Deep learning for diabetic retinopathy prediction

Ciro Rodriguez-Leon[1,2][0000−0002−3920−7207], William Arevalo[3][0000−0002−7695−7154], Oresti Banos[1][0000−0001−5434−4253], and Claudia Villalonga[1][0000−0003−4253−7909]

[1] Research Center for Information and Communication Technologies, University of Granada, C/ Periodista Rafael Gómez, Nº2, E-18071, Granada, Spain.
crleon@correo.ugr.es, {oresti,cvillalonga}@ugr.es

[2] Department of Computer Science, University of Cienfuegos, Km 4, Cuatro Caminos, Cienfuegos, Cuba.
crleon@ucf.edu.cu

[3] School of Engineering and Technology, Universidad Internacional de La Rioja, Av. de la Paz, 137 26006, Logroño (La Rioja), Spain.
williamfabian.arevalo697@comunidadunir.net

Abstract. Diabetic retinopathy is a complication of diabetes mellitus. Its early diagnosis can prevent its progression and avoid the development of other major complications such as blindness. Deep learning and transfer learning appear in this context as powerful tools to aid in diagnosing this condition. The present work proposes to experiment with different models of pre-trained convolutional neural networks to determine which one fits best the problem of predicting diabetic retinopathy. The Diabetic Retinopathy Detection dataset supported by the EyePACS competition is used for evaluation. Seven pre-trained CNN models implemented in the Keras library developed in Python and, in this case, executed in the Kaggle platform, are used. Results show that no architecture performs better in all evaluation metrics. From a balanced behaviour perspective, the MobileNetV2 model stands out, with execution times almost half that of the slowest CNNs and without falling into overfitting with 20 learning epochs. InceptionResNetV2 stands out from the perspective of best performance, with a Kappa coefficient of 0.7588.

Keywords: Diabetic retinopathy · Deep learning · Transfer learning

1 Introduction

Diabetes mellitus (DM) is a group of metabolic disorders characterized by a high blood glucose level over a long time. People with DM are likely to have other major health problems. DM patients with poor control of the disease have high chances to develop complications. Especially chronic hyperglycemia can cause vascular damage and lead to developing diabetic retinopathy (DR) [21]. This complication is the ocular manifestation of DM and can cause microaneurysms, hemorrhages, exudates, venous changes, neovascularization, retinal thickening, and blindness depending on the stage of the disease. DR is the center of attention of many researchers due to its incidence. For example, in the United States 40% of diabetic patients suffer some stage of DR and it is the main cause of blindness among people of working age [17].

There are four possible stages of progress in which the DR can be diagnosed in four possible stages ranging from mild to severe: 1) mild non-proliferative retinopathy –microaneurysms are formed–; 2) moderate non-proliferative retinopathy –some blood vessels that nourish the retina are blocked–, 3) severe non-proliferative retinopathy –multiple hemorrhages in different places, venous beading, or intraretinal microvascular abnormalities–, and 4) proliferative retinopathy –formation of abnormal new vessels and/or vitreous hemorrhages– [7]. To minimize the complications of DR, healthcare professionals are looking for improvements through different strategies. One way of progress is the use of artificial intelligence techniques to support the decisions making on the diagnose of the disease. There are already multiple ways to address the problem of diagnosing DR or identify the stages of progress, artificial neural networks (ANN) are a good example, performing very well in these tasks [6]. Results in competitions in Kaggle platform range between 0.70 and 0.80 Kappa coefficient in 2015 [1]. The progress achieved in this field recently allows the rapid execution and deployment of complex algorithms such as convolutional neural networks (CNN), leaders in the detection of patterns in images [14].

With the development of the ANN and deep learning (DL), transfer learning has emerged. The general idea of this type of learning is to build on a pre-trained model generated using an extensive dataset from the same domain. The main advantage of transfer learning is that the model has not to be initialized from scratch, which saves time and provides robustness. For example, CNN's are pre-trained with reference datasets as large and representative as possible [16]. In this paper, we propose to experiment with different pre-trained CNN models to determine the most suitable according to computational cost and accuracy in the prediction of DR in healthy or any of its four stages of progression. To choose the best model, in addition to performance metrics such as accuracy, we measure its efficiency in computational resource consumption at replication and implementation. Therefore, processing times and computational resources required are also considered as evaluation metrics.

