IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DIABETIC RETINOPATHY MICROANEURYSMS DETECTION USING PRETRAINED CONVOLUTIONAL NEURAL NETWORKS

[1] V. Prema, [2] V. Ripponika, [3] V. Rishikesh, [4] M.P. Roshan

[1] Assistant Professor, [234] UG Scholar

[1234] Department of Computer Science and Engineering,

SRM Valliammai Engineering College, Kattankulathur, Chengalpattu District, India.

ABSTRACT- Diabetic retinopathy is a major cause of blindness worldwide, affecting millions of people. Early detection and treatment of diabetic retinopathy can prevent severe vision loss. In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have demonstrated exceptional performance in various image recognition tasks, including medical image analysis. In this context, this project aims to develop a CNN-based model for detecting diabetic retinopathy in retinal images. The proposed approach involves collecting a dataset of retinal images labelled for the presence or absence of diabetic retinopathy, pre-processing the data, selecting a suitable CNN architecture, training the model on the data, and evaluating its performance. The anticipated outcome of this project is a CNN model that can accurately and reliably detect diabetic retinopathy in retinal images, which can assist in early diagnosis and treatment of this severe condition.

***Keywords* – Diabetic retinopathy; Convolutional Neural Networks; fundus image; dataset; early detection; ensemble.**

I. INTRODUCTION

Diabetic Retinopathy (DR) is a leading cause of blindness in the developed world. Diabetes is expected to increase further due to growing life expectancy, decadent lifestyles, and other contributing factors. Screening diabetic patients for DR on a regular basis has been proved to be a cost-effective and vital element of their therapy. The accuracy and timing of this care are critical to both the cost and effectiveness of treatment. Effective therapy for DR is available if discovered early enough, making this a critical process.

The weighting of multiple features and the placement of such features are involved in the classification of DR. This takes up a lot of time for clinicians. Once educated, computers can perform substantially faster classifications, allowing them to assist clinicians in real-time classification. The efficacy of automated grading for DR has been a hot topic in computer imaging research, with promising results. Significant research has been conducted to detect DR features using automated methods such as support vector machines and k-NN classifiers. Many of these classification systems are based on two classes for Diabetic Retinopathy or no Diabetic Retinopathy.

Convolutional Neural Networks (CNNs), a subset of deep learning, have a strong track record for image processing and interpretation, including medical imaging. Network architectures designed to operate with picture data were commonly constructed in the 1970s and outperformed other techniques to difficult jobs such as handwritten character recognition. However, it wasn't until many advances in neural networks, such as the incorporation of dropout and rectified linear units, as well as the concomitant rise in computational capacity via graphical processor units (GPUs), that they became practical for more sophisticated image identification applications. Large CNNs are currently being utilized to successfully tackle highly complicated picture identification jobs with a wide range of object classes. CNNs are currently employed in numerous cutting-edge image classification tasks, including as the biennial ImageNet and COCO challenges.

There are two major concerns with automatic grading, particularly CNNs. One goal is to achieve a suitable balance between sensitivity (patients correctly recognized as having DR) and specificity (patients correctly identified as not having DR). This is substantially more difficult for national standards, which divides the problem into five classes: normal, mild DR, moderate DR, severe DR, and proliferative DR. Furthermore, overfitting is a significant problem in neural networks. Skewed datasets cause the network to overfit to the class that appears most frequently in the dataset. Large datasets are frequently biased. Less than 3% of the images in the dataset we utilized were from the 4th and 5th classes, requiring adjustments to our network to ensure it could still learn the properties of these images.

In this study, we present a deep learning-based CNN technique for classifying DR in fundus imaging. As previously noted, this is a medical imaging task with increasing diagnostic value that has been the topic of numerous investigations in the past. As far as we know, this is the first study to examine the five classes of DR categorization using a CNN approach. Several innovative strategies for adapting the CNN to our large dataset are presented. We then examine the performance and capabilities of our network.

The rest of this paper is structured as follows. Section 2 provides an overview of related work, Section 3 details the CNN architecture and training methods utilized in this work, Section 4 offers the findings of our trials, and Section 5 finishes the paper with a discussion of the results and future work.

