Diabetic Retinopathy Detection Using Convolutional Neural Networks for Mobile Use

Meltem Esengönül[1,2], Anselmo Cardoso de Paiva[3], João Rodrigues[4], and António Cunha[1,5(✉)]

[1] Escola de Ciências e Tecnologia, University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal
acunha@utad.pt
[2] University of Applied Sciences Technikum Wien, Vienna, Austria
[3] Applied Computing Group NCA-UFMA Federal University of Maranhão, Sao Luis, MA 65072-561, Brazil
[4] LARSyS and ISE, Universidade de Algarve, 8005-139 Faro, Portugal
[5] Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência, 4200-465 Porto, Portugal

Abstract. Diabetes has significant effects on the human body, one of which is the increase in the blood pressure and when not diagnosed early, can cause severe vision complications and even lead to blindness. Early screening is the key to overcoming such issues which can have a significant impact on rural areas and overcrowded regions. Mobile systems can help bring the technology to those in need. Transfer learning based Deep Learning algorithms combined with mobile retinal imaging systems can significantly reduce the screening time and lower the burden on healthcare workers. In this paper, several efficiency factors of Diabetic Retinopathy detection systems based on Convolutional Neural Networks are tested and evaluated for mobile applications. Two main techniques are used to measure the efficiency of DL based DR detection systems. The first method evaluates the effect of dataset change, where the base architecture of the DL model remains the same. The second method measures the effect of base architecture variation, where the dataset remains unchanged. The results suggest that the inclusivity of the datasets, and the dataset size significantly impact the DR detection accuracy and sensitivity. Amongst the five chosen lightweight architectures, EfficientNet-based DR detection algorithms outperformed the other transfer learning models along with APTOS Blindness Detection dataset.

Keywords: Diabetic Retinopathy · Deep Learning · Transfer Learning · Convolutional Neural Networks · Mobile Use

1 Introduction

Diabetes mellitus is a set of metabolic illnesses distinguished by hyperglycemia caused by abnormalities in insulin production, insulin action, or perhaps both [1]. Diabetes

A. Cunha et al. (Eds.): MobiHealth 2022, LNICST 484, pp. 14–23, 2023.
https://doi.org/10.1007/978-3-031-32029-3_2

causes long-term significant harm, malfunction, and breakdown of multiple organ systems, including the eyes, kidneys, nerves, heart, and blood vessels [1]. One of the most common types of eye complications associated with Diabetes Mellitus is Diabetic Retinopathy (DR), and it is considered a major cause of visual loss worldwide [2]. Early detection/screening is the key to preventing blindness from DR, since this illness in its early stages is asymptomatic [3]. With the growth of Deep Learning (DL) methods such as Convolutional Neural Networks (CNNs), various stages of DR can be detected automatically, assisting in detecting more cases in mass screenings. There are several challenges in regions with restricted eye health services access, like underdeveloped and overcrowded regions. A mobile-based automatic DR detection system can be used to overcome these challenges. In this paper, several different CNN-based models are investigated, and their efficiency is compared for mobile DR detection systems.