2 Related Work

Before the increasing popularity of DL techniques, researchers tried to solve the problem of detecting DR from images using traditional machine learning methods such as decision trees, support vector machine (SVM), or k-nearest neighbor. These classical techniques did not achieve the desired performance and were computationally expensive [25]. Over the years, the development and improvement of different algorithms enhanced the performance of DR detection. In 2012, a review of algorithms ranging from SVM, Bayesian optimizations, and rule-based systems to multilayer perceptrons and deep neural networks, showed the advantage of ANN over other algorithms [8]. In the last years, CNN has become one of the main techniques used in tasks like image recognition, natural language processing, and time series analysis [26]. One of the first CNN architectures used in these domains was AlexNet in 2012. Later, many more were introduced like VGG16, VGG19, ResNet, GoogLeNet, DenseNet, and Xception [16].

As a result of the above, in recent years the most common approach to DR recognition has been the DL techniques. The scientific community has focused the attention on developing and applying such techniques because it has proven to be a cost-effective tool for DR screening [18]. Some works have chosen to take only one of these architectures and try to get the best performance. In this case the most common CNN architecture used is Inception-V3 [15, 19]; Inception-v4 is also used by some studies [10, 20]; other studies used GoogLeNet, VGGNet-19, VGG16, VGGNet-16, DenseNet, and AlexNet [16]. In contrast, other authors experiment with a set of CNN and compare its results, for example AlexNet and VGG [5]; InceptionV3, ResNet18, ResNet101, VGG19, and Inception@4 (an authors´ version of InceptionV3) [9].

3 Materials and methods

3.1 Dataset

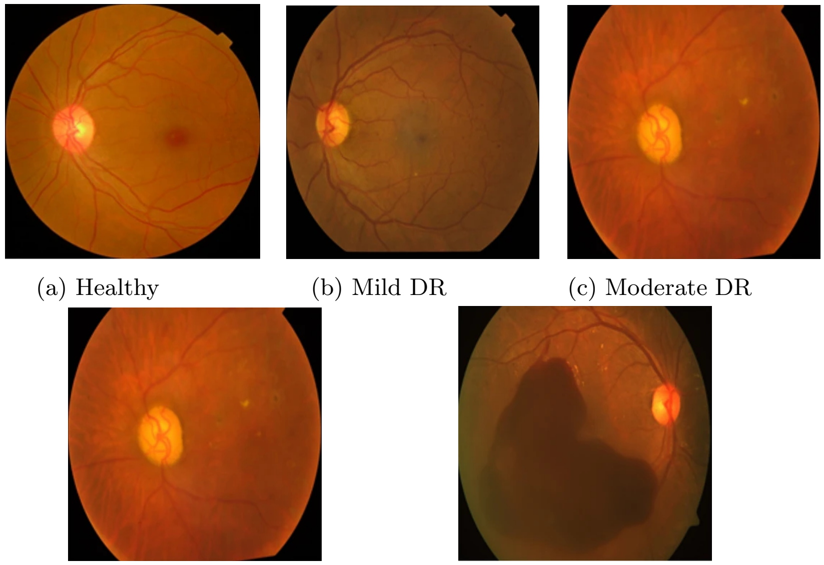
The dataset used for the experimentation is the Diabetic Retinopathy Detection by EyePACS with data augmentation [1]. It is the result of a preprocessing of the images, taken with different types and models of cameras, through a filter that allows highlighting the main features, such as microaneurysms and hemorrhages. The dataset has five classes, one representing healthy patients and the remaining four represent the degrees of retinopathy (see Fig. 1). It also has a preprocessed image magnification and therefore has a high number of observations, namely 88712 with an approximate resolution of 400 x 400 pixels. It is published by the medical institution EyePACS and preprocessed by TensorFlow.

3.2 Computing platform specs

For this study, we choose a set of CNN architectures to compare its performance and results and then get more substantial conclusions. The models used are VGG16 and its variant VGG19 [22], ResNet50V2 variant of ResNet [11], InceptionResNetV2 [23], MobileNetV2 [12], DenseNet121 variant of DenseNet [13], and EfficientNetB2 variant of EfficientNet [24].

