overall accuracy achieved was 73.3%.

Harangi et al. [19] used different CNN architectures like ResNet, VGGNet, GoogleNet and AlexNet, on about 552 images from e-optha-MA, ROC, DIARETDB1, and only 32 images from the dataset Messidor. They also resized the images to 224 x224 pixels to reach an accuracy between 78.56% & 83.35%.

Khan et al.[20] used cropped resized images, and performed histogram equalization on the Messidor dataset before inputting them into these pretrained models: SqueezeNet, AlexNet and VGG-16, to detect the presence of Diabetic Retinopathy. Finally, their classification produced an accuracy between 91.82 % and 94.49 %.

Zeng et al. [21] classified Diabetic Retinopathy using the Kaggle data set as input fundus images, with a training set of only 28,104 images, and a test set of 7024 images. And also weight-sharing layers based on two architecture Inception-V3 pretrained model siamese-like network structure. They also used pre-processing methods like flipping the images horizontally, geometric transforming as cropping, scaling, translating, and shearing the fundus images. Their final result showed that they achieved a score of 82.2%.

Carson et al. [22] classified the disease into four distinct levels using convolutional neural networks (CNN) such as: AlexNet and GoogLeNet models. The dataset used was a mix between the Kaggle dataset of around 35,000 fundus images, and the Messidor dataset of 1,200 fundus images after passing from different preprocessing methods as cropping the images

to separate the circular colored image of the retina using Otsu. They also normalized the images using (CLAHE) histogram equalization algorithm, and data augmentation by zooming, rolling and rotating the images to reduce the over-fitting. The final overall accuracy was between 57.2% and 74.5%.

### III. METHODOLOGY

The main goal of this software is to automatically detect the early stages of Diabetic Retinopathy and classify the level of the disease in the patient's body. Our aim in this project is to help as many Diabetic patients as possible, by preventing Diabetes from affecting their eyesight and progressing to Diabetic Retinopathy. The idea of the system, after thoroughly reading about Diabetes and Eye diseases, was that Diabetic Retinopathy doesn't show any symptoms, until a very late stage in life. As shown in Fig. 1, the system starts by taking a retinal fundus image as input from the diseased patient and apply some data preprocessing techniques. Using a deep learning approach, the system will then detect whether a person suffers from Diabetic Retinopathy or not; based on the answer, the system will then classify the level of the disease and finally propose a solution to the patient.

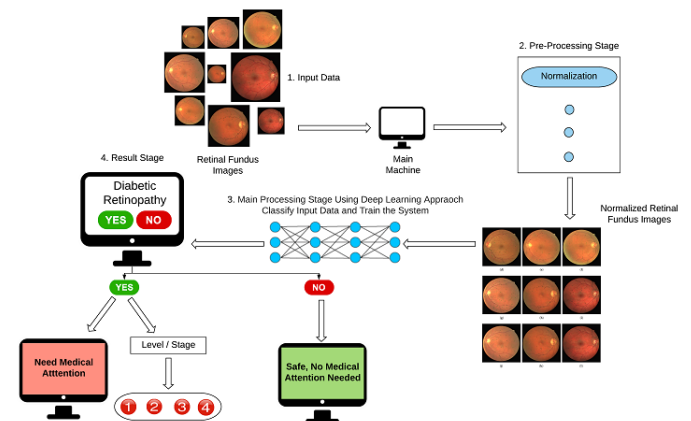


Fig. 1. System Overview.

First, The Main Computer/Machine will collect the information from the Input Images, which are unnormalized Retinal Fundus Images with different sizes chosen from our dataset. The System will then apply some data pre-processing algorithms such as Normalization so that all the images are the same size and dimension. This will thoroughly help us in the main processing phase by simply reducing the complexity of the Input images. Then comes the Main Processing stage in which we test and train our system, We use a Convolutional Neural Network in which a group of connected nodes distributed on multiple layers enhance and strengthen each other along with the Tensor-flow library for feature extraction and classification of the Input images. The System will then proceed to the final stage, which is the Result stage; If the result turned out to be (YES), the System will show the Level/Stage of the disease on a scale of 4-stages, and finally propose that the patient needs medical attention right away. Else (NO) the System will propose that there is no need for medical attention.