2 Related Work

There are several state-of-the-art techniques to detect DR for mobile use. In 2018, a CNN method based on a pre-trained MobileNet model was trained on Kaggle DR Detection Dataset with 16,798 fundus images and tested to detect the presence of DR, which did not need an internet connection [4–6]. By using this system, authors were able to achieve approx. 0.73 accuracy score, and their results suggest that their model compared to earlier Inception and VGGNet counterparts was 4 times lighter [6–8]. In the same year, as another essential indicator of DR, researchers applied Optic Nerve Head (ONH) detection method based on Radon Transform for the Android smartphone application with the use of images taken with D-EYE lenses attached to a smartphone [9, 10]. This method was able to reach approx. 0.96 and 1.00 accuracy scores for STARE and DRIVE datasets, respectively [10–12]. An automated mobile DR diagnosis system was introduced in 2019 based on the pre-trained Inception CNN model and a decision tree-based binary ensemble classifier [7, 13]. The method proposed was trained on Kaggle DR Detection Dataset with 1000 images and was able to classify DR into 5 different classes with an accuracy score of 0.99 [5, 13]. In another study, cloud computing and big data analytics are employed aside from mobile DR detection with AI [14]. The authors used a Deep Convolutional Neural Network (DCNN) for training, followed by a Support Vector Machine (SVM) for classification. Also, they created their mobile application called "Deep Retina" to use with a handheld ophthalmoscope [14]. Results depict that, for binary classification the system's accuracy was higher than the multi-class classification technique with approx. 0.91 and 0.86, respectively [14]. In a 2020 study, authors explored the fine-tuning effects after the CNN architecture with MobileNetV2 for DR detection [15, 16]. Their outcomes resulted in an increase of 0.21 in training and 0.31 in validation accuracy (from 0.7 to 0.91 and from 0.5 to 0.81 correspondingly).]. Another effective mobile application implementation was using the EfficientNet-B5 based CNN model for DR detection and classification into 5 labels, where the authors achieved a training accuracy value of approx. 0.94 and Quadratic Weighted Kappa value of 0.93 [17, 18]. In 2021, researchers introduced a new DL-based software called VeriSee™ for image analysis and validated its use for DR severity classification [19, 20]. The VeriSee™ software achieved higher sensitivity and specificity than the ophthalmologists [19, 20].

These results indicated that it could be viable for clinical use in ophthalmology. Around this time, another team tested the use of Phelcom Eyer Technology, a smartphone-based DR detection system, integrated with CNN based software algorithm PhelcomNet was also able to achieve high sensitivity and specificity results and an Area Under Curve (AUC) score of 0.89 [21, 22]. Their research gave a real-life example of portable retinal imaging systems combined with DL algorithms for patients enrolled in the Itabuna Diabetes Campaign in Brazil [22]. In a recent study, a Diabetic Retinopathy Graph Neural Network, in other words, the DRG-NET model was introduced for DR grading and trained with public datasets such as APTOS 2019 Blindness Detection and MESSIDOR datasets [23–25]. In this network, the input images are given as nodes, which are 3D graphs, and Scale Invariant Feature Transform methods feature extraction is performed [25]. Using this technique, the authors reached accuracy scores of 0.9954 and 0.9984, respectively, according to the datasets [25]. Another approach in 2022 was to test the efficiency of the NasnetMobile network with a multi-layer perceptron (MLP) classifier on 440 fundus images taken with smartphone-based systems [26, 27]. With this light algorithm, on low-quality images (compared to table-top fundus camera images), they achieved a high accuracy value of around 0.96 in a fast mobile execution time of less than a minute [27].

3 Methods

In this paper, two different methods were applied to test the smartphone-based applications of DL methods for DR detection. The first three public DR detection datasets were pre-processed, and data augmentation was applied and later trained and tested with MobileNet pre-trained model [4]. The second method used the same preprocessing and data augmentation techniques; however, only one chosen dataset was trained with five different CNN-based transfer learning architectures and tested for mobile use.

3.1 Datasets

The datasets used in the paper are: APTOS 2019 Blindness Detection [23], IDRID [28], and MESSIDOR [24] datasets. From the APTOS dataset, a total of 3662 fundus images were used. Whereas, we used all images from the IDRID and MESSIDOR datasets, 516 and 1200 images respectively. The APTOS and IDRID datasets included similar image classes, which had five distinct labels such as: 0 - No DR, 1 - Mild DR, 2 - Moderate DR, 3 - Severe DR, and 4 - Proliferative DR [23, 28].

