An Analysis of Pre-Trained CNNs for Diabetic Retinopathy Detection

Michael Scott

I. Introduction

Visual impairment is a problem that is becoming more prevalent as the aging population grows. Blindness and vision impairment become more of a problem the older someone becomes given the amount of age-related eye diseases that exist. The United Nations (UN) reported that there were more people aged 65 years old and over than there were children under the age of 5 years old.[1] In 2019, the World Health Organization (WHO) reported that at least 2.2 billion people worldwide were visually impaired.[2] The loss of vision has emotional, societal, and economic repercussions. Whether someone was born visually impaired or eventually became so, one's life is dramatically different compared to someone who has not experienced a visual disability. One way to measure the effects of visual impairment on someone's life is to look at their Quality of Life (QOL).[3] A study in the UK found that people with reduced vision were at an increased risk of unemployment and mental health issues that would impact their QOL.[4] In 2012, vision loss and eye disorders alone created an economic burden of 27.5 billion dollars in the United States for people under the age of 40 years old.[5] The problem with these diseases is the ability to detect them quickly and efficiently. According to the WHO, almost half of the people worldwide have visual impairments that could have been prevented or are yet to be addressed.[2] In 2015, approximately 17% of the global population in 132 countries only had access to less than 5% of the world's ophthalmologist population.[6] This creates the need to provide countries that have lower ophthalmologist populations with a way to fill this gap.

Computer-aided Detection (CADe) can help with this. CADe uses machine learning (ML) algorithms to help doctors with detecting points of interest in medical images. CADe has been used to help detect Diabetic Retinopathy (DR) at a lower financial cost and at an efficient rate.[7] DR typically occurs in middle-aged to elderly people and creates vision loss primarily because of diabetes. [8] Fundus images are used for data as they are a much cheaper alternative to optical coherence tomography (OCT) images.[7] Fundus images show us a 2D image of the back of the eye. A lot of research has been done with only using fundus images. When detecting DR, a deep convolutional neural network (DCNN) with fractional max-pooling replaced by the standard max-pooling was used to yield an accuracy of 86.17% when detecting five different levels of DR. Another group using CNNs on colored fundus images was able to achieve an accuracy of 75% and a runtime of only 188 seconds on a dataset of over 80,000 images. [9] Using classifiers such as the Gaussian Mixture model (GMM), k-nearest neighbor (KNN), and support vector machine (SVM), the researchers were able to attain a 0.904 AUC. [10]

The goal of this paper is to determine which pre-trained ImageNet model best succeeds at detecting DR with various severity levels and achieving the best quadratic weighted kappa score. Upon figuring out which of these models performs best, we will be able to have an idea of what pre-trained model might be used in the future for real-world application.

II. Methods

A. Dataset

The image dataset used is publicly available image and from the APTOS 2019 Blindess Detection Competition on Kaggle. The images vary in dimensions but are all classified according to the severity of diabetic retinopathy that the patient possesses. The levels are as follows: 0 meaning no DR, 1 meaning mild, 2 meaning moderate, 3 meaning severe, and 4 meaning proliferative DR. The entire dataset consists of 3,662 training images, 1,992 test images, and around 13,000 test images that Kaggle uses for private testing in addition to the other provided testing images. A sample of the images in the dataset can be seen below in Figure 1.

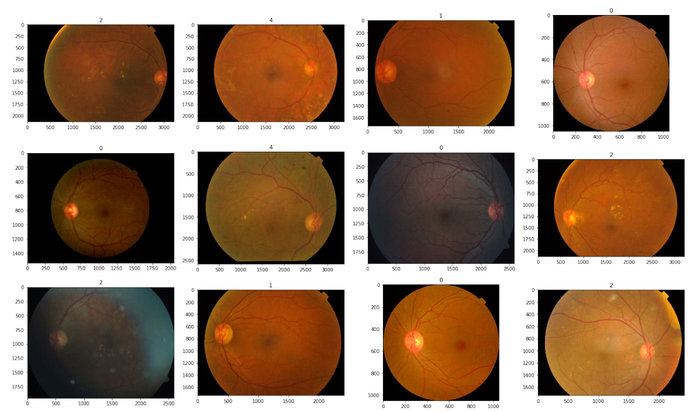

Fig. 1. Sample Images from "xhlulu"'s Kaggle Notebook

B. Evaluation Metric

The methods examined were scored by using the quadratic weighted kappa, or κ. The equation for κ is as follows:

$$\kappa = 1 - \frac{\sum_{i=1}^{k} \sum_{i=1}^{k} w_{ij} x_{ij}}{\sum_{i=1}^{k} \sum_{i=1}^{k} w_{ij} m_{ij}} \tag{1}$$