The upcoming techniques and operations have been carefully decided upon, to best fit the proposed research, and emphatically improve the overall system performance.

1) *Dataset*: After thoroughly researching and analyzing the different kinds and sizes of datasets, the following dataset has been specifically chosen, since it contains the largest number of reliable retinal fundus images.

The High-Resolution dataset being utilized in this study is called (EyePACS) [23], which consists of 88,702 images, provided by Kaggle [24], and classified into 5 stages as shown below in Fig. 2. Furthermore, the number of images within each class is shown below in Table I.

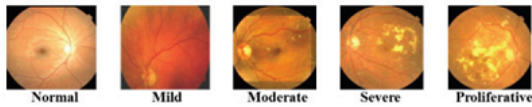


Fig. 2. Stages of Diabetic Retinopathy (DR) with Increasing Severity. [25]

TABLE I. DATASET IMAGES DISTRIBUTION

	Class 0	Class 1	Class 2	Class 3	Class 4
Images	65,343	6,205	13,153	2,087	1,914

### A. Dataset Preparation

1) *Data Filtration*: After examining the dataset, it was found that about 245 images were totally corrupted as shown below in Table II, and could eventually cause distractions and obstructions to the model while training. Accordingly, it was decided to eliminate these images from the model’s training process.

TABLE II. DATASET FILTRATION PROCESS

Class Name	Original Images	Original Images (After Filtration)
Class 0 (No DR)	65,343	65,167
Class 1 (Mild)	6,205	6,190
Class 2 (Moderate)	13,153	13,116
Class 3 (Severe)	2,087	2,087
Class 4 (Proliferative)	1,914	1,901
Total	88,702	88,457

2) *Data Normalization*: After filtering the dataset from all the corrupted images, a new technique called Normalization is then applied. All the Input retinal images maintain different sizes and dimensions, so to overcome this difficulty it was promptly decided to normalize/resize all the retinal fundus images, so they regularly have similar measurements before continuing to the main processing stage. At this point the normalized images keep a unified size of 224 pixels in width and 224 pixels in height, as shown below in Fig. 3.

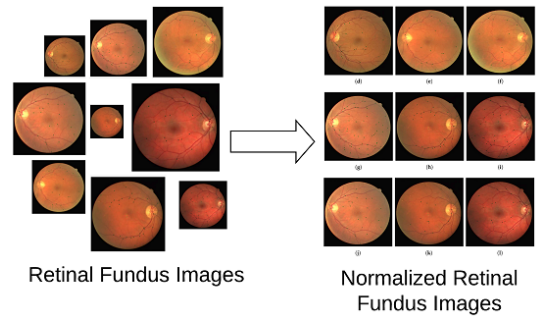


Fig. 3. Retinal Fundus Images Normalization Process.

3) *Data Splitting*: During the model training process several different splitting techniques were tested out, so as to be able to choose the most suitable, unbiased splitting method.

1) **Random Splitting**: The dataset is randomly split into 70% training and 30% testing, among the 5 different classes shown below in Fig. 4.

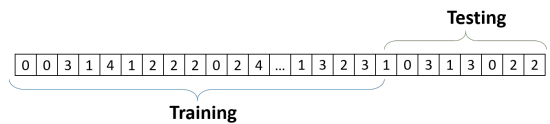


Fig. 4. Random Data Splitting.

2) **Per-Class Splitting**: The dataset is split into 70% training and 30% testing for each individual class. To further ensure that the operation is applied perfectly on each class; the retinal images were grouped together according to their class name, then each group (class) was split into 70% training and 30% testing, shown below in Fig. 5.

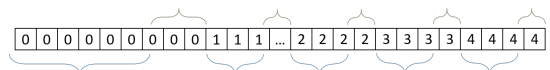


Fig. 5. Per-Class Data Splitting.

3) **Equal Per-Class Splitting**: An equal number of images is taken from each class, and then each class is split into 70% training and 30% among the 5 different classes, shown below in Fig. 6.