VGG16 is used because achieved an accuracy of 91.90% in the 2014 ImageNet competition, ranking it within the top five. The main difference between VGG16 and VGG19 is that VGG19 has 19 convolutional layers instead of 16 VGG16, increasing its parameter amount from 138 357 544 to 143 667 240 because of the additional layers. The authors suggest that these additional layers make the architecture more robust and capable of learning more complex patterns.

ResNet achieved first place in the 2015 ImageNet competition with an accuracy of 94.29% in the top five. The variant ResNet50V2 used is a modified



(a) Healthy (b) Mild DR (c) Moderate DR

(d) Severe DR (e) Proliferative DR

Fig. 1: Degrees of retinopathy (DR).

version of ResNet50 that performs better than ResNet50 and ResNet101 in the ImageNet dataset. ResNet50V2 has a convolutional layer, followed by a pooling layer and 17 convolutional blocks with 3x3 filters with residual connections.

InceptionResNetV2 is a highly layered and complex network as internally it has convolutional layers with 3x3 filters, pooling layers, and dropout layers. Also, there are multiple configurations of convolutional blocks with residual connections throughout the architecture.

MobileNetV2 is designed and optimized to work in mobile and embedded computer vision applications. The standard convolutional layers are replaced by two layers that address this feature allowing for faster training. Besides, two hyperparameters are introduced to handle latency and accuracy.

DenseNet121, a variant of DenseNet, are architectures inspired by ResNet, but instead of residual connections, dense blocks are used, consisting of convolution layers placed sequentially, similar to VGG, but each layer is connected to all subsequent layers. The intended is to reduce the loss of information between layers, especially in deep layers.

Finally, EfficientNet is claimed to be 8.4x smaller and 6.1x faster in inference than the best existing convolutional network architectures. The EfficientNetB2 version, similar to MobilNet, has a novel optimization system added to the convolution blocks. All named CNN architectures are implemented in Keras, a DL API from the TensorFlow platform.

3.3 Experimental setup

Computing platform specs. Kaggle, the online platform for data science, is selected as a platform to perform the experiments. The R and Python 3 languages can be used, as well as the ANN Keras library developed in Python and widely supported by the scientific community. Besides, the platform provides a free GPU and TPU (Tensor Processing Unit) quota weekly (Table 1).

Table 1: Computing platform specs [4, 3].

Resource	**Specification**
Processor	Intel(R) Xeon(R) CPU @ 2.30GHz
# Physical cores	1
# Virtual cores per physical core	2
# Processing threads per virtual core	2
Amount of RAM	16GB
Hard disk	155GB
GPU	Tesla P100 - 13GB
# of hours of continuous execution allowed	9 hours
# of weekly working hours with GPU	39 hours

CNN's configuration. The data set used is unbalanced, since the number of examples where there is no retinopathy is much higher (65 343) than in the rest of the classes (23 359). Hence, a class dictionary is created to compute the weights passed to the model to give the same importance to the different classes regardless of the existing unbalance. The extraction of weights is done with the API provided by the Scikit-Learn library. At the same time, the images are resized to a size of 400 by 400 pixels, a common resolution in health problems in Kaggle, and besides small enough to maintain computational performance but with the quality to recognize relevant patterns. Later the pixels are normalized to improve the performance in the training of the CNN.

Upholding the stated objective of easy implementation, good performance, and minimum cost resources, as well as the utilization of the previously mentioned pre-trained CNN models, the following hyperparameter configuration, are used:

- Adam optimizer; as it allows adjustment in the middle of epochs, leading to great flexibility to improve the model performance while training.
- Learning rate: 0.0001; strongly extended across different computer vision projects on the Kaggle platform.
- Error metric: categorical_crossentropy; due to the type of single labeled multiple class problem, this metric is the ideal metric for this project.
- Performance metric: CategoricalAccuracy; as it allows finding the average number of hits regardless of the class.

– Number of epochs: 20; as it is the average where no model exceeds the execution time allowed in Kaggle.