II. RELATED WORK

Extensive research on approaches for binary classification of DR has yielded promising results. Gardner et al. used Neural Networks and pixel intensity values to reach sensitivity and specificity findings for yes or no categorization of DR of 88.4% and 83.5%, respectively. They employed a small dataset of roughly 200 photos and divided each image into patches, which were then classified for characteristics by a physician before SVM implementation.

In the three-class classification of DR, neural networks have also been applied. Nayak et al17 combined textural characteristics with factors such as exudate area and blood vessel area. The neural network uses features to classify images as normal, non-proliferative retinopathy, or proliferative retinopathy. These features were fed into the neural network as input for classification. The detection results were validated by comparing them to professional ophthalmologist grading. They achieved 93% classification accuracy, 90% sensitivity, and 100% specificity. This was done on a dataset of 140 photos, and feature extraction on all images was necessary in both training and testing, which can be time demanding.

The great majority of research on five-class classification has relied on support vector machines (SVMs). Acharya et al.18 developed an automated approach for classifying the five types. The SVM classifier uses features collected from raw data using a higher-order spectrum approach to capture variance in the shapes and contours in the images.

The accuracy, sensitivity, and specificity of this SVM approach were all 82% on average. Acharya et al. also developed a five-class classification approach based on the areas of various features such as haemorrhages, micro-aneurysms, exudate, and blood vessels. The most important features, blood vessels, micro-aneurysms, exudates, and haemorrhages, were retrieved from raw pictures using image processing techniques. These were then fed into the SVM to be classified. This approach attained a sensitivity of 82%, specificity of 86%, and accuracy of 85.9%. These methods were applied to very small datasets, and the decrease in sensitivity and specificity was most likely related to the complexity of the five-class problem.

Each of the preceding five class techniques needed picture feature extraction before being fed into an SVM classifier and were only verified on tiny test sets of about 100 photographs. These methods are less applicable in real-time than a CNN.

II. METHOD AND STRUCTURE

3.1 Convolutional Neural Networks

Convolutional neural network (CNN), a type of artificial neural network that has been dominant in several computer vision tasks, is gaining popularity in a range of fields, including radiology. CNN is designed to learn spatial hierarchies of data automatically and adaptively by backpropagation using several building blocks such as convolution layers, pooling layers, and fully connected layers. The convolutional layers of a CNN are the central component, which use filters to extract features such as edges, textures, and forms from the input image. The output of the convolutional layers is then fed into pooling layers, which downsample the feature maps and retain the most critical data while decreasing the spatial dimensions. Finally, to forecast or categorise the image, one or more fully connected layers are applied to the output of the pooling layers. CNNs can be trained to recognise patterns and features associated with specific objects or classes by employing a huge collection of labelled pictures. These networks can be used to classify new images as well as for other tasks like object detection or image segmentation by extracting features. Overall, CNNs are an effective tool for image processing and analysis, and their capacity to learn and extract significant characteristics from data makes them useful in a variety of applications.

