HAL
open science

A deep learning-based smartphone app for real-time detection of five stages of diabetic retinopathy

Sharmin Majumder, Yaroub Elloumi, Mohamed Akil, Rostom Kachouri, Nasser Kehtarnavaz

To cite this version:

Sharmin Majumder, Yaroub Elloumi, Mohamed Akil, Rostom Kachouri, Nasser Kehtarnavaz. A deep learning-based smartphone app for real-time detection of five stages of diabetic retinopathy. Real-Time Image Processing and Deep Learning 2020, Apr 2020, San Diego, California (Online Only), United States. pp.5, 10.1117/12.2557554 . hal-02556991

HAL Id: hal-02556991

https://hal.science/hal-02556991v1

Submitted on 28 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Deep Learning-Based Smartphone App for Real-Time Detection of Five Stages of Diabetic Retinopathy

S. Majumder[a], Y. Elloumi[b,c], M. Akil[b], R. Kachouri[b] and N. Kehtarnavaz[a]

[a]Embedded Machine Learning Laboratory, University of Texas at Dallas, TX, USA

[b]LIGM, Univ Gustave Eiffel, CNRS, ESIEE Paris, F-77454 Marne-la-Vallée, France

[c]LabTIM, Faculty of Medicine, University of Monastir, Tunisia

ABSTRACT

This paper presents the real-time implementation of deep neural networks on smartphone platforms to detect and classify diabetic retinopathy from eye fundus images. This implementation is an extension of a previously reported implementation by considering all the five stages of diabetic retinopathy. Two deep neural networks are first trained, one for detecting four stages and the other to further classify the last stage into two more stages, based on the EyePACS and APTOS datasets fundus images and by using transfer learning. Then, it is shown how these trained networks are turned into a smartphone app, both Android and iOS versions, to process images captured by smartphone cameras in real-time. The app is designed in such a way that fundus images can be captured and processed in real-time by smartphones together with lens attachments that are commercially available. The developed real-time smartphone app provides a cost-effective and widely accessible approach for conducting first-pass diabetic retinopathy eye exams in remote clinics or areas with limited access to fundus cameras and ophthalmologists.

Keywords: Real-time implementation of deep neural networks on smartphones, real-time smartphone app for detection and classification of diabetic retinopathy, first-pass eye exam by smartphone app.

1. INTRODUCTION

According to the International Diabetes Federation (IDF), there are currently about 460 million adults worldwide suffering from diabetes which is the leading cause of blindness in most countries[3]. Diabetic Retinopathy (DR) is an eye retinal disease that can lead to visual impairments. DR is caused by microvascular complications of diabetes reflected by morphological changes in the eye fundus. There are five stages associated with DR. These stages correspond to no apparent retinopathy, mild non-proliferative DR (NPDR), moderate NPDR, severe NPDR, and proliferative DR (PDR), respectively.

The current practice for diagnosis of DR involves the examination of color fundus images taken from the retina by an expert eye professional. Works have been reported in the literature to make this process more automated, more time efficient, and more cost-effective, e.g.[4,5]. Due to the success of deep learning models in object detection and image classification applications[6,7], recently there have been studies using deep neural networks to diagnose DR from fundus images[1,8,9,10,11].

The medical device that is currently used to capture retina images for diagnosis of DR is fundus cameras which are expensive, bulky and not available in all clinics, in particular in remotely located areas. Nowadays, smartphones are widely available and also there exist commercially available lenses that are designed for smartphones to capture retina images, e.g., WelechAllyn PanOptic Ophthalmoscope[12], D-EYE lens[13,14,15]. A picture of two such lens is shown in Figure 1. These lenses can be easily attached to the front of a smartphone camera for capturing retina images. The use of smartphones together with such lenses enables a cost-effective and widely deployable approach for initial screening of DR at clinics where fundus cameras and ophthalmologists are not available.

