Using MobileNetV2 to Classify the Severity of Diabetic Retinopathy

Sarah Sheikh, Uvais Qidwai

Department of Computer Science and Engineering, Qatar University, Doha, Qatar.

sarahsheikh27@gmail.com; uqidwai@gmail.com

Abstract **– With the increasing number of diabetic patients, there is a need to screen patients regularly for Diabetic Retinopathy (DR) to detect it at early stages and thus prevent blindness. Existing DR screening programmes are expensive, and thus recent studies are focusing on developing effective Point-Of-Care-Technology (POCT) based screening programmes on mobile devices. Since mobile devices have limited memory and computation power it is imperative to choose light weight architectures which give a good performance. Currently the number of studies analyzing the performance of light weight mobile friendly architectures for classifying the severity of retinal images are very limited. In this paper, we propose a novel approach by using a lightweight mobile network and test the performance of our classifier built using MobileNetV2 – a lightweight, mobile friendly architecture, which is trained using retinal fundus dataset. We enhanced the retinal features using bio-inspired retinal filters and tuned the hyper-parameters to achieve an accuracy of 91.68% and AUC score of 0.9. The macro precision, recall, and f1-scores are 77.6%, 83.1%, and 80.1% respectively. Our results demonstrate that our model achieves promising results and can be deployed as a mobile application for clinical testing.**

Keywords - Diabetic Retinopathy, MobileNetV2, CNN, Retinal fundus images

I. INTRODUCTION

Diabetes, in which glucose metabolism is impaired leads to several neuropathic complications. Diabetic Retinopathy (DR) is one such neuropathic condition which is characterized by damaged blood vessels at the back of the retina. Nearly 463 million people suffer from diabetes globally [3] and nearly one third have signs of DR. Doctors have categorized DR into five different stages based on the severity viz. No DR, Mild, Moderate, Severe and Proliferative DR, characterized by symptoms shown by the retinal fundus photography images or retinal fundus images. Micro-aneurysms, Exudates, and Hemorrhages are considered indications of the presence of DR and are detected using these retinal fundus scans. In addition to that, the formation of abnormal blood vessels, called neovascularization is the characteristic for later stages of DR [4]. DR can be effectively managed in the early stages, however detection in the later stages may cause irreversible loss of vision.

Several developed countries have already put forward a well-structured screening system to effectively manage the disease and provide quality timely treatment. The adequate identification /grading of DR aids physicians in identifying suitable intervention procedures, allowing timely treatment to be administered quickly. The cost of running these screening programs is high and the lack of sufficient trained healthcare providers has forced the medical community to look for alternative ways to save time and resources in the grading of DR. Using automated computerized approach involving artificial intelligence is the current state-of- the-art to solve this issue. Artificial Intelligence (AI) is the simulation of human intelligence with the help of complex algorithms by a software/machine where the algorithms learn to detect patterns in the data and then predict/detect patterns in unseen data [5] [6].

With the rise in the users of smartphone-based technology, mobile application based retinal imaging is the need of the hour providing cheap, faster and smarter Point-Of-Care-Technology (POCT). Classifying the stages of DR using smartphone mobile technology screening systems would help generate a treatment plan for the patients, thus reducing the global disease burden and provide budget friendly, cost effective tool. Some recent studies have evaluated the performance of smartphone-based retinal imaging in the research community [7] [8] [9] and high-risk patients, will then be referred to the appropriate medical center for treatment. As the patients requiring treatment would be less than 5% of the screened patients, smartphone based automated screening tools will significantly be a stepping stone in effective management of DR.

Mobile-based AI to detect DR severity has been studied previously in a couple of studies [7] [10]. Since mobile devices have less memory capacity and less computation efficiency, and most of the state-of-the art research work focusses on using architectures which are dense, heavy and computationally expensive. In this paper we try to accomplish the task of developing a mobile based classification system for grading the severity of retinal fundus images using a much more efficient and light weight architecture of MobileNetV2 which is known to give better performance than its predecessor MobileNetV1 without compromising on the key desired characteristics of the developed model viz. low latency and increased efficiency [11] [12]. We trained and tested the model on a custom-made dataset which is an amalgamation of 3 publicly available retinal fundus images datasets viz. the EyePacs, Kaggle APTOS dataset, and the Messidor2 dataset. We used bio-inspired retinal filters and fine-tuned the hyperparameters to achieve promising results. To the best of our knowledge our technique is novel in terms of the dataset and the preprocessing steps used to build the build, thereby contributing to the field.