As per the MESSIDOR dataset, the classification labels did not include the proliferative DR (PDR), thus were labeled from 0–3 with the following distinction [24]:

- 0 - When both microaneurysm AND hemorrhage numbers are zero
- 1 - When microaneurysm numbers are between 0- 5 (including 5) AND hemorrhage numbers are zero
- 2 - When microaneurysm numbers are between 5–15 OR hemorrhage numbers are between 0–5 AND neovascularization is not found

- 3 - When microaneurysm numbers are equal or more than 15 OR hemorrhage numbers are equal or more than 5 OR neovascularization is found

3.2 Pre-processing and Data Augmentation

For pre-processing of several images and testing, first both training and validation images were resized to 224x224 pixels in order to fit into the MobileNet model, as the original dataset contained pictures with varying pixel sizes, later the images were changed to grayscale, and cropped. CLAHE, which stands for contrast limited adaptive histogram equalization, was also applied after cropping. As the dataset sizes are small on all three chosen public datasets, data augmentation was employed to avoid overfitting, which utilized a zooming factor of 0.035 and rotation range of 0.025. In order to keep a similar appearance to fundus images they were not flipped. For training and testing all datasets were split with 80–20 split percentage.

3.3 CNN Architectures

Transfer learning is the use of prior knowledge from another task to apply it to a newer problem. Here five different pre-trained models were employed as part of transfer learning to increase the accuracy of DR detection methods. These pre-trained models were MobileNet [4], MobileNetV2 [15], EfficientNetB0 [17], EfficientNetV2B0 [29], and NasNetMobile [26]. The reason behind this selection is that according to Keras applications, these models have the least model size, meaning they are lighter weight compared to other pre-trained models, which was taken as an indicator of faster mobile applications.

All the models were tested three times. First, without fine-tuning and a smaller added dense layer structure where on top of the pre-trained MobileNet model several sequential dense layers were added. The first two layers are *global_average_pooling2d, flatten*, Then two *dense* layers are added with *(None, 32)* output shape and *ReLU* activation function, followed by a *dropout* layer, and finally a *dense* layer with *(None, 5)* output shape with *sigmoid* activation function since there were 5 different DR classes. Except for the MESSIDOR dataset, where there are four different DR classes, thus the last layer had *(None, 4)* output shape with a sigmoid activation function. As an optimizer, *adam* was chosen and for the loss function several were employed such as *mse, categorical_crossentropy,* and *binary_crossentropy*. This model was denoted as Model-1.

The second model, Model-2, has a slightly bigger architecture for dense. In this model, similar to Model-1, sequential layers were added on top of the MobileNet. These layers start with global_average_pooling2d, flatten and then continues with dense layers starting from (None, 256) output shape and ReLU activation function, followed by a dropout layer, continually adding dense layers that has the output shape halved in each layer until to the last dense layer with (None, 5) output shape with sigmoid, as before.

The third model, named Model-FT, first applied fine-tuning by setting the trainable layer true and after the same model structure as Model-1 was used, which had MobileNet as a pre-trained model.

3.4 Evaluation Metrics

This paper utilizes AUC score, and Receiver Operator Characteristic (ROC) values, Training time in addition to the basic DL evaluation metrics like Accuracy, Loss, Sensitivity, and Specificity.

The ratio of true predictions to total outputs is known as accuracy, and it may be expressed as [30]:

$$\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

Sensitivity is also known as Recall or True Positive Rate, and it has the following formula [30]:

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

The True Negative Rate is also known as Specificity, and it contains the following formula [30]:

$$\frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{3}$$

Loss value comes from the loss function, or in other words, the cost function, which boils down all of the positive and negative characteristics of a potentially complicated system to a scalar value, between 0–1, that can be used to rank and compare viable solutions [31].

ROC plots are 2D graphs with the true positive (TP) rate on the Y axis and the false positive (FP) rate on the X [32]. The relative tradeoffs between gains (TP) and costs (FP) are depicted on a ROC plot [32].

AUC Score, or area under a ROC curve, is a numerical value that quantifies a binary classifier's current effectiveness and ranges between 0.5–1.0 [33].

4 Results

As explained above, there are two main methods used to detect the effectiveness of DR detection algorithms. First, the dataset used to train the algorithm influences the evaluation metrics; second, the pre-trained model is based on architecture, and that has a different impact on the results. Lastly, the added layers on the model or whether the model include fine-tuning or not can change the effectiveness of the DL method used.