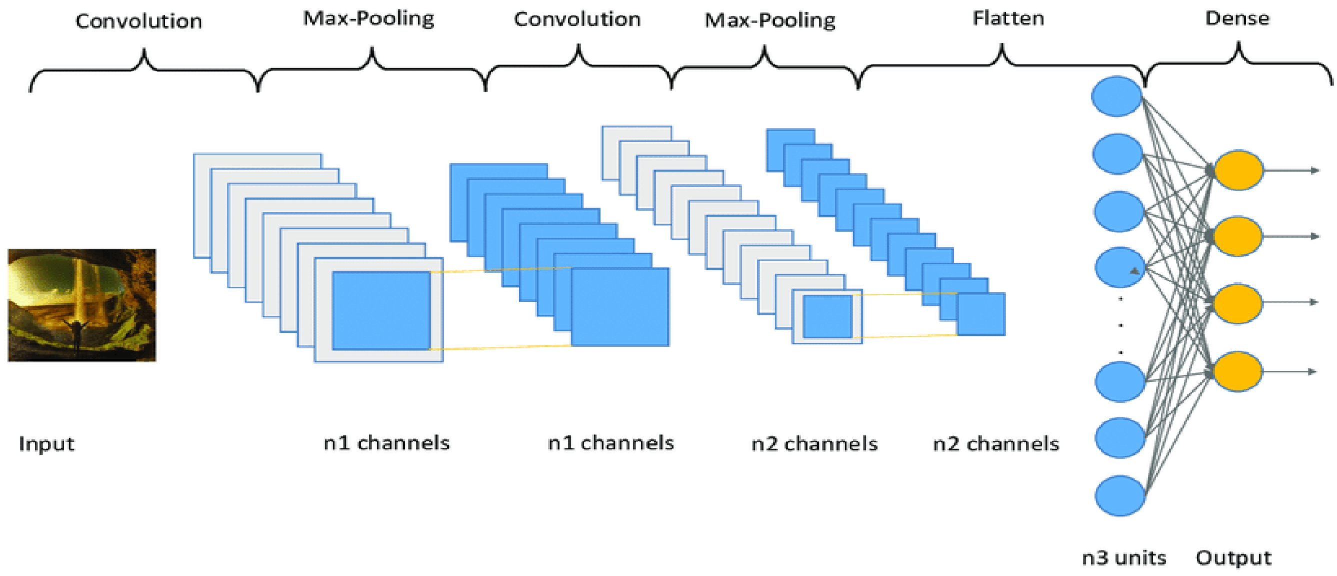


Fig 3.1: Convolutional Neural Networks

3.2 ResNet 50

ResNet stands for Residual Network and is a specific type of convolutional neural network (CNN). The bottleneck design is used for the building block in the 50-layer ResNet. A bottleneck residual block employs 11 convolutions to limit the number of parameters and matrix multiplications. This allows for significantly faster layer training. It employs a three-layer stack rather than two levels. The 50-layer ResNet architecture includes the following element:

- A 7×7 kernel convolution alongside 64 other kernels with a 2-sized stride
- A max pooling layer with a 2-sized stride
- 9 more layers 3×3,64 kernel convolution, another with 1×1,64 kernels, and a third with 1×1,256 kernels.

  These 3 layers are repeated 3 times.
- 12 more layers with 1×1,128 kernels, 3×3,128 kernels, and 1×1,512 kernels, iterated 4 times.
- 18 more layers with 1×1,256 cores, and 2 cores 3×3,256 and 1×1,1024, iterated 6 times.
- 9 more layers with 1×1,512 cores, 3×3,512 cores, and 1×1,2048 cores iterated 3 times.