Towards this objective, a real-time smartphone app based on deep learning models was previously developed in reference[1] to detect the presence of DR and other retinal abnormalities from fundus images. This paper is an extension of this previous work by first detecting the presence or absence of DR and then if DR is detected, by finding the stage associated with it. More specifically, a deep learning-based smartphone app (both iOS and Android versions) is

developed in this paper to detect the presence or absence of DR and its stages in real-time. In addition to the stage 0 of DR, meaning absence of DR or no DR, four stages of DR are detected which correspond to mild, moderate, severe, and proliferate. The app developed here enables an initial screening of DR in a cost-effective and widely accessible manner by only using a smartphone and a lens attachment. The initial screening enabled by this app will be of great benefit to those living in remote areas since a first-pass eye examination can be easily performed to refer users to an ophthalmologist.

The remainder of the paper is organized as follows. In section 2, the dataset used to train the deep neural networks for detecting and classifying DR is presented. The training and testing of the deep neural networks are covered in section 3. The deployment of the deep neural networks on smartphones is described in section 4. The real-time processing rates of the developed smartphone app are then reported in section 5 followed by the conclusion in section 6.

2. DATASET CONSIDERED

Availability of training dataset is essential for training deep neural networks to distinguish the five stages of DR including stage zero which means no DR. The two public domain datasets that incorporate a large number images of all the five stages of DR are EyePACS (Eye Picture Archive Communication System)[16] and APTOS 2019 (Asia Pacific Tele-Ophthalmology Society)[17] dataset. In this work, both of the datasets are considered.

The EyePACS dataset has 35126 retina images for both the left and right eyes. It includes the five stages of DR or the severity level 0 to 4, where label 0 indicates no DR, label 1 mild DR, label 2 moderate DR, label 3 severe DR, and label 4 proliferate DR. The dataset contains 25,810 labeled 0 DR (no DR), 2,443 mild DR, 5,292 moderate DR, 873 severe DR, and 708 proliferate DR images. For no DR (labeled 0), 10,000 images from EyePACS dataset were considered here for designing the deep neural network. Figure 2 illustrates 10 sample retina images from the EyePACS dataset (five from the left eye and five from the right eye) corresponding to the five stages of DR.

The APTOS 2019 Blindness Detection dataset contains a total of 3662 retinal images belonging to the five stages of DR similar to the EyePACS dataset. In this dataset, 1805 images belong to no DR (label 0), 370 images to mild DR (label 1), 999 to moderate DR (label2), 193 images to severe DR (label 3), and 295 images to proliferate DR (label 4).

(a) (b)

Figure 1. Pictures of commercially available lenses for smartphones: (a) WelechAllyn lens, and (b) D-EYE lens

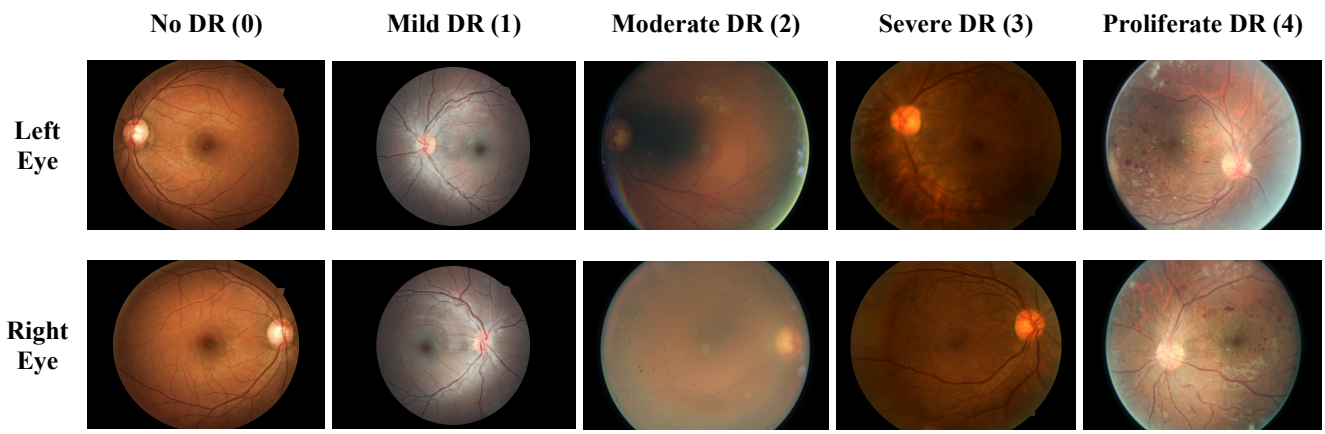

Figure 2. Sample fundus images (top five left eye, bottom five right eye) for five stages of DR from the EyePACS dataset.