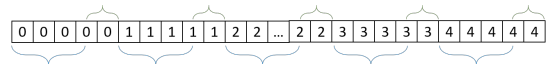


Fig. 6. Equal Per-Class Data Splitting.

4) **Conclusion**: After trying out the three different data splitting methods, it was apparent that the “Equal Per-Class Data Splitting” proved to be the best technique, since it ensures that an equal number of training and testing images is used per class, which ultimately guarantees a fair and unbiased system.

4) *Data Sampling*: As mentioned earlier, the 5 classes within the dataset are unbalanced, and this situation will eventually cause bias towards the largest class, which in this case is class 0 (No DR). To overcome this difficulty, it was decided to apply various up-sampling techniques, so that the number of images per class becomes even. Therefore, to find the most suitable technique, we had to try out three distinct methods (“Rotation”, “Flipping”, “Resize & Crop”).

- 1) **Rotation:**  
Starting with the Rotation technique, we applied this method through rotating the images by (90, 180, 270) degrees shown below in Fig. 7, so as to be able to generate new image samples.

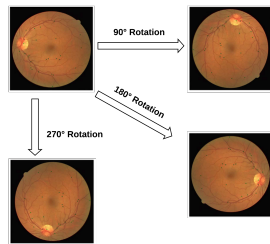


Fig. 7. Retinal Image Rotation.

- 2) **Flipping:**  
The second method, involves flipping the images horizontally and vertically, shown below in Fig. 8.

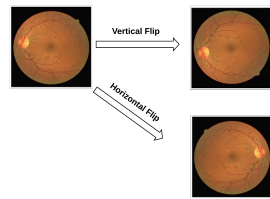


Fig. 8. Retinal Image Flipping.

- 3) **Resize & Crop:**  
The final technique starts by resizing the images into “384 x 256” images, to maintain the aspect ratio (preventing any loss of data), and then begins cropping random windows of size “224 x 224” to create new dataset samples, shown below in Fig. 9. This technique offered a wide variety of new possibilities than the previous ones, and proved to be the most effective.

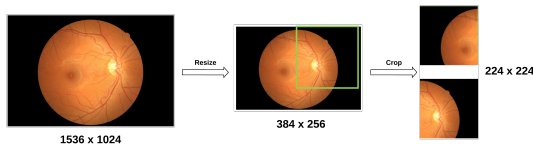


Fig. 9. Retinal Image Resizing and Cropping.

- 4) **Conclusion:**  
After thoroughly testing out the three different sampling techniques, it was apparent that the “Resize &

Crop” method provided the best results, shown below in Fig. 10.

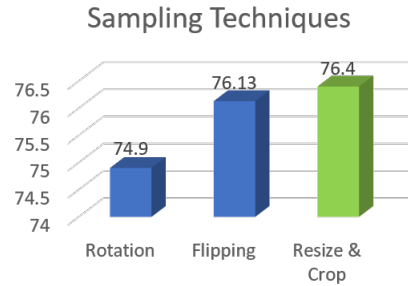


Fig. 10. Sampling Techniques Accuracies.

## B. Convolution Neural Network

A Convolutional Neural Network (CNN) is one of the most famous, dominant methods nowadays in object detection and classification, since it doesn't require any feature extraction or segmentation processes, and because it is better at working with enormous datasets. Therefore, a comparison between the distinctive CNN models and methods was conducted, to compare their results and select the most efficient and reliable one out of them.

## IV. EXPERIMENTAL SECTION

The purpose of this experimental section is previewing and discussing the performance and the results of all the trials that have been accomplished throughout this project. To ensure that we get the best output and performance out of our system, we regularly compared and analyzed the results of our trials, so as to be able to figure out the ideal setup and build a more reliable system. Furthermore, the following comparisons were mainly focused on two main aspects; the time taken, and the accuracy achieved. Finally, our conclusion was mainly based on choosing or selecting the parameter that achieved the most reasonable accuracy relative to the time taken.