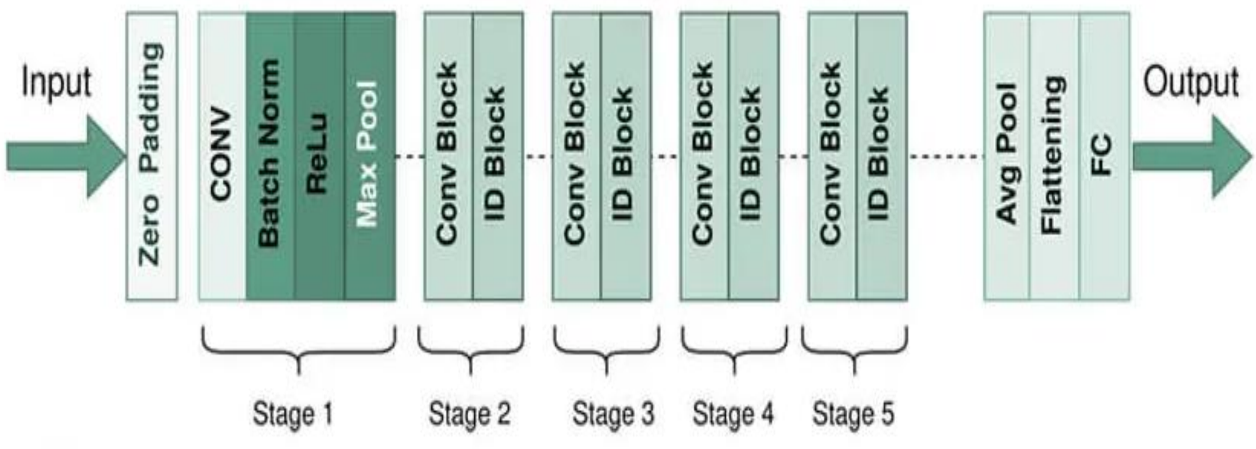


Fig 3.2: Resnet50 Model Architecture

3.3 Densenet

The term densenet refers to densely connected-convolutional networks. It is quite similar to a ResNet, with a few key distinctions. ResNet employs an additive technique, which means it uses a prior output as an input for a future layer, whereas DenseNet uses all previous output as an input for a future layer. As a result, DenseNet was created specifically to increase accuracy caused by the vanishing gradient in high-level neural networks caused by the great distance between input and output layers, and the information vanishes before reaching its goal.

3.4 EfficientNet

EfficientNet is a convolutional neural network design and scaling method that uses a compound coefficient to consistently scale all depth/width/resolution dimensions. In contrast to current practice, which arbitrarily scales these elements, the EfficientNet scaling method evenly scales network breadth, depth, and resolution using a predetermined set of scaling coefficients. The baseline network also has a significant impact on the success of model scaling. To enhance performance even further, we created a new baseline network by conducting a neural architecture search with the AutoML MNAS framework, which optimises both accuracy and efficiency. (FLOPS). The resulting architecture, like MobileNetV2 and MnasNet, employs mobile inverted bottleneck convolution (MBConv), but is significantly larger due to an increased FLOP budget. We then build up the baseline network to create the EfficientNets family of models.

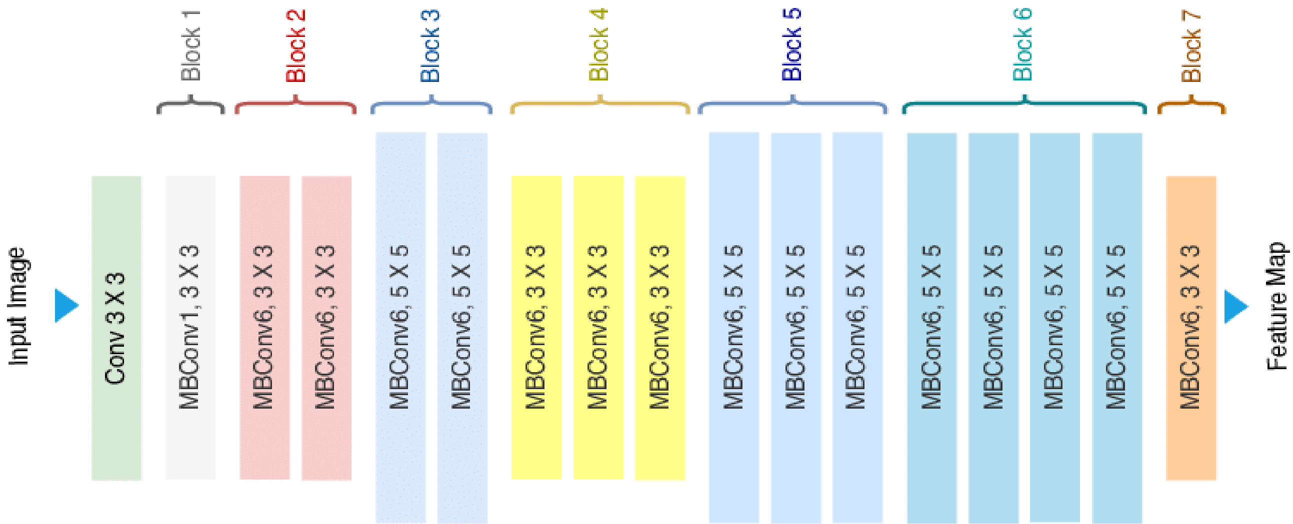


Fig 3.3: EfficientNet

3.5 Nasnet

Although the fundamental architecture of NASNet is specified as illustrated above, writers do not predefine the blocks or cells. Instead, they are discovered using the reinforcement learning search strategy. The number of motif repetitions N and the number of first convolutional filters are therefore free parameters that can be scaled. These cells are known as Normal Cells and Reduction Cells.