Evaluation. To run the experimentation the data set is divided into training, validation, and test using 40%, 12%, and 48% of the data respectively. After being trained and stored at the epoch of best performance for the loss metric in the validation set, the model is evaluated with independent data. From this last evaluation, the loss and the categorical accuracy are obtained. Also, the Kappa coefficient is calculated to compare the solutions of this work with those of the competition using the same dataset on the Kaggle platform [2].

Computational complexity is measured by taking into account two features. First, it is compared the number of layers contained in each CNN architecture, whereas the higher number of layers higher the complexity. Secondly, is used the overall running time consumed by the CNNs on training, validation, and testing.

4 Results and discussion

The results of the pre-trained CNN models are shown in Table 2. Analyzing the results shown we see that, depending on the metric analyzed, the best model changes. This is evidence that no architecture performs better on all metrics at the same time for this problem. Therefore, it is necessary to verify which model best approximates a balanced behavior among all the metrics taken into account. MobileNetV2 stands out because it is a small architecture (156 layers), fast to run (17288s), almost half time that of the slowest CNNs, and at the same time performing well on the other metrics with the second-best loss (0.7897) and third-best AUC (0.9254), for example. These scores may be due to the use of two types of layers that replace the traditional convolutional layers. The network does not appear to have fallen into overfitting during training and evaluation times, even with 20 epochs, so it may still improve if more epochs are added. More details on the performance of this network are available in Fig. 2.

Table 2: CNN models' results

Model	**# of layers**	**Run time**	**Loss**	**Categorical Accuracy**	**Kappa**	**AUC**
VGG16	23	21543s	0.824	0.7169	0.7345	0.9161
VGG19	26	24526s	0.8503	0.7361	0.7188	0.9179
ResNet50V2	192	16732s	0.8076	0.7035	0.6024	0.9173
InceptionResNetV2	782	29872s	0.7919	0.7272	0.7588	0.9233
MobileNetV2	156	17288s	0.7897	0.726	0.6648	0.9254
DenseNet121	429	19460s	0.7561	0.7462	0.718	0.9347
EfficientNetB2	342	29142s	0.8455	0.7585	0.6528	0.9306

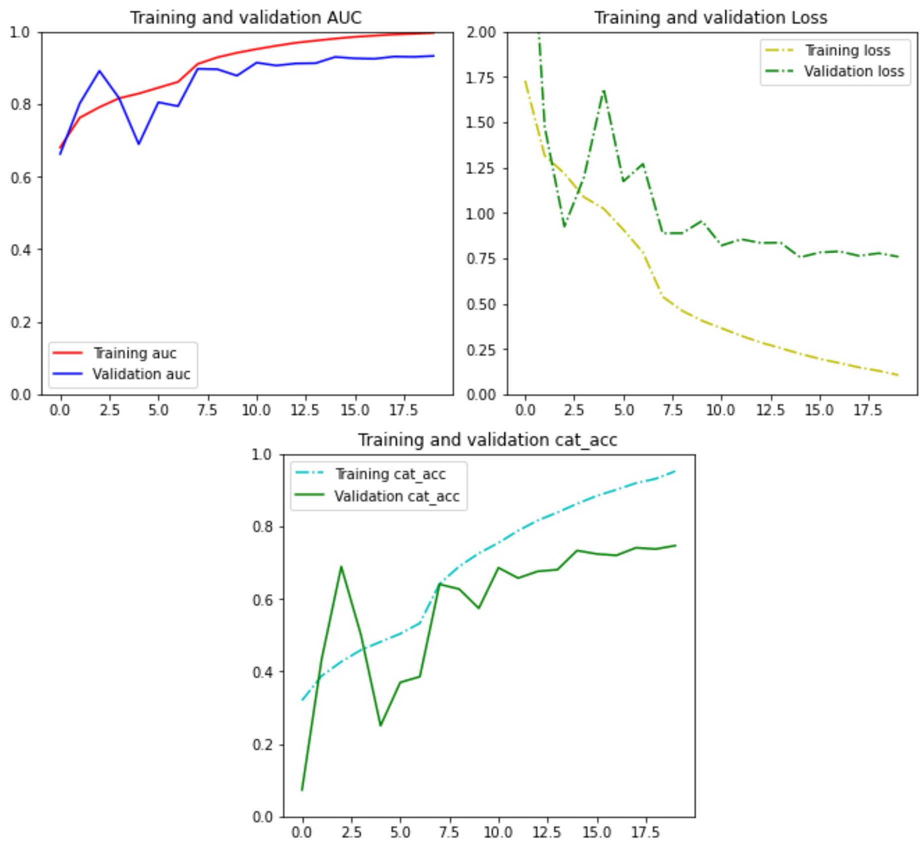


Fig. 2: Performance of MobileNetV2.