In Table 1 below, it is possible to see that the highest mean values for accuracy, sensitivity, specificity, and AUC were with the use of the APTOS dataset. In fact, in terms of sensitivity and AUC values, there is a notable drop for the MESSIDOR dataset. This drop might be due to the difference in the number of training images and the representation of different classes. The APTOS dataset included more than triple the number of images of the MESSIDOR dataset and more than 7 times higher than the IDRID dataset. Thus, we can see that the average training time is also higher compared

Table 1. Comparison of mean evaluation metrics by dataset types

	APTOS	IDRID	MESSIDOR
Accuracy	0.880	0.803	0.750
Loss	0.084	0.147	0.174
Sensitivity	0.966	0.909	0.511
Specificity	0.994	0.849	0.815
AUC	0.894	0.700	0.649
Time (min)	62.633	8.056	14.250

to other dataset types. The loss value of APTOS is considerably lower than IDRID and MESSIDOR models, which means that the average loss over time is smaller.

Table 2 depicts that, in general, EfficientNets perform higher than MobileNets, followed by NasNetMobile for the APTOS dataset. However, on the contrary, the mean training time is much lower for MobileNets compared to EfficientNets. In terms of sensitivity and AUC scores, there was a significant decrease in NasNetMobile models. The explanation is that fine-tuning did not perform well with this network, thus decreasing the average values.

Table 2. Comparison of mean evaluation metrics by architecture types.

	EfficientNetB0	EfficientNetV2B0	MobileNet	MobileNetV2	NasNetMobile
Acc.	0.903	0.902	0.880	0.850	0.841
Loss	0.072	0.074	0.084	0.124	0.126
Sens.	0.987	0.979	0.966	0.829	0.668
Spec.	0.997	0.995	0.994	0.958	0.946
AUC	0.933	0.929	0.894	0.824	0.753
T (min)	96.589	70.228	62.633	67.745	123.378

Acc. is short for Accuracy, Sens. is short for Sensitivity, Spec. is short for Specificity, AUC stands for Area Under Curve, and T (min) is to express the training time given in minutes.

Effects of different model testing are given in Table 3, where models with the Model-1 structure, which had a smaller structure on top of MobileNet and did not include fine-tuning, overall achieved the highest sensitivity values. However, there was generally no significant difference in terms of accuracy, loss, and AUC scores. As per the specificity, Model-2, where a larger model was used on top of MobileNet and did not include fine-tuning, showed the highest average value with being slightly more than Model-1. As well as, mean training time was the least with the Model-2. In this metric, the mean values are highly variable depending on the dataset size as well, however in terms of models, fine-tuning significantly increases the training time, more than triple that of Model-2, which shows the fastest method.

Table 3. Comparison of mean evaluation metrics by model types.

	Model-1	Model-2	Model-FT
Accuracy	0.811	0.810	0.811
Loss	0.135	0.136	0.134
Sensitivity	0.853	0.790	0.744
Specificity	0.887	0.892	0.879
AUC	0.748	0.747	0.748
Time (min)	25.244	13.000	46.695

Overall, the highest ROC values were achieved with the APTOS dataset. For the EfficientNet-based models, the Model-1 structure achieved the highest ROC values shown in Fig. 1, given below.

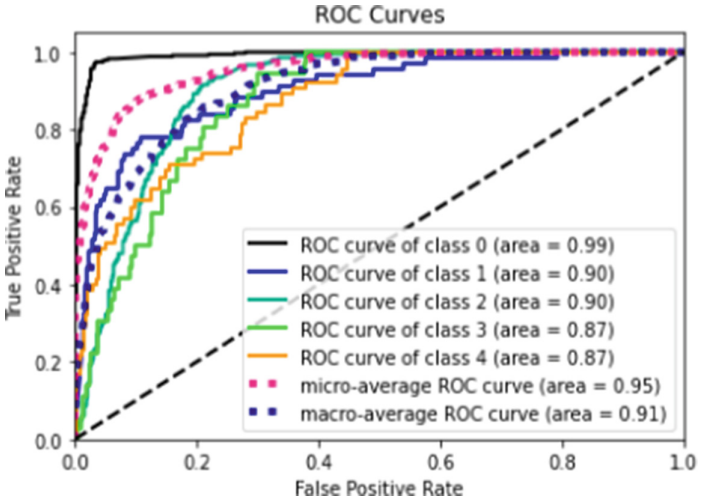

Fig. 1. ROC values achieved with APTOS 2019 Blindness Detection Dataset with EfficientNetB0 base, Model-1.