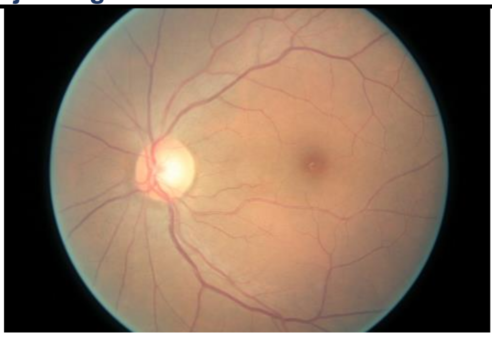
- **Normal Cell**: Convolutional cells that return a feature map of the same dimension.
- **Reduction Cell**: Convolutional cells that return a feature map where the feature map height and width is reduced by a factor of two.

3.6 Ensemble model

Machine learning ensemble approaches incorporate inputs from different learning models to make more accurate and improved conclusions. These methods follow the same logic as the previous example of purchasing an air conditioner. The main sources of error in learning models are noise, variation, and bias. Machine learning ensemble methods serve to minimise these error-causing elements, ensuring the accuracy and stability of machine learning (ML) algorithms.

3.7 Classification

After researching the literature for different image identification challenges, we agreed on the form of our neural network. Increased convolution layers are thought to allow the network to learn more detailed information. For example, whereas the first layer learns edges, the network's deepest layer, the final convolutional layer, should learn DR classification features such as hard exudate. The network begins with convolution blocks that are activated, followed by batch normalisation after each convolution layer. As the number of feature maps grows, we transition to one batch normalisation per block.

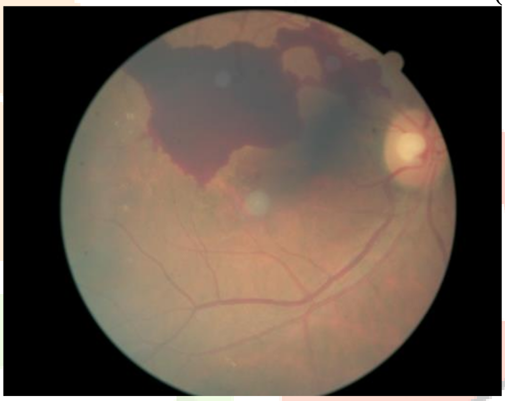


(a) No DR

(b) Mild DR

(c) Moderate DR

(d) Severe DR



(e) Proliferative DR

Fig 3.7: Diabetic retinopathy (DR) stages of increasing severity

All maxpooling is done using kernel sizes of 3x3 and strides of 2x2. The network is flattened to one dimension after the final convolutional block. We then dropout on dense layers until we reach the dense five node classification layer, which predicts our classification using a softmax activation function. To avoid over-reliance on specific network nodes, the leaky rectified linear unit activation function was utilised with a value of 0:01. Similarly, L2 regularisation was applied for weight and biases in the convolution layers. To reduce initial training time, the network was also initialised with Gaussian initialization. The categorical crossentropy function was utilised to optimise the loss function.

IV. IMPLEMENTATION

4.1 Dataset and Software

The Aptos 201 dataset was used for testing and contains about 6,000 images with around 6M pixels per image and retinopathy scales. Colab (short for Collaboratory) is a free platform from Google that allows users to code in Python. Google offers a cloud service based on Jupyter Notebook. This platform enables us to train Machine Learning models on the cloud directly and for free. Google Colab does everything your Jupyter Notebook does and more, such as allowing you to access GPU and TPU for free. Some of the benefits of Google Colab include rapid installation and real-time sharing of Notebooks amongst users. However, loading a CSV file necessitates the addition of a few lines of code. This post will go through three different methods for loading a CSV file and storing information in a pandas dataframe. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

4.2 Preprocessing

The dataset included fundus photography photos from patients of various ethnicities, age groups, and illumination settings. This has an effect on the pixel intensity values in the photos, causing excessive variance unrelated to classification levels. To combat this, the OpenCV (http://opencv.org/) tool was used to perform colour normalisation on the images. Figure shows the end consequence of this. The photographs were likewise high resolution, requiring a large amount of RAM. The dataset was downsized to 512x512 pixels, which kept the detailed features we wanted to identify while shrinking the dataset to a size that the NVIDIA K40c could handle.