From another perspective, if the complexity and training time are put aside and only based on the resources that the Kaggle environment provides, it can be seen that the best architecture is InceptionResNetV2 because its loss is low (0.7919), its categorical accuracy is high (0.7272), even its Kappa metric (0.7588) is positioned in the 23rd position out of 610 positions for the 2015 Kaggle competition organized by EyePACS [1].This may be due to the complexity of the problem and the need to have an architecture with a large number of layers. InceptionResNetV2 mainly relies on different configurations of convolutional blocks with residual connections which makes it a robust CNN. On the other hand, according to the obtained results, there is a tendency to specialize in the predominant class even if it has regularization, such as dropout and normalization layers. More details on the performance of this network are available in Fig. 3.

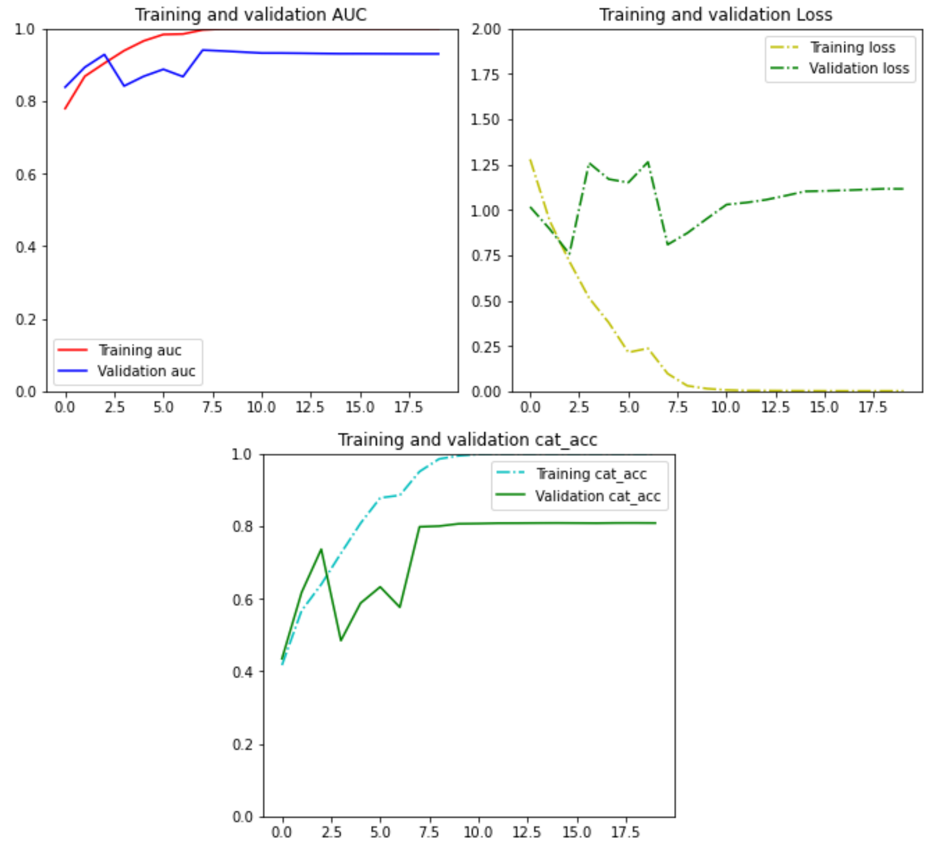


Fig. 3: Performance of InceptionResNetV2.