The following sections are organized as follows: Section II highlights the related work in the domain of DR grading using convolution neural network especially transfer learning giving a brief insight into MobileNetV2 architectures used in related studies. Section III presents the materials and methods used i.e. the dataset, pre-processing methods employed to the digital fundus images of our custom dataset, the details of the MobileNetV2 architecture, and the hyperparameter tuning of our model. Section IV presents the experimental settings, results and discussion Section V presents the conclusions and future works.

II. LITERATURE REVEIW

A plethora of recent studies are targeting to solve the problem of DR grading via Machine Learning (ML) Algorithms. Among these studies there are two broad categories of research viz. Traditional approaches and end-to-end methods. The traditional methods have each of the various feature extraction, training and classification phases to be distinct and separate. Each of the phases are manually designed and thus are prone to be incomplete at times, or over specified at other times, or they require a long time and experience to design and validate them. For example, the feature extraction phase depends on one or more of the DR symptoms viz. exudates, microaneurysms, hemorrhages, etc. and the techniques used to extract these features are heavily varied in traditional methods. Some studies [13] [14] extracted exudates as features to classify the retinal images while some others used number of microaneurysms [15] yet some others focused on retinal blood vessels [16] to identify DR. For example, Nijalingappa and Sandeep [4] used traditional methods to detect Microaneurysm, which are localized capillary dilations in circular shape and appear like small red dots often in clusters in digital fundus images on a local dataset and achieved an accuracy of 87%. Seoud et al. [17] conducted experiments for detecting red lesions (i.e., Microaneurysm and Hemorrhages) in 2016 and achieved a sensitivity of 96%. They used a traditional model based on both Random Forests and Decision Trees algorithms. In [18], Support Vector Machines (SVMs) have been used to detect and classify the exudates and predict the severity of DR. In several studies multiple image processing techniques are being used to extract the features. Rakshitha et al. [19] used image transformation methods like contourlet transform, curvelet transform and wavelet transform and compared the performances of the classifiers. Another study [20] experimented with discriminant texture features to detect DR. Most of these image processing techniques aim at identifying the lesions (exudates, hemorrhages, microaneurysms etc.) and the blood vessels to detect DR. These studies however, require domain experts and/or the medical doctors to contribute their knowledge and experience in the feature selection and extraction process, which sometimes varies vastly and is often a time-consuming process.

Current trends in the literature points towards the Deep Learning (DL) based systems which particularly use Convolution Neural Networks (CNN's) and depend on pixel locations, correlations between the pixels, the color intensities and/or the combination of these features to create more sophisticated features are outperforming the traditional approaches. Deep Learning algorithms are being used for classification either from scratch or for transfer learning where the algorithms transfer the weights from a pre-trained model to adjust to our learning dataset. Maninis et al. [21] used VGG-16 for transfer learning for optic disk and blood vessel segmentation and Mohammadian et al. [22] used the Inception-V3 algorithm and fine-tuned the parameters and also used the Xception pre-trained models and achieved very promising results. In addition to balancing the dataset they used data augmentation techniques and achieved an accuracy of 87.12% using Inception-v3 algorithm. The study published in [23] demonstrated how transfer learning could solve the issue of insufficient training data. The authors in this paper use transfer learning for retinal vessel segmentation. Along similar lines studies like [1] [24] [25] used transfer learning and in [25] the best model was developed using VggNet-16 which achieved a 78.3% accuracy. Hence, we see a number of recent studies are targeting transfer learning to solve the inadequacy of data in the domain and also reducing the training time. In addition to this, these transfer learning algorithms have very time-consuming training process and are usually computationally expensive.