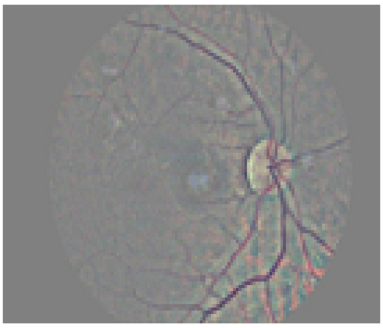


Fig 4.2: Preprocessed Image

4.3 Training

The CNN was initially pre-trained on 3,660 images until it reached a significant level. This was needed to achieve a relatively quick classification result without wasting substantial training time. Following table shows number of Epoch and training time per Epoch.

Table 4.3: Model training

CNN models	No of Epoch	Training per epoch (time)
ResNET50	20	713s
DenseNET	20	223s
EfficientNet	20	512s
NasNet	5	572s
Proposed model	4	1868s

4.4 Augmentation

The original pre-processed photos were only used once for network training. Following that, real-time dataaugmentation was used during training to improve the network's capacity to localize. Each image was randomly supplemented with random rotation 0-90 degrees, random yes or no horizontal and vertical flips, and random horizontal and vertical shifts for each epoch. Figure depicts the outcome of image augmentation.

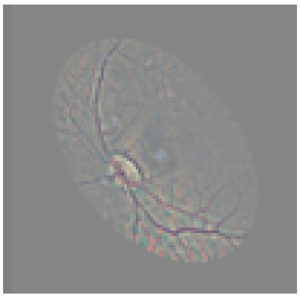


Fig 4.4: Augmented Image

V. RESULTS

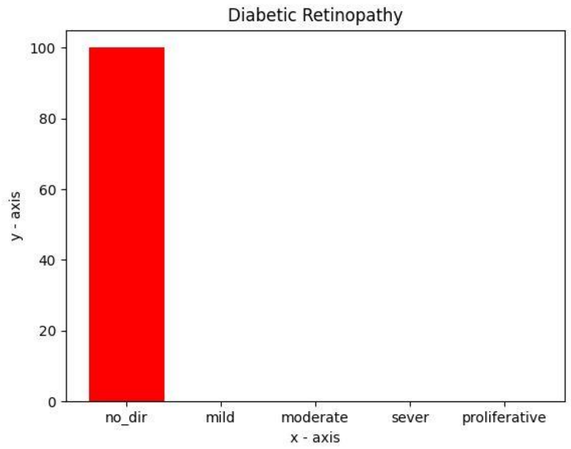


Fig 5.1: Result 1

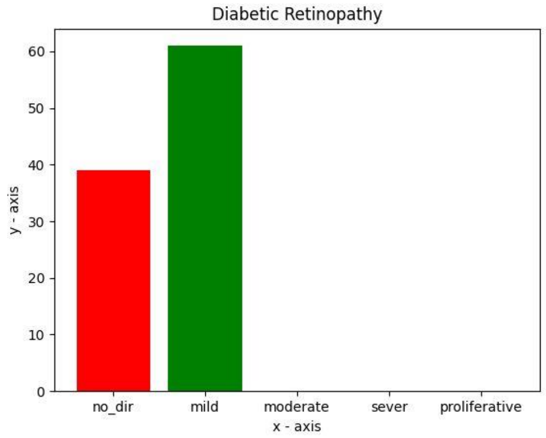


Fig 5.2: Result 2

For validation purposes, 5,500 images from the dataset were kept. It took 188 seconds to run the validation photos across the network. For this five-class problem, specificity is defined as the number of patients correctly recognised as not having DR out of the true total number of patients without DR, and sensitivity is defined as the number of patients correctly identified as having DR out of the true total number of patients with DR. We define accuracy as the number of cases correctly classified. The final trained network has a specificity of 95%, an accuracy of 75%, and a sensitivity of 30%. The network classes were quantitatively defined as follows: 0 - No DR 1 - DR Mild DR 2 - Moderate DR 3 - Severe 4 - DR proliferative.