The second best results were achieved with MobileNet based architecture with the APTOS dataset and fine-tuning model (Model-FT), which can be seen in Fig. 2. Compared to the MobileNet, EfficientNetB0 was able to identify all the classes effectively, whereas the MobileNet did not perform at a similar level for Class 1, Class 3 and Class 4. In comparison, EfficientNetV2B0, a lighter version of EffficientNets, did not perform higher than EfficientNetB0.

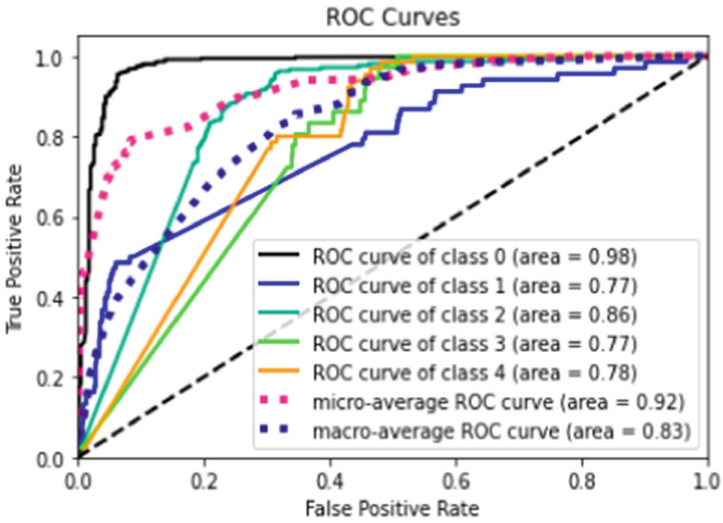

Fig. 2. ROC values achieved with APTOS 2019 Blindness Detection Dataset with MobileNet, Model-FT.

5 Conclusion

Overall, testing results show that EfficientNet-based DR detection algorithms perform better in terms of DR severity detection along with APTOS 2019 Blindness Detection dataset, even though this dataset was not distributed equally. However, they significantly take more time to train compared to MobileNets. For comparison of various models, fine-tuning might contribute to higher outcomes although significantly increasing the training time.

Acknowledgment. This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020. Furthermore, this paper is a part of the Master's thesis titled "Deep Learning Methods for Diabetic Eye Disease Screening and Smartphone based Applications" by M.E.

References

1. American Diabetes Association: Diagnosis and classification of diabetes mellitus. Diabetes Care **27**(Suppl_1), s5–s10 (2004). https://doi.org/10.2337/diacare.27.2007.S5
2. The progress in understanding and treatment of diabetic retinopathy. Prog. Retin. Eye Res. **51**, 156–186 (2016). https://doi.org/10.1016/j.preteyeres.2015.08.001
3. Detection and classification of retinal lesions for grading of diabetic retinopathy. Comput. Biol. Med. **45**, 161–171 (2014). https://doi.org/10.1016/j.compbiomed.2013.11.014
4. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications (2017). https://doi.org/10.48550/arXiv.1704.04861
5. Diabetic Retinopathy Detection. https://kaggle.com/competitions/diabetic-retinopathy-detection. Accessed 10 May 2022