3. DEEP NEURAL NETWORKS FOR DETECTION AND CLASSIFICATION OF DR

This section discusses the training and the testing of two deep neural networks designed for the purpose of separating the five stages of DR images for the above two datasets. Since the training dataset is relatively small for training deep neural networks, the transfer learning approach is utilized here.

3.1 Designing the networks

Deep learning-based methods have been previously proposed to detect ocular pathologies, e.g.[18,19,20]. For transfer learning, a pre-trained Inception-v3 model[21] is considered here similar to the work previously reported in reference[1]. This network is pre-trained with the large ImageNet dataset which contains 1.2 million images placed into 1000 different object classes[22,23]. The Inception-v3 model employs convolutional layers and residual connections. The layers of this pre-trained model are used up to the last fully connected layer. The last fully connected layer is then trained with the above datasets. A graphical representation of the transfer learning approach using the Inception-v3 architecture is shown in Figure 3.

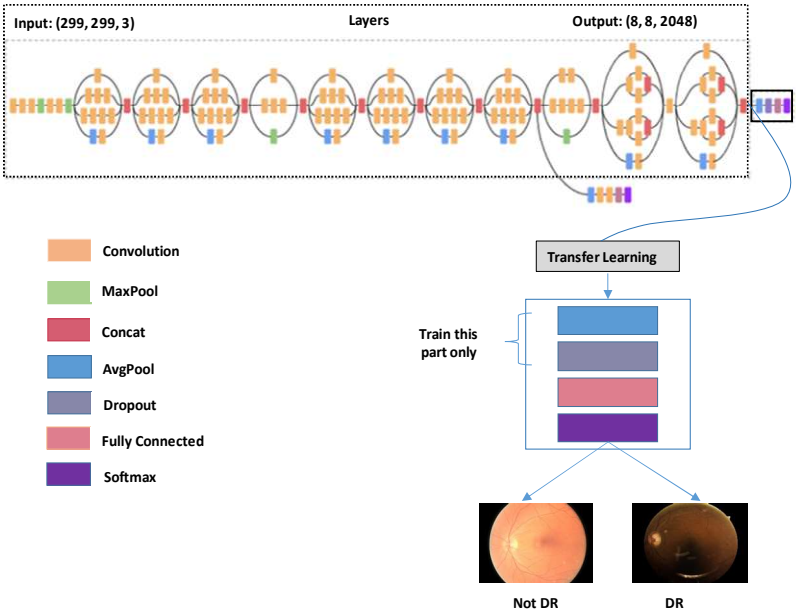

Figure 3. Depiction of Inception-v3 transfer learning

In this work, a two-step approach is considered involving two deep neural networks. For the EyePACS dataset, in the first step, a deep neural network is designed to classify the retinal images into four classes: No DR, Mild DR, Moderate DR, and Profound DR. In the second step, another deep neural network is designed to classify the last stage of Profound DR images into two further stages: Severe DR and Proliferate DR. As a result, as shown in Figure 4, a retinal image gets classified into one of the five classes: "No DR", "Mild DR", "Moderate DR", "Severe DR", and "Proliferate DR". Various combinations of networks were examined and this two-step combination was found to generate the best accuracy for the EyePACS dataset.