For this equation, κ is the number of codes, w_{ij} are the elements in the weight, x_{ij} are the elements in the observed, and m_{ij} are the elements in the expected matrices. The κ score is determined by the scores that a human rater assigns as well as the predicted scores.

C. Types of Models

The models used in this analysis are all convolutional neural networks (CNN) pre-trained on ImageNet. EfficientNet-B5, EfficientNet-B4, ResNet-50, DenseNet-121, and ResNet-34 were used. These pre-trained ImageNet models were compared as they have millions of trainable parameters which will aid in making very accurate predictions. The code used comes from various Kaggle users. Kaggle user, Carlo Lepelaars, created the code for the EfficientNet-B5 model used in this analysis and ran for 35 epochs. Kaggle user, Henrique Mendonça, created the code for the EfficientNet-B4 model and ran for 24 epochs. Kaggle user, "KeepLearning", created the code for the ResNet-50 model and ran for 30 epochs. Kaggle user, "xhlulu", created the code for the DenseNet-121 model and ran for 15 epochs. The code from these users was run using Kaggle's notebooks with a GPU accelerator.

D. Preprocessing

The preprocessing methods vary from each user. Carlo Lepelaars uses Ben Graham's preprocessing methods to resize the images to 128x128 and change the lighting in the photos so that more features may be easy to detect for the EfficientNet-B5 model as shown in Figure 2. Henrique Mendonça resizes the images to 224x224 and randomly rotates some of them for the EfficientNet-B4 model. Kaggle user, "KeepLearning", used image augmentation code from Github user, "aleju", to preprocess the images for the ResNet-50 model. Kaggle user, "xhlulu", resizes the images to 224x224 and randomly zooms and flips about both the vertical and horizontal axis.

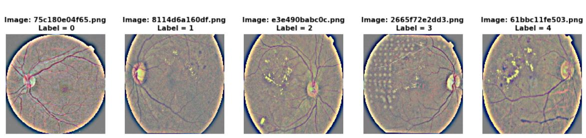

Fig. 2. Ben Graham's Preprocessing from Lepelaar's Kaggle Notebook

III. Results

EfficientNet-B5 had a runtime of 23018 seconds and attained a κ score of .9345. EfficientNet-B4 had a runtime of 17009 seconds and attained a κ score of .9287. ResNet-50 had a runtime of 4732 seconds and attained a κ score of .8869. DenseNet-121 had a runtime of 1260 seconds and attained a κ score of .9213.

IV. Discussion

EfficientNet-B5 achieved both the greatest κ score and the highest runtime. EfficientNet-B4 achieved the second greatest κ score and the second highest runtime. DenseNet-121 achieved the third greatest κ score and the lowest runtime. Lastly, ResNet-50 had the lowest κ score and the third highest runtime.

The varying scores and runtime values likely comes from the different number of trainable parameters from each model as well as the number of epochs. A comparison of these each of these models except for DenseNet-121 is displayed in Figure 3. EfficientNet-B5 most likely attains this very high runtime because of its around trainable 30 million parameters and 35 epochs. This would make sense for EfficientNet-B4 as well as it has the second highest number of parameters at around 20 million with 24 epochs. DenseNet-121 is not pictured but its low runtime likely correlates with its also low trainable parameters at around 8 million and low number of epochs. ResNet retains a very low runtime despite having more trainable parameters and more epochs than EfficientNet-B4. This likely stems from significant model architectural differences.