VI. Discussion and Conclusion

Our network has showed promising signals of learning the features needed to identify from fundus images, correctly distinguishing many proliferative cases and patients with no DR. As in other large dataset research, high specificity has come at the expense of poorer sensitivity. Our method achieves comparable results to these earlier methods despite the absence of feature-specific identification and the use of a much larger dataset.

Our trained CNN has the potential benefit of being able to categorise hundreds of photos per minute, allowing it to be used in real-time anytime a new image is collected. Images are provided to physicians for grading in practice, but they are not appropriately rated when the patient comes in for screening. The trained CNN allows for a speedy diagnosis and immediate response to a patient. These results were likewise attained by the network with only one image per eye.

The network has no difficulty learning to recognise an image of a healthy eye. This is most likely because the sample contains a large number of healthy eyes. The learning required to classify photos at the extreme ends of the scale was substantially less during training. The difficulties arose when the network was tasked with distinguishing between mild, moderate, and severe cases of DR. The low sensitivity, primarily from the mild and moderate classes, suggests that the network struggled to learn deep enough features to detect some of the more intricate aspects of DR. An associated issue identified, which was certified by a clinician, was that by national UK standards, around 10% of the images in our dataset are deemed upgradeable, and these images were classified as having at least a certain level of DR.

In the future, we intend to collect a much cleaner dataset from real UK screening settings, as ongoing developments in CNNs allow much deeper networks to learn better the intricate features that this network struggled to learn. The results from our network are very promising from an orthodox network topology, as nothing specifically related to the features of our fundus images, such as vessels, exudate, and so on, have been used.

To summarize, we demonstrated that CNNs can be trained to recognise Diabetic Retinopathy characteristics in fundus images. CNNs have the potential to be extremely beneficial to DR physicians in the future when networks and datasets improve and they can provide real-time classifications.

VII. Reference

[1] Verma, K., Deep, P., and Ramakrishnan, A. (2011). “Detection and classification of diabetic retinopathy using retinal images.” 2011 Annual IEEE India Conference, IEEE, 1–6
[2] Rakhlin, A. (2018). “Diabetic retinopathy detection through integration of deep learning classification framework.” bioRxiv, 225508.
[3] Kavitha, S. and Duraiswamy, K. (2011). “Automatic detection of hard and soft exudates in fundus images using color histogram thresholding.” European Journal of Scientific Research, 48(3), 493–504.
[4] Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. (2016). “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs.” Jama, 316(22), 2402–2410.
[5] Conde, P. P., De la Calleja, J., Benitez, A., and Medina, M. A. (2012). “Image based classification of diabetic retinopathy using machine learning.” 2012 12th International Conference on Intelligent Systems Design and Applications (ISDA), IEEE, 826–830.
[6] Burges, C. J. (1998). “A tutorial on support vector machines for pattern recognition.” Data mining and knowledge discovery, 2(2), 121–167.
[7] Adem, K. (2018). “Exudate detection for diabetic retinopathy with circular hough transformation and convolutional neural networks.” Expert Systems with Applications, 114, 289–295.

[8] Carson Lam, D. Y., Guo, M., and Lindsey, T. (2018). “Automated detection of diabetic retinopathy using deep learning.” AMIA Summits on Translational Science Proceedings, 2017, 147.
[9] Ghosh, R., Ghosh, K., and Maitra, S. (2017). “Automatic detection and classification of diabetic retinopathy stages using cnn.” 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, 550–554.
[10] Kanungo, Y. S., Srinivasan, B., and Choudhary, S. (2017). “Detecting diabetic retinopathy using deep learning.” 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE, 801–804.
[11] Rajalakshmi, R., Subashini, R., Anjana, R. M., and Mohan, V. (2018). “Automated diabetic retinopathy detection in smartphone-based fundus photography using artificial intelligence.” Eye, 32(6), 1138.
[12] Silberman, N., Ahrlich, K., Fergus, R., and Subramanian, L. (2010). “Case for automated detection of diabetic retinopathy.”2010 AAAI Spring Symposium Series.