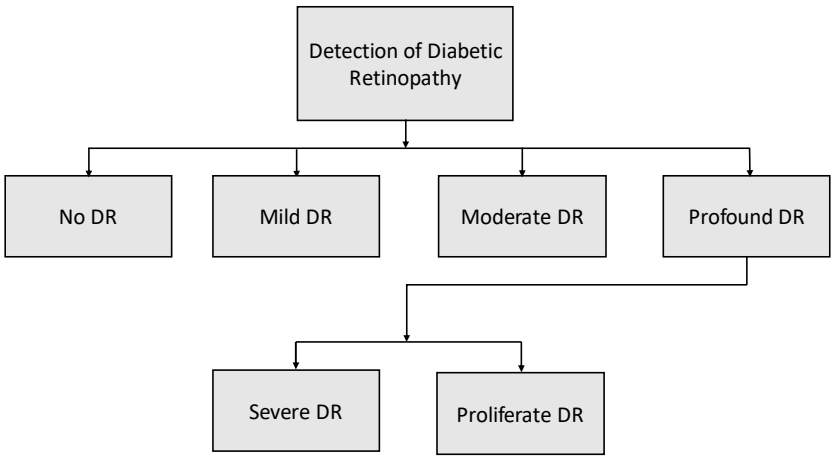

Figure 4. Two-step approach to classify five stages of DR for the EyeSPACS dataset

A two-step approach is also considered for the APTOS dataset. However, after examining different combinations of two networks, the best combination for this dataset was found to be the one depicted in Figure 5. In this combination, the absence and presence of DR is considered first and then the DR cases are further classified into four stages.

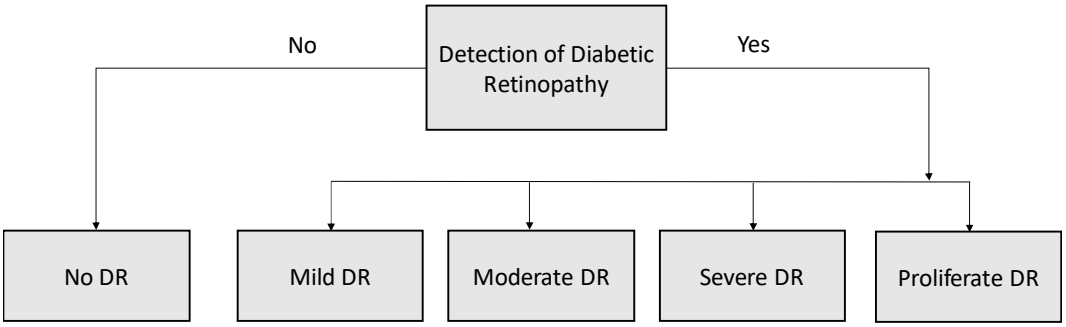

Figure 5. Two-step approach to classify five stages of DR for the APTOS dataset

3.2 Performance of the networks

EyePACS Dataset - The first deep neural network for the EyePACS dataset considered four classes, and was trained and tested on the retina images from the EyePACS dataset corresponding to no diabetic retinopathy (label 0), mild diabetic retinopathy (label 1), moderate diabetic retinopathy (label 2), and profound diabetic retinopathy (labels 3, and 4), where the severe and proliferate images of the EyePACS dataset were considered together in one stage. Randomly selected 90% of the images from each class were used for training and the remaining 10% of the images were used for testing. The confusion matrix associated with the first network is shown in Table 1. As shown in this table, the overall accuracy of the first network for the classification of four stages of diabetic retinopathy was found to be 88.4%.

Table 1. Confusion matrix of the first deep neural network for the EyePACS dataset

Actual / Classified	Not DR	Mild DR	Moderate DR	Profound DR
Not DR	994	6	0	0
Mild DR	41	201	2	0
Moderate DR	51	54	424	0
Profound DR	10	2	58	88

The second deep neural network was trained to classify Profound DR into two more stages: Severe DR and Proliferate DR. Randomly selected 90% of the images from each class, i.e. 786 images corresponding to Severe DR and 637 images corresponding to Proliferate DR, were used for training and the remaining 10% of the images, or a total of 158 images, were used for testing. The confusion matrix associated with the second network for the separation of Severe DR and Proliferate DR is shown in Table 2. As shown in this table, the overall accuracy for this second network was found to be 84.8%.

Table 2. Confusion matrix of the second deep neural network for the EyePACS dataset

Actual / Classified	Severe DR	Proliferate DR
Severe DR	78	9
Proliferate DR	15	56

To examine the overall response or accuracy, the output of the first network corresponding to Profound DR was passed to the second network. The confusion matrix for the combined response outputting the five stages of DR is shown in Table 3, indicating an overall accuracy of 87.4%.