**The following trials represent ten of our most effective trials, with respect to different parameters:**

### A. Dataset Trials

#### 1) Images Dimensions:

a) *Description*: This trial aims to illustrate the most suitable images dimensions to use while training the model. This experiment was conducted using three different image dimensions (“224 x 224”, “384 x 256”, “512 x 512”) to conclude the ideal image size that will be used in resizing the training images.

b) *Result*: As shown in Table III, there was a huge difference between the output of the third trial and the rest of the trials regarding the time taken. Although the “512 x 512” image sizes achieved slightly the highest accuracy, the time taken was excessively large which is not worth it at all. So, “224 x 224” seems to be the best image size to be used due to its low time taken and its reasonable accuracy.

TABLE III. IMAGES DIMENSIONS

Images Dimensions	Time Taken	Accuracy
224 x 224	1:51:47	75.53%
384 x 256	3:27:04	75.13%
512 x 512	13:56:45	76.14%

TABLE V. BASE MODELS

Base Model	Time Taken	Accuracy
VGG16	4:22:09	76.40%
VGG19	2:30:44	71.69%
Inception V3	1:50:25	68.00%
ResNet50	2:17:57	62.83%

## 2) Sampling Techniques:

*a) Description:* This trial aims to illustrate the most suitable sampling technique to use while training the model. This experiment was conducted using three different sampling techniques (“Rotation”, “Flipping”, “Resize & Crop”) to conclude the most suitable technique to be used in our case. Starting with the “Rotation” technique which depends on rotating the images (90, 180, 270) degrees in generating new image samples. “Flipping” technique performs similarly as well by flipping the images (horizontally, and vertically). Finally, the “Resize & Crop” technique which starts by resizing the images into “384 x 256” images to maintain the aspect ratio (to prevent any loss of data) and then begins cropping random windows of size “224 x 224” to create new samples. This technique offer a wide variety of possibilities than the previous ones.

*b) Result:* As shown in Table IV, there wasn’t a huge difference between the output of these trials. As the “Resize & Crop” technique achieved both the least time taken and the highest accuracy in addition to that it did not cause our model to overfit unlike the other techniques. So, it seems to be the best choice when data sampling is needed.

TABLE IV. SAMPLING TECHNIQUES

Sampling	Time Taken	Accuracy
Rotation	2:48:41	74.97%
Flipping	3:37:13	76.13%
Resize & Crop	2:46:26	76.43%

## B. CNN Setup Trials

### 1) Base Model:

*a) Description:* This trial aims to illustrate the most suitable base model to use while training the model. This experiment was conducted using four different base models (“VGG16”, “VGG19”, “Inception V3”, “ResNet50”).

*b) Result:* As shown in Table V, there was a huge difference between the output of these trials. In terms of time taken the “Inception V3” model achieved the best timing while its accuracy was considered as the lowest. However, the “VGG16” trial achieved the highest time taken in addition to being the best performing model in terms of accuracy, with a large difference compared to others. So, we believe that the “VGG16” model is the best solution as the output is worth the time taken.

### 2) Model Weights:

*a) Description:* This trial aims to illustrate whether it is better to use “Imagenet” weights while training the model or to train the model from scratch. This experiment was conducted once without any weights and once using “Imagenet” weights to compare between both results.

*b) Result:* As shown in Table VI, there was a huge difference between the output of these trials, as training from scratch achieved a lower accuracy and took a longer time than using the “Imagenet” weights while training the model, which is obviously the best choice in this case.

TABLE VI. MODEL WEIGHTS

Weights	Time Taken	Accuracy
None	15:06:13	62.50%
Imagenet	4:22:09	76.40%

### 3) Layers Freezing Techniques:

*a) Description:* This trial aims to illustrate the most suitable layer freezing technique to be used while training the model. This experiment was conducted using two different techniques (“Without freezing layers”, “Freezing layers”). Starting with the “Without freezing layers” approach in which no layers are frozen and the whole model is trained at once, unlike the second approach “Freezing layers” in which the base model layers are frozen at the beginning of the training process and later on unfreezed while training.

*b) Result:* As shown in Table VII, there was a huge difference between the output of these trials, as the “Freezing layers” approach achieved a higher accuracy and took less time than the “Without freezing layers” approach which is obviously the best choice in this case.

