Detecting Diabetic Retinopathy by using Transfer Learning with the help of Progressive Resizing

Shubham Dangwal

Department of Information Technology

MGM's JNEC, Aurangabad, India

Abstract--This paper tries to study whether Progressive Resizing works and whether it helps us train for longer and what kind of accuracy can be achieved by such a single model trained using Transfer Learning [1] to detect Diabetic Retinopathy. For small datasets or datasets where it is not possible to gain increase our data it is not possible for the data to be trained for long as the amount of information is less so in order to train for longer and to improve accuracy, we can make use of Progressive Resizing.

This basically means that suppose the size of our images is 256 x 256 then we first resize it to 128 x 128 and train on it and once this training is finished, we can again resize our data to 256 x 256 and train here. As a change in size means that for a deep neural network this is completely new set of data we can train for twice as long without overfitting and also as this model was already good for detecting images at 128 x 128 by Transfer Learning idea it will also be fairly good at size 256 x 256 and after training our accuracy will improve. This also ensures that our model generalize better.

I. INTRODUCTION

Diabetic Retinopathy (DR) is a disease which leads to vision defects or blindness, this is caused by long-term diabetes. Anyone with Diabetes is at a danger of getting DR. Since, this is a progressive disease which is not noticeable during the early stages if not caught early and treated, it may cause irreparable damage to the eye and may also lead to complete blindness [2]. DR is generally checked by taking a fundus image of the eye which is then studied by the doctor to check whether it has DR or not. But there is a limited number of doctors and hence there is an urgent need for automatic detection techniques which can be used to detect DR in digital fundus images [3].

Automatic detection of DR is challenging task as there are at least five things which need to be identified if we want to detect DR. The comparison between normal and DR retina is shown in figure A, and B below, with different lighting conditions, angle, etc.

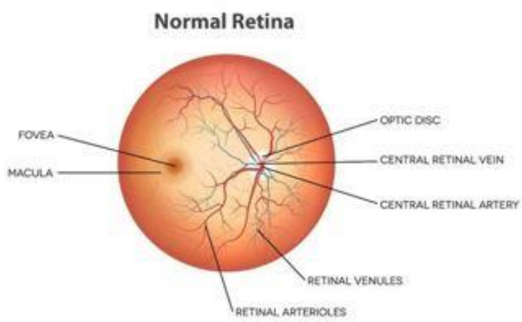

Figure A: Animation of Normal Retina [4]

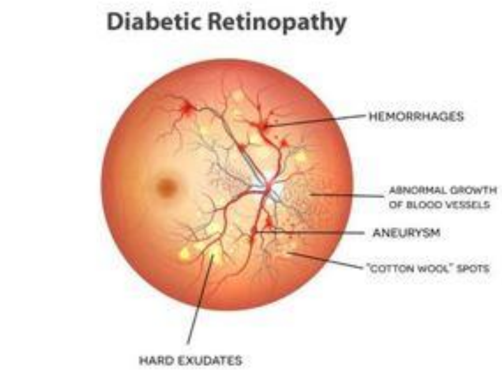

Figure B: Animation of Retina with DR [4]

As the above task is hard and complex, I have decided to use Convolution Neural Network to create the model. But training a deep enough model for this type of problem will typically take a lot of time and resources if we are training from scratch. Since, our main purpose here is to check the validity of Progressive Resizing (which was widely popularized by Jeremy Howard in his online MOOC on Deep Learning [5]) idea I had decided to use Transfer Learning as this would help me train faster and we do not need to worry about the initialization of our parameters.

II. METHODOLOGY

The dataset used in this paper comes from a Kaggle Competition [6]. This dataset consists of 3662 labelled images and 1982 unlabeled images. The unlabeled images are Kaggle's public test set and there is also a Kaggle's private test set which we do not have access to but once a model is submitted to the competition, we will get scores on the public set as well as the private set.

The data is categorized into 5 categories from 0 to 4. 0 being No DR while 4 meaning Proliferative DR. These categories are ordinal. Here, I have used the Fastai [7] package which sits on top of pyTorch for model training and inference.

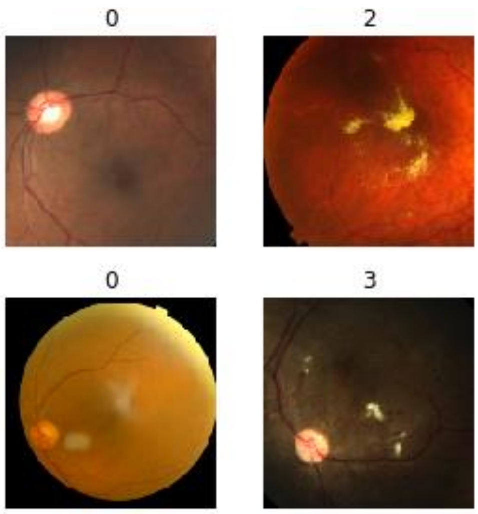

Figure C: Train images with the numbers above being the classes they belong to

For training I divided the labelled images into 80% (2930) train data and 20% (732) valid data. Both the train and valid data were then normalized using ImageNet [8] stats as the pretrained model I am going to use is Resnet [9] (specifically resnet34) which was trained on ImageNet. For data augmentation I have used the transforms like rotation, zoom, lighting, resize, etc. All these transforms are applied to the image when it is called with a probability of 0.75. So, after all these transformations the train data looks as shown in figure C.