Table 3. Confusion matrix of the combined deep neural networks for the EyePACS dataset

Actual / Classified	Not DR	Mild DR	Moderate DR	Severe DR	Proliferate DR
Not DR	994	6	0	0	0
Mild DR	41	201	2	0	0
Moderate DR	51	54	424	0	0
Severe DR	7	6	29	41	4
Proliferate DR	7	0	21	15	28

APTOS Dataset - The first deep neural network for the APTOS dataset involved separating the presence or absence of diabetic retinopathy by placing no diabetic retinopathy (label 0) in one class, and all the diabetic retinopathy stages (labels 1, 2, 3, and 4) in another class. This model was trained using randomly selected 90% of the total 3,662 images, or 1,624 images belonging to no DR and 1,671 images belonging to DR. This network was tested on the remaining 10% of the images or 367 images in the dataset. The confusion matrix associated with the first network to detect the presence or absence of diabetic retinopathy is shown in Table 4. The second network was trained to separate the DR images into four stages using randomly selected 90% of the total 1,857 DR images, which corresponded to 370 mild images, 999 moderate images, 193 severe images and 295 proliferate images. The images classified as DR in the first network are passed to the second network to evaluate the overall performance. Table 5 shows the confusion matrix of the second

network focusing on the classification of DR images into four stages. The overall confusion matrix for these two deep neural networks appears in Table 6, indicating an overall accuracy of 88.5%.

Table 4. Confusion matrix of the first deep neural network for the APTOS dataset

Actual / Classified	Not DR	DR
Not DR	158	23
DR	0	186

Table 5. Confusion matrix of the second deep neural network for the APTOS dataset

Actual / Classified	Mild DR	Moderate DR	Severe DR	Proliferate DR
Mild DR	30	6	0	1
Moderate DR	0	100	0	0
Severe DR	0	4	12	3
Proliferate DR	0	5	0	25

Table 6. Confusion matrix of the combined deep neural networks for the APTOS dataset

Actual / Classified	Not DR	Mild DR	Moderate DR	Severe DR	Proliferate DR
Not DR	158	8	15	0	0
Mild DR	0	30	6	0	1
Moderate DR	0	0	100	0	0
Severe DR	0	0	4	12	3
Proliferate DR	0	0	5	0	25

4. REAL-TIME IMPLEMENTATION ON SMARTPHONES

The implementation aspects of the deep learning models running in real-time on smartphones as apps are discussed in this section.

4.1 Deep learning and smartphone software tools

After designing the deep learning models, they are implemented on smartphones (Android and iOS) to run in real-time as per the guidelines developed by our research team in reference[2]. The steps to perform the real-time implementation of the deep learning model appear in Figure 6.

The networks or the deep learning models are developed here by using TensorFlow and Keras in Python. TensorFlow is a dataflow programming library developed by Google[24]. TensorFlow models can be easily integrated with the Android software environment. Keras[25] is a higher level library written in Python which allows TensorFlow to be used as its backend and can be extracted to implement Android apps. The use of TensorFlow Lite enables the deployment of deep learning models on mobile devices[24]. CoreML is a software framework which was developed by Apple. It allows converting trained Keras models into CoreML models[26], which can then be deployed on iOS devices.

The integrated software development environment (IDE) for Android app development is Andriod Studio which uses the Java programming language[27]. Xcode is the integrated development environment (IDE) for iOS app development which uses the Objective-C or Swift programming language[28]. All the software tools mentioned above are publicly available and are regularly updated by their respective organizations.