TABLE VII. LAYERS FREEZING TECHNIQUES

Layers Status	Time Taken	Accuracy
No Freezed Layers	12:36:30	62.50%
Freezed Layers	4:22:09	76.40%

### 4) Batch Size:

*a) Description:* This trial aims to illustrate the most suitable batch size to use while training the model. This experiment was conducted using three different batch sizes (“16”, “32”, “64”).

b) *Result:* As shown in Table VIII, there wasn't a huge difference between the output of these different batches. In terms of time taken the "64" batch size achieved the best timing, However, the "32" trial achieved a better accuracy with a minor increase in the time taken than the "64" trial. So, accordingly the "32" batch size was considered the best one.

TABLE VIII. BATCH SIZE

Batch Size	Time Taken	Accuracy
16	4:21:57	75.66%
32	4:22:09	76.40%
64	4:18:48	74.08%

### 5) Images Distribution within Batch:

a) *Description:* This trial aims to illustrate the most suitable images distribution within the Batch to be used while training the model. This experiment was conducted using three different distributions ("Consecutive", "Batch", "Block"). Starting with the "Consecutive" approach in which an image from each class is added to the batch in the following order (ex: 012012012...). The "Batch" approach depends on filling the whole batch with images from the same class (ex: 111111111...). Finally, the "Block" approach in which all images from the same class are introduced together before moving on to another class (ex: 0000... - 1111... - 2222...).

b) *Result:* As shown in Table IX, there wasn't huge difference between the output of these trials. However, the "Consecutive" approach achieved the highest accuracy and best time taken. So, using the "Consecutive" approach seems to be the best way to get the best out of the trained model.

TABLE IX. IMAGES DISTRIBUTION WITHIN THE BATCH

Images Distribution	Time Taken	Accuracy
Consecutive	4:22:09	76.40%
Batch	5:47:09	74.91%
Block	5:42:43	75.46%

### 6) Optimizers:

a) *Description:* This trial aims to illustrate the most suitable optimizer to use while training the model. This experiment was conducted using four different optimizers ("adam", "adagrad", "adadelata", "RMSprop") to conclude the most suitable optimizer to be used in our case.

b) *Result:* As shown in Table X, there wasn't a huge difference between the output of these trials. Although the "adagrad" optimizer achieved slightly the lowest time taken, its accuracy wasn't the best. So, we believe that "adam" is the best choice to be used due to its high accuracy; taking into consideration that its timing was close to "adagrad" timing.

TABLE X. OPTIMIZERS

Optimizers	Time Taken	Accuracy
adam	4:22:09	76.40%
adagrad	4:11:08	75.18%
adadelata	5:33:42	74.98%
RMSprop	4:12:21	75.96%

### 7) Number of Epochs:

a) *Description:* This trial aims to illustrate the most suitable number of epochs to use while training the model. This experiment was conducted using four different epochs numbers ("5", "10", "15", "20").

b) *Result:* As shown in Table XI, there wasn't huge difference between the output of these trials specially for the output accuracy. In terms of time taken the "5" epochs model achieved the best timing, although its accuracy was not the best. However, the "10" epochs trial achieved a slightly higher accuracy, but the time taken was over the double of the "5" epoch trial. So, its obviously clear that "5" epochs is the best solution to be used with respect to these results.

TABLE XI. EPOCHS

No. of Epochs	Time Taken	Accuracy
5	4:22:09	76.40%
10	10:23:08	76.56%
15	12:51:24	75.68%
20	16:20:22	75.55%

### C. CNN Architecture

a) *Description.:* Regarding our CNN design there were two architectures to choose between with respect to their results. First, a "Cascaded Architecture" which consists of two consecutive models "Yes-No Model" which is mainly responsible for detecting the presence of the disease, and "Stages Model" which specify the level of the detected disease by the first model. Second, a one model "5-Stages Architecture" that classifies the presence and the level of disease at once.

b) *Result.:* Fig. 11 and Fig. 12 represents the confusion matrix of the "Cascaded Architecture" and the "5-Stages Architecture", respectively. To evaluate the performance of the two architectures it was decided to calculate the Accuracy, Sensitivity, and Specificity for each of them.