To overcome the hurdles of latency and low computational efficiency, there is a need to shift to architectures which are fast and computationally efficient such as MobileNet's. MobileNet's are based on depth wise separable convolutions [11] which reduce the number of computations and thus make them computationally efficient networks, thereby improving speed. They are useful networks for mobile devices which run on limited memory access and computation power. Smartphone based Point-Of-Care-Technology (POCT) has been studied by Rajalakshmi et. al. [7] and Xu et. al. [10] which have evaluated the systems which use retinal fundus images and validated them against ophthalmologists grading. These systems are coupled with a fundus camera hardware on the smartphones for DR grading. In [26] the authors experimented with MobileNet and MobileNetV2 among other transfer learning approaches on the problem of DR grading. They used MobileNetV2 which is an improvement of the MobileNetV1 architecture in terms of speed and computational efficiency and also gave good performance [12]. They experimented on the publicly available DR dataset viz. Kaggle and Messidor retinal fundus datasets and achieved an accuracy of 58.3% using the MobileNet architecture and an accuracy of 78.1% using the MobileNetV2 architecture. Based on the related literature in the domain we lay forth the foundations of our study where we experiment the performance of MobileNetV2's latest architecture on DR retinal fundus dataset coupled with several image preprocessing techniques and fine-tuning of the hyperparameters which has resulted in improved results.

III. MATERIALS AND METHODS

In this section we present the materials and methods that we used to conduct this research. The section mainly highlights the dataset, the preprocessing techniques, the model architecture, the hyperparameter used and the performance metrics.

A. Dataset

We prepared a custom dataset using 3 different publicly available datasets to train and test our model. The final dataset that we used in our experiments is an amalgamation of retinal fundus images from the three datasets mentioned below:

- EyePacs dataset: EyePacs has provided a large dataset of retinal images from diabetic screening programmes [27]. The dataset is sponsored by the California Health care Foundation and was used in the Kaggle DR Detection Challenge. The dataset available from Kaggle consists of 35,000+ high resolution images acquired with a variety of fundus camera. The dataset is labelled with a severity grade of DR from 0 to 4.
- APTOS 2019 dataset: APTOS stands for Asia Pacific Tele-Ophthalmology Society [28]. It is a subset of the EyePacs dataset obtained after performing some preliminary operations and available in different image formats. It consists of 3000+ images.
- Messidor2 dataset: The Messidor2 is an extension of the original Messidor dataset for DR [29]. It contains 1500+ retinal fundus images which are labelled with 4 classes from 0 to 4.

The classes in the datasets are as follows:

0 - Negative or No DR: Patient has no disease.

1 - Mild DR (Stage 1): Patient has mild level of disease.

2 - Moderate DR (Stage 2): Patient has moderate level of disease.

3 - Severe DR (Stage 3): Patient has severe level of disease, the most part of the retina is damaged, can lead to complete blindness.

4 - Proliferative DR (Stage 4): Patient has proliferative level of disease. The patient's eye is damaged to an extent where treatment is elusive, about 80 percent of blindness exists.

B. Preprocessing

We applied image preprocessing techniques to improve the quality of the images or to enhance the images. We changed the luminous intensity of the image which altered the brightness and made the details of the image more visible. The operations that we applied are changing alpha, beta and gamma channels which are the important channels which control the amount of the light. We checked for images which were too dark or over-bright and could be augmented by altering the alpha, beta and gamma channels and controlled the light by using *alpha = 2.5, beta = 40 and gamma = 1.44.* These values were obtained using trial and error method on several dim images using OpenCV's ConvertscaleAbs function [30].

In order to transform the image such that texture analysis can be performed with an enhanced signal to noise ratio and provide better luminance ranges to the input images we used OpenCV's Bioinspired_retina function on some of the retrieved dim images and then augmented some particular parameters [31]. We particularly altered the default values of the following parameters in the function using retina configurations:

- photoreceptorsLocalAdaptationSensitivity: 0.69
- photoreceptorsTemporalConstant: 8.9999997615814209e-01
- photoreceptorsSpatialConstant: 5.2999997138977051e-01
- horizontalCellsGain: 0.75
- hcellsSpatialConstant: 7.0
- ganglionCellsSensitivity: 0.75

These preprocessing steps helped us in bringing up enriched features from the images, following which, we augmented our data using horizontal and vertical flips and rotation using -20 to +20 degree rotation. We were able to bring 3400+ images in each class with a total of 17,121 images in the final dataset. 80 % of the images were used for training, and 20 % we used for validation.