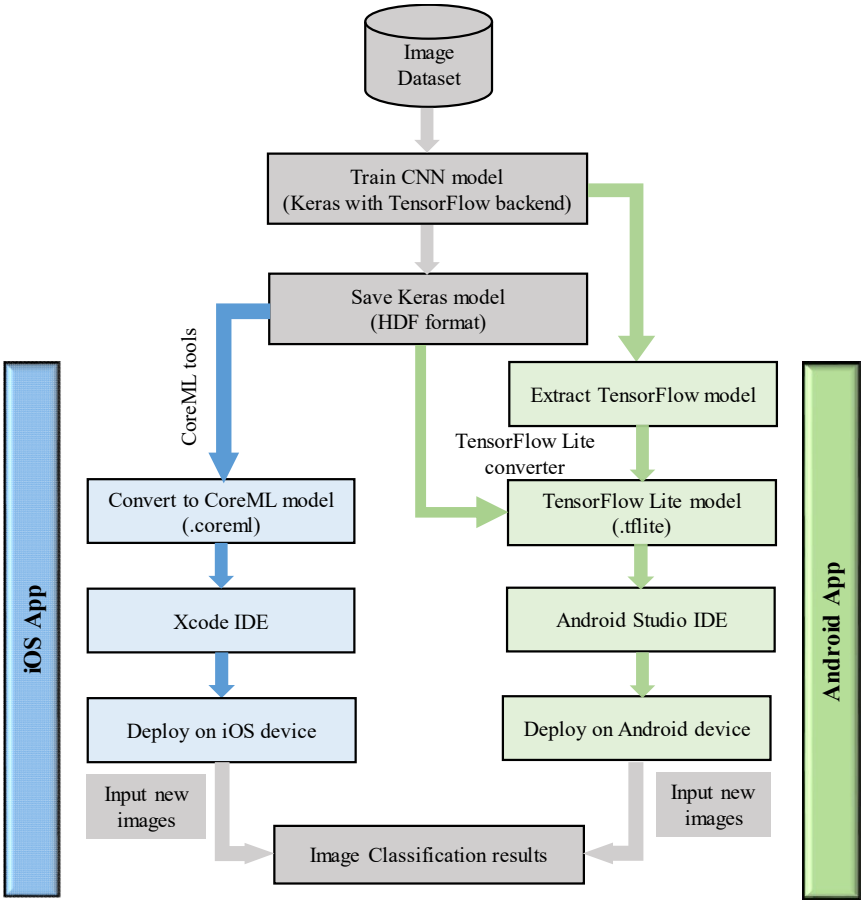

Figure 6. Steps for real-time implementation on smartphone platforms

4.2 Deployment on iOS smartphones

The deep neural network model trained in Keras is saved as an HDF (Hierarchical Data Format .h5) file format. The architecture and the connection weights of the model are saved together in the HDF file as numerical arrays. After loading the .h5 keras model into Python, the CoreML framework converts the Keras model into CoreML model using the tools provided by CoreML. This converted CoreML model is saved as an ".mlmodel" file. The conversion provides an option to define the input as an image or as a multidimensional array. If an image is specified as the input option, then another option is provided to define the pre-processing parameters used for that model. Because of the variation of pre-processing through different models, this is a useful option as it allows a model to be used on raw images without explicitly applying pre-processing for every different model. Finally, the CoreML model is deployed on an iOS device through the Xcode IDE using Swift.

4.3 Deployment on Android smartphones

At first, the deep neural network model is trained in Keras with TensorFlow backend and the TensorFlow model is extracted from the Keras model. After extracting the TensorFlow model, it is saved as a Protocol Buffer (.pb) file by using the input/output utility graph. Similar to a HDF file, the architecture and the connection weights of the model are stored in the .pb file. Then, the TensorFlow Lite interpreter is used to convert the .pb model to a .tflite model. This .tflite model is implemented on Android smartphones using Java trough the Android Studio IDE. TensorFlow Lite is used to

create a session to handle the model computations, thus making it easier for users to focus on deploying and testing the model.

The steps for the deployment of deep learning models on Android and iOS smartphones are described in more details in reference[2].

5. REAL-TIME PROCESSING RATES

iPhone 7 and Google Pixel 2 Android phones were used to run the trained deep learning models developed in this work. Screenshots of the implemented app on iPhone 7 are shown in Figure 7 for the four stages of DR and the screenshots of the implemented app on Google Pixel 2 Android phone are shown in Figure 8 for the four stages of DR. If no DR gets detected (stage 0), the app screen displays "No DR detected". When DR gets detected, it displays one of the four DR stages with the highest probability: "Mild DR" for stage 1, "Moderate DR" for stage 2, "Severe DR" for stage 3, or "Proliferate DR" for stage 4.