	0	1	2	3	4
0	4867	24	375	4	52
1	1654	3174	189	197	108
2	2614	843	1167	382	316
3	163	969	305	2722	1163
4	87	318	188	1115	3614

Fig. 11. Cascaded Architecture

	0	1	2	3	4
0	4745	1	518	3	55
1	1624	2905	230	426	137
2	2213	670	1578	544	317
3	119	605	370	3278	950
4	63	157	211	1322	3569

Fig. 12. 5-Stages Architecture

As shown below is Table XII, the “5-Stages Architecture” achieved better results in the three metrics and so this architecture was chosen to be used in our proposed system.

TABLE XII. PERFORMANCE EVALUATION

	Accuracy	Sensitivity	Specificity
Cascaded Architecture	83.37%	58.41%	89.6%
5-Stages Architecture	84.16%	60.41%	90.1%

#### D. Conclusion

To sum it up, these experiments were applied to monitor the effect of applying various changes to the model’s parameters, in addition to analyzing the output results and use them in building a well trained model that will help in providing more reliable results. The below Tables XIII and XIV state the final parameters used and the final results of the system.

TABLE XIII. FINAL MODEL SETUP 1/2

Images Dimensions	Sampling Technique	Base Model	Model Weights	Freezing Technique	Batch Size	Images Distribution
224 x 224	Resize & Crop	VGG16	Imagenet	Freezed Layers	32	Consecutive

TABLE XIV. FINAL MODEL SETUP 2/2

Optimizers	Epochs	Train/Test Percentage	Model Architecture	Time Taken	Accuracy
adam	5	70/30	5-Stages	2 Days, 7 Hrs	84.16%

#### V. CONCLUSION AND FUTURE WORK

The aim of this proposed system is to be able to automatically detect and classify the various Diabetic Retinopathy stages using a “5-Stages” model architecture, in which deep learning mainly depends on raw colored Retinal Fundus images as its source of input. This system was tested over a total of 26,610 images, which represents almost 30% of the given dataset; after being trained over 62,090 images. As an output, the system achieved an overall accuracy of 84.16% for detecting the presence of the disease and determining its stage. Although, its clear that using these techniques provides a better output compared to the usual machine learning techniques, however, it still requires some extra work to improve these results.

A future work for this paper will be mainly concerned with testing the trained model against real data that has a wide range of variation, to prove its reliability and make sure that this solution is ready to be implemented on real life Diabetic Retinopathy patients. It may also be taken into consideration trying other models, which may offer better results compared to the current ones.

#### ACKNOWLEDGMENTS

The work done in this research was aided and hugely supported by Dr. Dina Hossam, an assistant professor of ophthalmology at Cairo university. She also offered her assistance with all the needed domain information and knowledge. The authors wish to express their gratitude and appreciation for her huge support.

#### REFERENCES

- [1] Diabetes, (2019, November 25). © 2020 WHO. <http://www.who.int/health-topics/diabetes>.
- [2] Ryan Lee, Tien Y Wong, and Charumathi Sabanayagam. Epidemiology of diabetic retinopathy, diabetic macular edema and related vision loss. *Eye and vision*, 2(1):17, 2015.
- [3] Diabetes facts and figures, (2019, December 13). © 2020 International Diabetes Federation. <https://idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html>.
- [4] Indronil Bhattacharjee and Tareq Mahmud. *Diabetic Retinopathy Classification from Retinal Images using Machine Learning Approaches*. PhD thesis, 02 2020.
- [5] Shailesh Kumar and Basant Kumar. Diabetic retinopathy detection by extracting area and number of microaneurysm from colour fundus image. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 359–364. IEEE, 2018.
- [6] Fernanda Cisneros-Guzmán, Saúl Tovar-Arriaga, Carlos Pedraza, and Arturo González-Gutierrez. Classification of diabetic retinopathy based on hard exudates patterns, using images processing and svm. In *2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI)*, pages 1–5. IEEE, 2019.
- [7] Handayani Tjandrasa, Ricky Eka Putra, Arya Yudhi Wijaya, and Isye Arieshanti. Classification of non-proliferative diabetic retinopathy based on hard exudates using soft margin svm. In *2013 IEEE International Conference on Control System, Computing and Engineering*, pages 376–380. IEEE, 2013.