C. Model Architecture

Transfer learning using Image net weights on MobileNetV2 has be taken into consideration as this network is the state-of-the-art approach in the most mobile compatible networks. MobileNetsV1 are a neural network architecture which are very efficient for mobile devices. MobileNetV2 is an enhancement of MobileNetV1 and is much more efficient and powerful than its predecessor. The original MobileNetV1 is a CNN which uses depth-wise separable convolutions and basically splits the convolution layer into two sub tasks i.e. The input is filtered by a depth-wise convolution layer and then a pointwise convolution of size 1x1 combines these filtered values to create new features. These two layers are together termed as 'depth-wise separable convolution block' which performs the tasks of a normal CNN but much faster, almost about 9 times as fast as other neural networks giving about the same accuracy. The structure of the MobileNetV1 has these layers followed batch normalization and the ReLU6 activation function is used which is known to give better performance than the regular ReLU. At the end, there is a global average pooling layer, followed by a fully connected layer or a 1x1 convolution, and a softmax. The depth multiplier which is also known as the width multiplier is a hyperparameter which can be tuned and controls the number of channels in each layer [11].

According to Sandler et.al. MobileNetV2 s similar to MobileNetV1, with differences in the architecture which contribute to its effectiveness. It also uses depth-wise separable convolutions but the structure of the building block has residual connections and the expansion and projection layers, in addition. They mentioned that the block consists of three convolution layers i.e. an expansion layer in which a 1x1 convolution layer expands the number of channels, a second layer which called the depth-wise convolution layer and filters the inputs, followed by a third layer which is called the projection layer (and is a 1x1 pointwise convolution) which makes the number of channels smaller. The expansion factor gives the factor by which the data gets expanded in the expansion layer and is a hyperparameter with a default value of 6. The network also has residual connections which helps with the flow of gradients through the network. Similar to MobileNetV1, every layer has batch normalization and the activation function is ReLU6 but the output of the projection layer in MobileNetV2 does not have an activation function applied to it. The complete MobileNetV2 architecture consists of 17 such building blocks followed by a regular 1×1 convolution, a global average pooling layer, and a classification layer [12].

Sandler et al mention that MobileNetV2 performs 300 million MACs which are the multiply-accumulate operations for an RGB image of 224x224, while MobileNetV1 performs 569 million MACs [12]. Additionally, V2 has nearly 20% less parameter counts than V1 has and this explains why V2's is more computationally efficient that V1 for mobile devices which have low memory access and less computation power [12].

D. Hyperparameter Tuning

We chose a learning rate of 0.00015 where the network converged faster. We used Adam + AdaGrad as our optimizer after trying several other optimizers and used a batch size of 32 for CPU training. We used a Dropout of 0.1. We ran a grid search on number of unfreezing layers and found that when we unfreeze half of the mobilenetv2 layers the model accuracy is more and the learning is very fast and the model converges very fast. We freezed the first 80 layers and unfreezed the rest of the layers for better training. We tested unfreezing of different layers and found that unfreezing from the 80th layer learned better features from the dataset.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section we briefly explain our experimental settings and lay out the results and discussion of our work.

A. Experimental Setting

We trained our MobileNetV2 model on Linux OS with 12 GB of RAM, 10 GB swap memory having i5 processor**.** We performed data normalization on images from the 3 datasets of our custom dataset mentioned in Section III in a way that each of the 5 classes has relatively the same number of images. This was done to ensure that the dataset is not biased towards any one particular class. We used partial datasets from EyePacs and APTOS 2019 dataset and used 50% of the Messidor2 dataset to create this custom dataset for training our classifier in our experiments and we only chose images which were of a good quality and would contribute good features to the classifiers. The other 50% of Messidor2 dataset was kept aside for testing purpose. The testing dataset has 961 images belonging to the 5 classes. We trained our MobileNetV2 model, applied the preprocessing and hyperparameter tuning settings mentioned earlier and achieved an accuracy of 91.68% on the test set mainly due to the good quality images that we fed into the network. The structure of our model is give in Table I below.