On iPhone 7, the app size is 132 MB. The app makes a decision at the rate of 3 frames per second, or one classification takes about 250ms. On Google Pixel 2, the app size is 73 MB. The app makes a decision at the rate of 4 frames per second, or one classification takes about 200ms.

Figure 7. Sample screenshots of the app running in real-time on iPhone 7

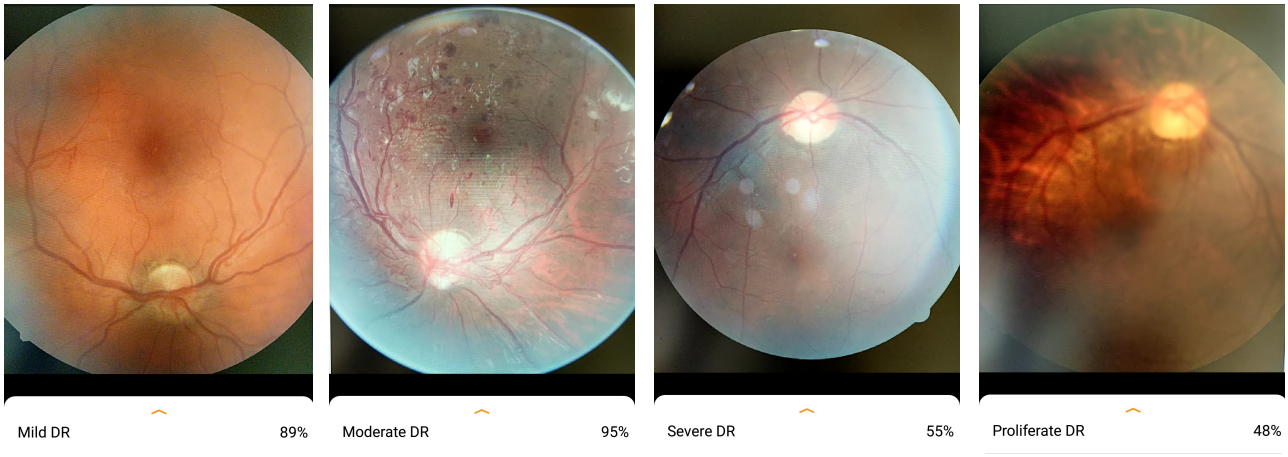

Figure 8. Sample screenshots of the app running in real-time on Google Pixel 2

6. CONCLUSION

Towards achieving a cost-effective and widely accessible first-pass diabetic retinopathy eye examination, a smartphone app has been developed in this work by capturing and processing its camera images in real-time. The app uses two trained deep neural networks, one to classify the absence of diabetic retinopathy and three stages, and the other to further classify the last stage into two stages for a total of five stages with zero stage indicating the absence of diabetic retinopathy. It has been shown how these trained networks are turned into a smartphone app running in real-time on both Android and iOS smartphones. The real-time app developed in this work enables conducting first-pass eye exams in clinics or remote areas whose access to fundus cameras and ophthalmologists are limited as well as in many third-world countries with no or limited access to fundus cameras and ophthalmologists.

7. ACKNOWLEDGEMENTS

The authors would like to thank the University Paris Est Marne-La-Vallée (UPEM) for supporting Dr. Kehtarnavaz's visit to the university to conduct this joint collaboration between Embedded Machine Learning Laboratory at the University of Texas at Dallas and Laboratoire d'Informatique Gaspard Monge at the University Gustave Eiffel, France.