5 Conclusions

Diabetic retinopathy affects many people worldwide. It can cause severe vision problems and even blindness. Its identification or classification into its different types by experts is complex because the differences between the stages of evolution of the disease are very subtle. In the present work, we propose to use transfer learning to detect the 5 different classes related to DR: healthy, or one of the 4 different stages of progression of the disease. Therefore, we have compared pre-trained CNN to solve this classification problem using the Diabetic Retinopathy Detection by EyePACS from Kaggle. Specifically, VGG16, VGG19, ResNet50V2, InceptionResNetV2, MobileNetV2, DenseNet121, and EfficientNetB2 architectures have been used on the Kaggle online platform using only the resources available on it. The comparison of the results between the different models has been performed using the metrics of loss, categorical accuracy, Kappa coefficient, execution time, and structural complexity based on the number of layers.

All evaluated architectures performed above 70% categorical accuracy. Two models have stood out in the experimentation. Firstly, MobileNetV2, being a relatively small network that has achieved a good performance in all the proposed metrics, has not fallen into overfitting and has been trained in approximately half the time of the more complex networks. Secondly, InceptionResNetV2, although it has taken the longest time to train, has been the best performer according to the Kappa coefficient, placing it within the top 25 of 610 solutions for the 2015 competition on the Kaggle platform with this same dataset. The potential of pretrained CNNs to solve this and similar problems is demonstrated, it's rapid prototyping, fast training, easy implementation, and good outcomes. Results such as those of the present work are a step forward for creating reliable tools for experts in health care areas to improve the quality of life of people with DM.

References

1. Diabetic Retinopathy Detection — Kaggle, https://www.kaggle.com/c/diabetic-retinopathy-detection/overview
2. Diabetic Retinopathy Detection — Kaggle, https://www.kaggle.com/c/diabetic-retinopathy-detection/leaderboard
3. Kaggle Machine Specification — CPU/GPU/RAM/OS, https://www.kaggle.com/lukicdarkoo/kaggle-machine-specification-cpu-gpu-ram-os
4. Notebooks Documentation — Kaggle, https://www.kaggle.com/docs/notebooks
5. Abràmoff, M.D., Lavin, P.T., Birch, M., Shah, N., Folk, J.C.: Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices. npj Digit. Med. **1**(1) (2018). https://doi.org/10.1038/s41746-018-0040-6, https://doi.org/10.1038/s41746-018-0040-6
6. Abràmoff, M.D., Niemeijer, M., Suttorp-Schulten, M.S., Viergever, M.A., Russell, S.R., Van Ginneken, B.: Evaluation of a system for automatic detection of diabetic retinopathy from color fundus photographs in a large population of patients with diabetes. Diabetes Care **31**(2), 193–198 (feb 2008). https://doi.org/10.2337/dc07-1312, http://care.diabetesjournals.org
7. Engerman, R.L.: Pathogenesis of diabetic retinopathyEngerman, R. L. (1989). Pathogenesis of diabetic retinopathy. Diabetes, 38(10), 1203–1206. https://doi.org/10.2337/diab.38.10.1203. Diabetes **38**(10), 1203–1206 (1989)
8. Faust, O., Acharya U., R., Ng, E.Y., Ng, K.H., Suri, J.S.: Algorithms for the automated detection of diabetic retinopathy using digital fundus images: A review. J. Med. Syst. **36**(1), 145–157 (feb 2012). https://doi.org/10.1007/s10916-010-9454-7, https://link.springer.com/article/10.1007/s10916-010-9454-7
9. Gao, Z., Li, J., Guo, J., Chen, Y., Yi, Z., Zhong, J.: Diagnosis of Diabetic Retinopathy Using Deep Neural Networks. IEEE Access **7**, 3360–3370 (2019). https://doi.org/10.1109/ACCESS.2018.2888639
10. Gulshan, V., Rajan, R.P., Widner, K., Wu, D., Wubbels, P., Rhodes, T., Whitehouse, K., Coram, M., Corrado, G., Ramasamy, K., Raman, R., Peng, L., Webster, D.R.: Performance of a Deep-Learning Algorithm vs Manual Grading for Detecting Diabetic Retinopathy in India. JAMA Ophthalmol. **137**(9), 987–993 (sep 2019). https://doi.org/10.1001/jamaophthalmol.2019.2004, https://jamanetwork.com/

11. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). vol. 9908 LNCS, pp. 630–645. Springer Verlag (2016)
12. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv (apr 2017), http://arxiv.org/abs/1704.04861
13. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. vol. 2017-Janua, pp. 2261–2269. Institute of Electrical and Electronics Engineers Inc. (nov 2017). https://doi.org/10.1109/CVPR.2017.243
14. Indolia, S., Goswami, A.K., Mishra, S.P., Asopa, P.: Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. In: Procedia Comput. Sci. vol. 132, pp. 679–688. Elsevier B.V. (jan 2018). https://doi.org/10.1016/j.procs.2018.05.069
15. Kanagasingam, Y., Xiao, D., Vignarajan, J., Preetham, A., Tay-Kearney, M.L., Mehrotra, A.: Evaluation of Artificial Intelligence-Based Grading of Diabetic Retinopathy in Primary Care. JAMA Netw. open **1**(5), e182665 (2018). https://doi.org/10.1001/jamanetworkopen.2018.2665, https://doi.org/10.1001/jamanetworkopen.2018.2665
16. Kandel, I., Castelli, M.: Transfer Learning with Convolutional Neural Networks for Diabetic Retinopathy Image Classification. A Review. Appl. Sci. **10**(6) (2020). https://doi.org/10.3390/app10062021, https://www.mdpi.com/2076-3417/10/6/2021
17. Kanski, J.J., Bowling, B.: Clinical ophthalmology: a systematic approach. Elsevier Health Sciences (2011)
18. Lim, G., Bellemo, V., Xie, Y., Lee, X.Q., Yip, M.Y.T., Ting, D.S.W.: Different fundus imaging modalities and technical factors in AI screening for diabetic retinopathy: a review. Eye Vis. **7**(1), 1–13 (2020). https://doi.org/10.1186/s40662-020-00182-7
19. Natarajan, S., Jain, A., Krishnan, R., Rogye, A., Sivaprasad, S.: Diagnostic Accuracy of Community-Based Diabetic Retinopathy Screening with an Offline Artificial Intelligence System on a Smartphone. JAMA Ophthalmol. **137**(10), 1182–1188 (oct 2019). https://doi.org/10.1001/jamaophthalmol.2019.2923, https://jamanetwork.com/
20. Raumviboonsuk, P., Krause, J., Chotcomwongse, P., Sayres, R., Raman, R., Widner, K., Campana, B.J., Phene, S., Hemarat, K., Tadarati, M., Silpa-Archa, S., Limwattanayingyong, J., Rao, C., Kuruvilla, O., Jung, J., Tan, J., Orprayoon, S., Kangwanwongpaisan, C., Sukumalpaiboon, R., Luengchaichawang, C., Fuangkaew, J., Kongsap, P., Chualinpha, L., Saree, S., Kawinpanitan, S., Mitvongsa, K., Lawanasakol, S., Thepchatri, C., Wongpichedchai, L., Corrado, G.S., Peng, L., Webster, D.R.: Deep learning versus human graders for classifying diabetic retinopathy severity in a nationwide screening program. npj Digit. Med. **2**(1) (2019). https://doi.org/10.1038/s41746-019-0099-8, https://doi.org/10.1038/s41746-019-0099-8
21. Rodriguez-Leon, C., Villalonga, C., Munoz-Torres, M., Ruiz, J.R., Banos, O.: Mobile and Wearable Sensing for the Monitoring of Diabetes-related Parameters: Systematic Review. JMIR mHealth uHealth (oct 2020). https://doi.org/10.2196/25138
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track

Proc. International Conference on Learning Representations, ICLR (sep 2015), http://www.robots.ox.ac.uk/

23. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-ResNet and the impact of residual connections on learning. Tech. Rep. 1 (feb 2017), www.aaai.org
24. Tan, M., Le, Q.V.: EfficientNet: Rethinking model scaling for convolutional neural networks. 36th Int. Conf. Mach. Learn. ICML 2019 **2019-June**, 10691–10700 (may 2019), http://arxiv.org/abs/1905.11946
25. Williamson, T.H.: Artificial intelligence in diabetic retinopathy. Eye **35**(2), 684 (2021). https://doi.org/10.1038/s41433-020-0855-7
26. Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D.: Convolutional neural networks for time series classification. J. Syst. Eng. Electron. **28**(1), 162–169 (feb 2017). https://doi.org/10.21629/JSEE.2017.01.18