TABLE I. STRUCTURE OF MOBILENETV2 FOR DR CLASSIFICATION

Input Dimension	Operator	t	c	n	s
224^2 x 3	Conv2D	-	48	1	2
112^2 x 48	Residual Module	1	24	1	1
112^2 x 24	Residual Module	6	32	2	2
56^2 x 32	Residual Module	6	48	3	2
28^2 x 48	Residual Module	6	88	4	2
14^2 x 88	Residual Module	6	136	3	1
14^2 x 136	Residual Module	6	224	3	2
7^2 x 224	Residual Module	6	448	1	1
7^2 x 448	Conv2D 1x1	-	1792	1	1
7^2 x 1792	AvgPool 7x7	-	-	1	-
1^2 x 1024	Dense	1	1024	2	1
1^2 x 512	Dense	1	512	1	1
1^2 x 5	Dense - Final	1	5	1	-

We used a width multiplier of 1.4 and an input size of 224 x 224 x 3 for each input image. Each residual module is constructed using n inverted residual blocks, where the first building block has a stride of s and the following (n − 1) building blocks have a stride of 1. c denotes the depth of the output feature map for each layer or sequence. s denotes the stride of the depth-wise Conv3x3 layer. Except for the first bottleneck layer, a constant expansion rate of t = 6 is applied throughout the network.

DOI 10.5013/IJSSST.a.21.02.16

ISSN: 1473-804x online, 1473-8031 print

B. Performance Metrics

We used a number of metrics to measure the performance of our model and have explained each one below:

- **Accuracy** Accuracy in a classification problem is the number of correct predictions made by the model over all kinds of predictions made.

Accuracy = (True Positives + True Negatives) / Total Outputs

- **Precision** Precision calculates the rate of actual positives out of those predicted positive. It is given by the formula:

Precision = True Positives / (True Positives + False Positives)

- **Recall** Recall measures the rate of actual positives over all predicted values that are actually positive. It is given by the formula:

Recall = True Positives / True Positives + False Negatives)

- **f1-score** f1-score is the harmonic mean of the precision and recall and conveys the balance between the precision and the recall. It is calculated by the formula:

f1-score = 2 * [(Precision*Recall) / (Precision + Recall)]

- **AUC** ROC which stands for Receiver Operating Characteristics curve is plotted with recall along the y-axis and the false positive rate (which is given by 1-Specificity) along the x-axis. Area under the ROC Curve also called AUC gives the degree of separability of classes which means it tells how much the model is capable of distinguishing between the classes. Higher value of AUC denotes a better model.

C. Results and Discussion

This section presents the results of our experiments and compares them to previous related work as discussions. Table II presents the confusion matrix obtained from testing the model on an unseen dataset which is 50% of the Messidor2 dataset. Table III presents the precision, recall and f1-score per class and Table IV presents the key performance measures of our MobileNetV2 classifier. The results demonstrate the model to have good generalization abilities and is compared with other related work in the literature.

TABLE II. CONFUSION MATRIX

		MobileNetV2 Predicted labels				
		0	***1***	***2***	***3***	***4***
Actual labels	***0***	576	11	14	3	2
	1	9	130	9	1	1
	2	3	5	143	7	3
	3	0	1	3	19	3
	4	1	1	1	2	13

TABLE III. PERFORMANCE MEASURE PER CLASS

	0	**1**	**2**	**3**	**4**
Precision	0.9779	0.8784	0.8412	0.5938	0.5909
Recall	0.9505	0.8667	0.8882	0.7308	0.7222
f1-score	0.9640	0.8725	0.8640	0.6552	0.6500

TABLE IV. PERFORMANCE MEASURES

Metric	**Value**
Macro Precision	77.64%
Macro Recall	83.17%
Macro f1-score	80.11%
AUC	0.9
Training accuracy	98.45%
Validation accuracy	91.90%
Testing accuracy	**91.68%**

On comparing with other related previous work, we see that our model has been able to achieve promising results. The previous related work [2] achieved 90.8% accuracy using the same architecture of MobileNetV2 but on a different dataset. We see that Gao et.al. in [2] divided the problem into a 2-class problem of DR and RDR (referable DR). According to them, grade 0 and 1 of the Messidor dataset form the DR category while grade 2 and 3 are considered the referable DR category which needs urgent attention. Their MobileNetV2 model has achieved and accuracy of 90.8% for class DR and 92.3% for class RDR. Our work on the contrary has studied the problem as a 5-class problem with classes 0, 1, 2, 3 and 4 as explained in section 3.1 and we have achieved an average accuracy of 91.68% and an AUC of 0.9 using our custom dataset for training and testing it on 50% of the Messidor2 dataset which is our unseen test set.