6. Mobile assisted diabetic retinopathy detection using deep neural network. https://ieeexplore.ieee.org/document/8400760. Accessed 10 May 2022
7. Szegedy, C., et al.: Going deeper with convolutions (2014). https://doi.org/10.48550/arXiv.1409.4842
8. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). http://arxiv.org/abs/1409.1556. Accessed 10 May 2022
9. Website. https://www.d-eyecare.com/en_US/product
10. A mobile computer aided system for optic nerve head detection. Comput. Methods Programs Biomed. **162**, 139–148 (2018). https://doi.org/10.1016/j.cmpb.2018.05.004
11. The STARE Project. https://cecas.clemson.edu/~ahoover/stare/. Accessed 10 May 2022
12. DRIVE - Grand Challenge, grand-challenge.org. https://drive.grand-challenge.org/. Accessed 10 May 2022
13. Automated Smartphone Based System for Diagnosis of Diabetic Retinopathy. https://ieeexplore.ieee.org/document/8974492. Accessed 10 May 2022
14. Li, Y.-H., Yeh, N.-N., Chen, S.-J., Chung, Y.-C.: Computer-assisted diagnosis for diabetic retinopathy based on fundus images using deep convolutional neural network. Mob. Inf. Syst. **2019** (2019). https://doi.org/10.1155/2019/6142839
15. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: MobileNetV2: inverted residuals and linear bottlenecks (2018). https://doi.org/10.48550/arXiv.1801.04381
16. Transfer Learning with Fine-Tuned MobileNetV2 for Diabetic Retinopathy. https://ieeexplore.ieee.org/abstract/document/9154014. Accessed 10 May 2022
17. Tan, M., Le, Q.V.: EfficientNet: rethinking model scaling for convolutional neural networks (2019). https://doi.org/10.48550/arXiv.1905.11946
18. CNN Based Detection of the Severity of Diabetic Retinopathy from the Fundus Photography using EfficientNet-B5. https://ieeexplore.ieee.org/document/9284944. Accessed 10 May 2022
19. VeriSee DR. Acer Medical (2021). https://www.acer-medical.com/solutions/verisee-dr/. Accessed 10 May 2022
20. Application of deep learning image assessment software VeriSee^{TM} for diabetic retinopathy screening. J. Formos. Med. Assoc. **120**(1), 165–171 (2021). https://doi.org/10.1016/j.jfma.2020.03.024
21. PHELCOM Technologies. PHELCOM Technologies (2019). https://phelcom.com/en/. Accessed 10 May 2022
22. Malerbi, F.K., et al.: Diabetic retinopathy screening using artificial intelligence and handheld smartphone-based retinal camera. J. Diabetes Sci. Technol. (2021). https://doi.org/10.1177/1932296820985567
23. APTOS 2019 Blindness Detection. https://kaggle.com/competitions/aptos2019-blindness-detection. Accessed 10 May 2022
24. Patry, G., et al.: Messidor. ADCIS (2019). https://www.adcis.net/en/third-party/messidor/. Accessed 10 May 2022
25. Salam, A.A., Mahadevappa, M., Das, A., Nair, M.S.: DRG-NET: a graph neural network for computer-aided grading of diabetic retinopathy. J. VLSI Signal Process. Syst. Signal Image Video Technol. 1–7 (2022). https://doi.org/10.1007/s11760-022-02146-x
26. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition (2017). http://arxiv.org/abs/1707.07012. Accessed 10 May 2022
27. Elloumi, Y., Abroug, N., Bedoui, M.H.: End-to-end mobile system for diabetic retinopathy screening based on lightweight deep neural network. In: Bouadi, T., Fromont, E., Hüllermeier, E. (eds.) IDA 2022. LNCS, vol. 13205, pp. 66–77. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-01333-1_6
28. IDRiD - Grand Challenge. grand-challenge.org. https://idrid.grand-challenge.org/. Accessed 10 May 2022

29. Tan, M., Le, Q.V.: EfficientNetV2: smaller models and faster training (2021). https://doi.org/10.48550/arXiv.2104.00298
30. Tharwat, A.: Classification assessment methods. Appl. Comput. Inform. **17**(1), 168–192 (2020). https://doi.org/10.1016/j.aci.2018.08.003
31. The MIT Press. Neural Smithing. The MIT Press. https://mitpress.mit.edu/books/neural-smithing. Accessed 10 May 2022
32. An introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006). https://doi.org/10.1016/j.patrec.2005.10.010
33. Melo, F.: Area under the ROC curve. In: Dubitzky, W., Wolkenhauer, O., Cho, K.-H., Yokota, H. (eds.) Encyclopedia of Systems Biology, pp. 38–39. Springer, New York (2013). https://doi.org/10.1007/978-1-4419-9863-7_209