- [8] Enrique V Carrera, Andrés González, and Ricardo Carrera. Automated detection of diabetic retinopathy using svm. In *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pages 1–4. IEEE, 2017.
- [9] Surbhi Sangwan, Vishal Sharma, and Misha Kakkar. Identification of different stages of diabetic retinopathy. In *2015 International Conference on Computer and Computational Sciences (ICCCS)*, pages 232–237. IEEE, 2015.
- [10] Chunyan Lian, Yixiong Liang, Rui Kang, and Yao Xiang. Deep convolutional neural networks for diabetic retinopathy classification. In *Proceedings of the 2nd International Conference on Advances in Image Processing*, pages 68–72, 2018.
- [11] Nor Hazlyna Harun, Yuhani Yusof, Faridah Hassan, and Zunaina Embong. Classification of fundus images for diabetic retinopathy using artificial neural network. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 498–501. IEEE, 2019.
- [12] Yung-Hui Li, Nai-Ning Yeh, Shih-Jen Chen, and Yu-Chien Chung. Computer-assisted diagnosis for diabetic retinopathy based on fundus images using deep convolutional neural network. *Mobile Information Systems*, 2019, 2019.
- [13] Uday Kiran Challa, Pavankumar Yellamraju, and Jignesh S Bhatt. A multi-class deep all-cnn for detection of diabetic retinopathy using retinal fundus images. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 191–199. Springer, 2019.
- [14] Pan Junjun, Yong Zhifan, Sui Dong, and Qin Hong. Diabetic retinopathy detection based on deep convolutional neural networks for localization of discriminative regions. In *2018 International Conference on Virtual Reality and Visualization (ICVRV)*, pages 46–52. IEEE, 2018.
- [15] Anuj Jain, Arnav Jalui, Jahanvi Jasani, Yash Lahoti, and Ruhina Karani. Deep learning for detection and severity classification of diabetic retinopathy. In *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, pages 1–6. IEEE, 2019.
- [16] Arkadiusz Kwasigroch, Bartłomiej Jarzembinski, and Michal Grochowski. Deep cnn based decision support system for detection and assessing the stage of diabetic retinopathy. In *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pages 111–116. IEEE, 2018.
- [17] Slavomír Kajan, Jozef Goga, Kristián Lacko, and Jarmila Pavlovičová. Detection of diabetic retinopathy using pretrained deep neural networks. In *2020 Cybernetics & Informatics (K&I)*, pages 1–5. IEEE.
- [18] Shorav Suriyal, Christopher Druzgalski, and Kumar Gautam. Mobile assisted diabetic retinopathy detection using deep neural network. In *2018 Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges (GMEPE/PAHCE)*, pages 1–4. IEEE, 2018.
- [19] Balazs Harangi, Janos Toth, and Andras Hajdu. Fusion of deep convolutional neural networks for microaneurysm detection in color fundus images. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3705–3708. IEEE, 2018.
- [20] Sharzil Haris Khan, Zeeshan Abbas, SM Danish Rizvi, et al. Classification of diabetic retinopathy images based on customised cnn architecture. In *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pages 244–248. IEEE, 2019.
- [21] Xianglong Zeng, Haiquan Chen, Yuan Luo, and Wenbin Ye. Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network. *IEEE Access*, 7:30744–30753, 2019.
- [22] Darwin Yi Carson Lam, Margaret Guo, and Tony Lindsey. Automated detection of diabetic retinopathy using deep learning. *AMIA Summits on Translational Science Proceedings*, 2018:147, 2018.
- [23] Eyepacs, (2020, January).
- [24] Diabetic retinopathy detection, (2019, October 25).
- [25] Yuqian Zhou and Shuhao Lu. Discovering abnormal patches and transformations of diabetics retinopathy in big fundus collections. pages 195–206, 01 2017.