Another paper [32] proposed a network called Zoom-in-Net which does two tasks simultaneously i.e. it mimics the zooming on of the clinician to examine retinal images by developing attention maps and highlights suspicious regions and make predictions based on these suspicious regions and also the whole image. Due to a difference in the annotation scales used in the datasets used for the study they have also used similar technique of Gao et al [2] and transformed the problem into a binary classification task of referable vs non-referable and achieved an accuracy of 91.1% on the Messidor dataset and an accuracy of 90.5% on the EyePacs dataset. Thus, we see that the performance of our model is comparable to the state-of-the-art and can be applied in clinical settings using mobile applications for testing purposes.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have been able to achieve promising results on our DR severity classification system using a custom-made dataset with several pre-processing and image augmentation techniques. Our model used the computationally efficient architecture of MobileNetV2 which is known to be fast and computationally efficient algorithm. The model was tested on unseen data to test the generalizability of the model. Our results show that the model has been able to achieve good performance due to the various techniques that we used at every stage of the ML pipeline. In future, we aim to deploy and test the model in a smartphone and thereby test its effectiveness as a point-of-care technology for grading the severity of DR. We may also experiment with other variations of the MobileNet architecture and compare their effectiveness amongst each other and hence improve the state-of-the-art.

REFERENCES

[1] S. Sheikh and U. Qidwai, "Smartphone-based Diabetic Retinopathy Severity Classification using Convolution Neural Networks", [Accepted] in Intelligent Systems Conference (IntelliSys) 2020, Amsterdam, 2020.

[2] J. Gao, C. Leung and C. Miao, "Diabetic Retinopathy Classification Using an Efficient Convolutional Neural Network", in 2019 IEEE International Conference on Agents (ICA), 2019.

[3] "IDF Diabetes Atlas 9th edition 2019", Diabetesatlas.org, 2020. [Online]. Available: https://www.diabetesatlas.org/. [Accessed: 24-Feb- 2020].

[4] P. Nijalingappa and S. B., "Machine learning approach for the identification of Diabetes Retinopathy and its stages", in Applied and Theoretical Computing and Communication Technology (iCATccT), 2015 International Conference on. IEEE, 2015.

[5] M. Bhaskaranand et al., "Automated Diabetic Retinopathy Screening and Monitoring Using Retinal Fundus Image Analysis", Journal of Diabetes Science and Technology, vol. 10, no. 2, pp. 254-261, 2016. Available: 10.1177/1932296816628546.

[6] Walton. OB et al., "Evaluation of automated tele retinal screening program for diabetic retinopathy.", JAMA Ophthalmol, 2016. [Accessed 24 February 2020].

[7] R. Rajalakshmi, R. Subashini, R. Anjana and V. Mohan, "Automated diabetic retinopathy detection in smartphone-based fundus photography using artificial intelligence", Eye, vol. 32, no. 6, pp. 1138-1144, 2018. Available: 10.1038/s41433-018-0064-9.

[8] A. Russo, F. Morescalchi, C. Costagliola, L. Delcassi and F. Semeraro, "Comparison of Smartphone Ophthalmoscopy With Slit-Lamp Biomicroscopy for Grading Diabetic Retinopathy", American Journal of Ophthalmology, vol. 159, no. 2, pp. 360-364.e1, 2015. Available: 10.1016/j.ajo.2014.11.008.

[9] M. Ryan et al., "Comparison Among Methods of Retinopathy Assessment (CAMRA) Study", Ophthalmology, vol. 122, no. 10, pp. 2038-2043, 2015. Available: 10.1016/j.ophtha.2015.06.011.

[10] X. Xu et al., "Smartphone-Based Accurate Analysis of Retinal Vasculature towards Point-of-Care Diagnostics", Scientific Reports, vol. 6, no. 1, 2016. Available: 10.1038/srep34603.

[11] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861. 2017.

[12] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 4510–4520.

[13] S. Joshi and P. Karule, "A review on exudates detection methods for diabetic retinopathy", Biomedicine & Pharmacotherapy, vol. 97, pp. 1454-1460, 2018. Available: 10.1016/j.biopha.2017.11.009.