The pretrained model I used was Resnet particularly Resnet34 (34 layers CNN). Again, a smaller model was used because our main focus is to check how progressive resizing performs. This model has been trained on ImageNet. Resnet is arguably the best architecture for CNN as shown by all the first five positions in the Stanford DAWNBench Benchmark [10] being occupied by it.

Since, here we are checking the validity of Progressive Resizing idea I created three models. The first model (Model A) does not use progressive resizing the images used here are resized to 512 x 512 (largest size) and training is performed on it. For the second model (Model B) two stage of training that is in the first stage of training the data is of size 256 x 256 and in the second stage the data is resized to 512 x 512 and then trained on it again. The third model (Model C) is trained for three stages from 128 x 128 to 256 x 256 and finally at 512 x 512.

Each stage of training for all the models consists of 10 epochs. In the first 6 epochs I only train the head of the pretrained model followed by 4 epochs where all the parameters in the model are trained. The learning rate is found by using an implementation of Leslie Smith's Paper on cyclical learning rate [11]. The fit method used for training with the mentioned learning rates uses an implementation of One Cycle Policy [12] for better convergence and faster training.

III. RESULTS

The model evaluation was done using Cohen-Kappa Score [13] which ranges from 0 to 1. For each model each stage of training consisted of 10 epochs only as going above this is possible but we would need very small learning rates and ensuring that our model does not overfit. Hence, Model A has been trained for 10 epochs while Model B for 20 epochs (10 for each stage) and Model C for 30 epochs. This at least proves that Progressive Resizing helps us train for longer at a higher learning rate without chances of it overfitting.

Table A: Result on the valid data, Kaggle public and private data

Model	**Cohen-Kappa Score**		
	Valid Data	***Public Data***	***Private Data***
Model A	0.888580	0.669540	0.864980
Model B	0.907887	0.690913	0.868331
Model C	0.908004	0.710091	0.878085

One other interesting observation was the fact that when Model B was training in the second stage it quickly achieved the highest accuracy that Model A got after 10 epochs. This is because as Model B was already good at recognizing images of sizes 256 x 256 it became easier for it to have the same validation accuracy on images of 512 x 512. More details can be found in the notebook whose link is present in the Appendix.

Table B: Training Speed of Model A and B to the same accuracy

Model	**Accuracy**	**Stage**	**Epochs**
Model A	0.888580	1	10
Model B	0.898478	2	4

This was also true with Model C it also achieved the same accuracy as Model B faster.

It is important to note that the competition winners are decided on the performance on Private Data. And although the difference in performance doesn't seem to be huge but **Model C** solution would rank **183** higher than **Model B** and **217** places higher than **Model A** solution.

IV. CONCLUSION

This paper proposes the use of Progressive Resizing to be used whenever we have small datasets and we would like to train for longer and improve our accuracy and we don't have any means of getting more data. The results above clearly show that this indeed is the case and not only for small datasets Progressive Resizing can be used whenever we would like to train for longer and we do not have any means of gaining more data. This method provides us with basically a new dataset on which no training has been done yet. The benefits of this size do diminish as we increase the stages as seen in our results above but nonetheless it improves our generalization and helps us train for longer. The future scope of this paper would be to test on various other datasets and find out a general rule of how small can you resize the images before the improvements become miniscule.

And as for the accuracy of the model it is satisfactory but not state of the art, to improve accuracy Resnet50 should be used.

V. REFERENCES

[1] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[2] D. Sidibé, I. Sadek, and F. Mériaudeau, ''Discrimination of retinal images containing brightlesions usingsparse codedfeatures and SVM,''Comput. Biol. Med., vol. 62, pp. 175–184, Jul. 2015.

[3] M. D. Abràmoff et al., ‘‘Evaluation of a system for automatic detection of diabetic retinopathy from color fundus photographs in a large population of patients with diabetes,’’ Diabetes Care, vol. 31, no. 2, pp. 62–83, 2008.

[4] https://www.shutterstock.com/image-illustration/diabetic-retinopathy-normal-eye-retina-328135913

[5] Howard, Jeremy, and others. “Practical Deep Learning for Coders” online MOOC, 2019.

[6] https://www.kaggle.com/c/aptos2019-blindness-detection/data

[7] Howard, Jeremy, and Sylvain Gugger. “Fastai: A Layered API for Deep Learning.” Information 11.2 (2020): 108. Crossref. Web.

[8] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 248-255.

[9] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[10] https://dawn.cs.stanford.edu/benchmark

[11] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, 2017, pp. 464-472, doi: 10.1109/WACV.2017.58.

[12] Smith, Leslie. (2018). A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay.

[13] McHugh, Mary L. “Interrater reliability: the kappa statistic.” Biochemia medica vol. 22,3 (2012): 276-82.

VI. APPENDIX

[1] Source code can be found at:

https://github.com/shdangwal/Detecting-Diabetic-Retinopathy-by-using-Transfer-Learning-with-the-help-of-Progressive-Resizing

[2] The Dataset can be found at:

https://www.kaggle.com/c/aptos2019-blindness-detection/data