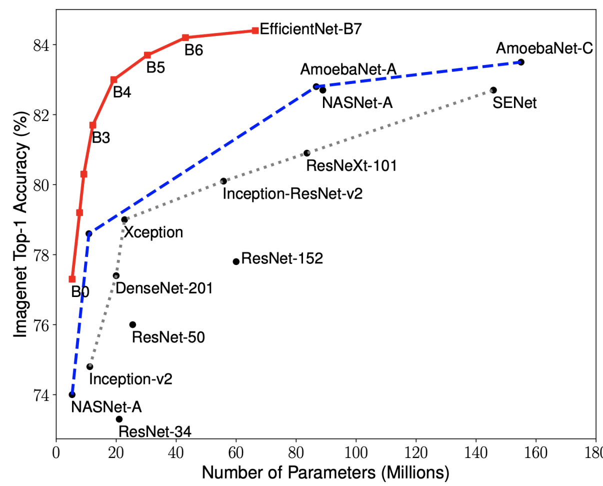

Fig. 3. Model Size vs. ImageNet Accuracy [11]

I do not find that the different preprocessing methods create a significant difference in κ scores. Both the EfficientNet-B5 model and ResNet-50 model receive images that have undergone color changes but their score rankings are only slightly greater or slightly less than the other models respectively. The EfficientNet-B4 model and DenseNet-121 model do not have any color changes yet still produce a strong score.

V. Conclusion

The results from these models are quite promising for the future of Diabetic Retinopathy detection. EfficientNet-B5 achieves the greatest score but on the downside it has quite a large runtime. On the other hand, ResNet-50 achieves a decent score with a much lower runtime than both EfficientNet models. Further research could be done to better optimize ResNet-50 in order to achieve a comparable κ score to the EfficientNets and retain the low runtime. Given that changing the color of the images did not create a significant change in κ score, perhaps simply randomly zooming future and rotating the images should suffexperiments. An implementatiice foron of this detection on a large-scale may prove to be impactful for people who have not yet developed DR. Something I could have examined more of were other submissions made by other users. It might be possible that simpler models will be able to achieve a really low runtime along with a comparable κ score. I could have also implemented a pre-trained image net model on my own. It would be interesting to see if a greater κ score could be achieved at a reasonable runtime with models such as EfficientNet-B7 or ResNet-152 which each have millions more trainable parameters than the models compared in this paper. It would also be worth evaluating these models on other metrics to see if these models achieve better scores with different evaluation metrics. A greater dataset size would also aid in creating a more accurate model.

REFERENCES

[1] Publications, U.N.: World Population Prospects 2019: Highlights. UN (2019)

[2] Organization, W.H.: World report on vision. World Health Organization (2019)

[3] of Sciences, N.A.: Making Eye Health a Population Health Imperative: Vision for Tomorrow. The National Academies Press, Washington, DC (2016)

[4] Cumberland, P.M., Rahi, J.S., for the UK Biobank Eye, Consortium, V.: Visual Function, Social Position, and Health and Life Chances: The UK Biobank Study. JAMA Ophthalmology **134** (2016) 959–966

[5] Wittenborn, J.S., Zhang, X., Feagan, C.W., Crouse, W.L., Shrestha, S., Kemper, A.R., Hoerger, T.J., Saaddine, J.B.: The economic burden of vision loss and eye disorders among the united states population younger than 40 years. Ophthalmology **120** (2013) 1728–1735

[6] Resnikoff, S., Lansingh, V.C., Washburn, L., Felch, W., Gauthier, T.M., Taylor, H.R., Eckert, K., Parke, D., Wiedemann, P.: Estimated number of ophthalmologists worldwide (international council of ophthalmology update): will we meet the needs? British Journal of Ophthalmology **104** (2020) 588–592

[7] hui Li, Y., 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) 6142839:1–6142839:14

[8] Wong, T.Y., Cheung, C.M.G., Larsen, M., Sharma, S., Simó, R.: Diabetic retinopathy. Nature Reviews Disease Primers **2** (2016)

[9] Pratt, H., Coenen, F., Broadbent, D.M., Harding, S.P., Zheng, Y.: Convolutional neural networks for diabetic retinopathy. Procedia Computer Science **90** (2016) 200–205 20th Conference on Medical Image Understanding and Analysis (MIUA 2016).

[10] Roychowdhury, S., Koozekanani, D.D., Parhi, K.K.: Dream: Diabetic retinopathy analysis using machine learning. IEEE Journal of Biomedical and Health Informatics **18** (2014) 1717–1728

[11] Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks (2020)