[14] P. Prentašić and S. Lončarić, "Detection of exudates in fundus photographs using deep neural networks and anatomical landmark detection fusion", Computer Methods and Programs in Biomedicine, vol. 137, pp. 281-292, 2016. Available: 10.1016/j.cmpb.2016.09.018.

[15] S. Kumar, and B. Kumar, "Diabetic Retinopathy Detection by Extracting Area and Number of Microaneurysm from Color Fundus Image.", 5th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, pp. (pp. 359-364)., 2018. [Accessed 4 December 2019].

[16] Avijit Dasgupta and Sonam Singh. A fully convolutional neural network based structured prediction approach towards the retinal vessel segmentation. In 14th International Symposium on Biomedical Imaging (ISBI 2017), 2017 IEEE, pages 248–251. IEEE, 2017.

[17] L. Seoud, T. Hurtut, J. Chelbi, F. Cheriet and J. Langlois, "Red Lesion Detection Using Dynamic Shape Features for Diabetic Retinopathy Screening", IEEE Transactions on Medical Imaging, vol. 35, no. 4, pp. 1116-1126, 2016. Available: 10.1109/tmi.2015.2509785.

[18] M. Haloi, et al., "A Gaussian scale space approach for exudates detection, classification and severity prediction", 2015. [Accessed 4 December 2019].

[19] T. R. Rakshitha, D. Devaraj and S.C. Prasanna Kumar, "Comparative study of imaging transforms on diabetic retinopathy images", IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE, (2016).

[20] H. H. Vo and A. Verma, "Discriminant color texture descriptors for diabetic retinopathy recognition", IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP), 2016. IEEE, (2016).

[21] K. Maninis, J. Pont-Tuset, P. Arbelaez and L. Gool, "Deep Retinal Image Understanding", in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016.

[22] S. Mohammadian, A. Karsaz and Y. Roshan, "Comparative Study of Fine-Tuning of Pre-Trained Convolutional Neural Networks for Diabetic Retinopathy Screening", in 2017 24th National and 2nd International Iranian Conference on Biomedical Engineering (ICBME), Tehran, Iran, 2017.

[23] J. Mo and L. Zhang, "Multi-level deep supervised networks for retinal vessel segmentation", International Journal of Computer Assisted Radiology and Surgery, vol. 12, no. 12, pp. 2181-2193, 2017. Available: 10.1007/s11548-017-1619-0.

[24] R. Mansour, "Deep-learning-based automatic computer-aided diagnosis system for diabetic retinopathy", Biomedical Engineering Letters, vol. 8, no. 1, pp. 41-57, 2017. Available: 10.1007/s13534-017-0047-y.

[25] S. Dutta, B. Manideep, S. Basha, R. Caytiles and N. Iyengar, "Classification of Diabetic Retinopathy Images by Using Deep Learning Models", International Journal of Grid and Distributed Computing, vol. 11, no. 1, pp. 99-106, 2018. Available: 10.14257/ijgdc.2018.11.1.09

[26] R. Sarki, S. Michalska, K. Ahmed, H. Wang and Y. Zhang, "Convolutional neural networks for mild diabetic retinopathy detection: an experimental study", 2019. Available: 10.1101/763136 [Accessed: 24-Feb-2020].

[27] [Online]. Available: https://www.kaggle.com/c/diabetic-retinopathy-detection/data. [Accessed: 24-Feb-2020].

[28] "APTOS 2019 Blindness Detection | Kaggle", Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/c/aptos2019-blindness-detection. [Accessed: 24- Feb- 2020].

[29] [Online]. Available: http://www.adcis.net/en/third-party/messidor2/. [Accessed: 24- Feb- 2020].

[30] "Operations on Arrays — OpenCV 2.4.13.7 documentation", Docs.opencv.org, 2020. [Online]. Available:https://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html#void%20convertScaleAbs(InputArray%20src,%20OutputArray%20dst,%20double%20alpha,%20double%20beta). [Accessed: 24- Feb- 2020].

[31] "OpenCV: Bioinspired Module Retina Introduction", Docs.opencv.org, 2020. [Online]. Available: https://docs.opencv.org/3.4/d2/d94/bioinspired_retina.html. [Accessed: 24- Feb- 2020].

[32] Z. Wang et.al., "Zoom-in-net:Deep mining lesions for diabetic retinopathy detection," in Proceeding of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, 2017, pp. 267–275.