8. REFERENCES

[1] H. Wei, A. Sehgal, and N. Kehtarnavaz, "A deep learning-based smartphone app for real-time detection of retinal abnormalities in fundus images," Proc. SPIE 11401 (2019).
[2] Sehgal, A., and Kehtarnavaz, N., "Guidelines and benchmarks for deployment of deep learning models on smartphones as real-time apps," Machine Learning and Knowledge Extraction, 1, 450-465 (2019).
[3] International Diabetes Federation. IDF Diabetes Atlas 159 (2019)
[4] Sánchez, C., García, M., Mayo, A., López, M., and Hornero, R., "Retinal image analysis based on mixture models to detect hard exudates," Medical Image Analysis 32, 650-658 (2009).
[5] Giancardo, L., Meriaudeau, F., Karnowski, T., Li, Y., Garg, S., Tobin Jr., K., and Chaum, E., "Exudate-based diabetic macular edema detection in fundus images using publicly available datasets," Medical Image Analysis 16, 216-226 (2012).
[6] Dawar, N., and Kehtarnavaz, N., "Action detection and recognition in continuous action streams by deep learningbased sensing fusion," IEEE Sensors Journal 18, 9660-9668 (2018).
[7] Wei, H., Laszewski, M., and Kehtarnavaz, N., "Deep learning-based person detection and classification for far field video surveillance," Proceedings of IEEE Dallas Circuits and Systems Conference, Richardson, TX (2018).
[8] Xu, K., Feng, D., and Mi, H., "Deep convolutional neural network-based early automated detection of diabetic retinopathy using fundus image," Molecules, 12 2054 (2017).
[9] Lam, C., Yu, C., Huang, L., and Rubin, D., "Retinal lesion detection with deep learning using image patches," Investigative Ophthalmology & Visual Science 59, 590-596 (2018).
[10] Gargeya, Rishab, and Theodore Leng. "Automated identification of diabetic retinopathy using deep learning." Ophthalmology 124(7), 962-969 (2017).
[11] Maninis, K., Pont-Tuset, J., Arbeláez, P., and GooL, L., "Deep retinal image understanding," Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention, Athens, Greece (2016).
[12] Panoptic Opthalmoscope, https://www.welchallyn.com/en/products/categories/physical-exam/eye-exam/ophthalmoscopes--wide-view-direct/panoptic_ophthalmoscope.html (2020)
[13] D-EYE, https://www.d-eyecare.com/ (2020).
[14] Elloumi, Y., Akil, M., and Kehtarnavaz, N., "A mobile computer aided system for optic nerve head detection," Computer Methods and Programs in Biomedicine 162, 139-148 (2018).
[15] Elloumi, Y., Akil, M., and Kehtarnavaz, N., "A computationally efficient retina detection and enhancement image processing pipeline for smartphone-captured fundus images," Journal of Multimedia Information System 5, 79-82 (2018).
[16] EyePACS, http://www.eyepacs.com (2018).
[17] APTOS 2019 blindness detection dataset, https://www.kaggle.com/c/aptos2019-blindness-detection/data (2020).

[18] Akil, M., Elloumi, Y., Kachouri, R., “Detection of retinal abnormalities in fundus image using CNN deep learning networks”, State of the Art in Neural Networks, vol. 1, in press (2020).
[19] Akil, M., Elloumi, Y., “Detection of retinal abnormalities using smartphone-captured fundus images: a survey,” Proc. SPIE 10996 Real-Time Image Processing and Deep Learning, Baltimore (2019).
[20] Elloumi, Y., Akil, M., Boudegga, H., “Ocular diseases diagnosis in fundus images using a deep learning: approaches, tools and performance evaluation,” Proc. SPIE 10996 Real-Time Image Processing and Deep Learning, Baltimore (2019).
[21] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., "Rethinking the inception architecture for computer vision," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV (2016).
[22] Krizhevsky, A., Sutskever, I., and Hinton, G., "Imagenet classification with deep convolutional neural networks," Communications of the ACM 60, 85-90 (2012).
[23] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Li, F., "ImageNet: A large-scale hierarchical image database," Proceedings of IEEE Computer Vision and Pattern Recognition, Miami Beach, FL (2009).
[24] Tensorflow, https://www.tensorflow.org/ and https://www.tensorflow.org/lite/tfmobile (2019).
[25] Keras, https://keras.io/ (2019).
[26] CoreML, https://developer.apple.com/documentation/coreml (2019).
[27] Android Studio, https://developer.android.com/studio/ (2019).
[28] Xcode, https://developer.apple.com/xcode/ (2019).