DIABETIC RETINOPATHY DETECTION

A PREPRINT

Shengbo Wang
Electromobility
University of Stuttgart
70569, Stuttgart
st169852@stud.uni-stuttgart.de

Junmao Liao
Electromobility
University of Stuttgart
70569, Stuttgart
st165800@stud.uni-stuttgart.de

February 10, 2021

ABSTRACT

Automatic detection in fundus images can assist in early diagnosis and screening of diabetic retinopathy. An efficient approach for the binary classification (NRDR or RDR) based on fundus retinal images is presented in this paper, which ensembles a VGG-like model after fine tuning, a ResNet18 and a ResNet34 model with L2 regularizer and He-initializer. We were able to achieve an accuracy of 89.32% and F1 score 0.908 in this experiment. For the further experiments, the accuracy for a five-class classification was 59.22%, which demonstrates a considerable result with the small IDRID dataset.

1 Introduction

Diabetic retinopathy has been proven to be a complication of diabetes, and is nominated as one of the most common causes of visual impairment and blindness[1][2]. Recently, there has been an increasing interest in reliable automatic scanning systems[3]. The paper describes a generic framework to classify nonreferable (NRDR) and referable (RDR) diabetic retinopathy based on fundus images with deep learning algorithms, and is organized as follows: Section 1 contains brief introduction. An efficient input pipeline is given in Section 2. Then in Sections 3, we present different models in details for classification, and some techniques to enhance the performance. The results of these models are described and compared in Section 4 followed by the deep visualization. Some further experiments like five-class classification or regression, or experiment on EyePACS are shown in Section 5. The last section summarizes the paper.

2 Input pipeline

2.1 The IDRID dataset

2.2 Preprocessing(Shengbo Wang)

We first balance the dataset by resampling, and then redefine the labels for binary classification, the corresponding distribution of the training set is shown in Figure 2.1. After that, we create TFRecord files to load data efficiently. Finally, we crop and resize the images.

2.3 Augmentation(Junmao Liao)

To make our model robuster, we augmented our data by flipping the image horizontally/vertically; randomly rotate/shear the image; randomly crop the image and zoom the image to the original size; randomly adjust the brightness, saturation, hue and contrast of the image. Some images after cropping, resizing, and augmentation are shown in Figure 2.2 compared with the original images.

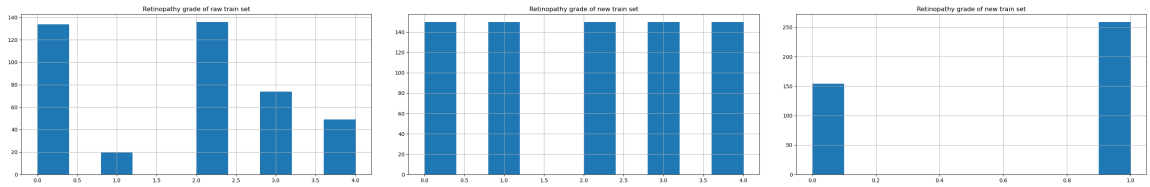


Figure 2.1: Resampling and redefining labels

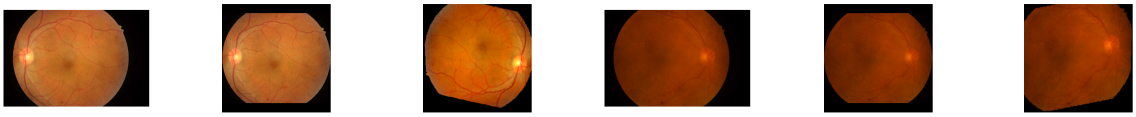


Figure 2.2: Cropping, padding and augmentation

3 Model

3.1 VGG(Junmao Liao)

We first use a VGG-like model with the following parameters: the number of base filter is 8, which is doubled for every VGG block; the number of VGG blocks is 3; the number of dense units is 32 and dropout rate is set to be 0.2.

3.2 Hyperparameter tuning(Shengbo Wang)

The corresponding performance on the test set is unsatisfied, therefore, we may need some better configurations and tune the hyperparameters by using grid search. We still select the above mentioned 4 parameters to tune, and we present part of the results in Table 3.1. A VGG model with relative more blocks and more units tends to predict more precisely.

3.3 ResNet(Shengbo Wang)

There are 5 differnet architectures of ResNet, namely ResNet18/34/50/101/152, for such a small dataset, we choose simpler ResNet18 and ResNet34 and use at the same time a L2 kernel regularizer to avoid overfitting. At the same time, for ResNet18, various initializers such as he-normal, glorot-normal, lecun-normal and so on are tested, the results are shown in Table 3.2.

3.4 Emsemble learning(Shengbo Wang)

In order to reduce generalization error, we try to combine multiple trained models, because different models will usually not make all the same errors on the test set[4]. Here we implement the combination by voting or averaging the predictions. We select 3 best models that we have already trained, namely a VGG model after fine tuning, a ResNet18 and a ResNet34 model with L2 kernel regularizer and He-initializer.

Table 3.1: Best configurations of VGG

	base filters	n blocks	dense units	dropout rate	validation accuracy
1	**16**	**5**	**64**	**0.442636894**	**93.34%**
2	16	4	32	0.259024206	91.13%
3	16	6	128	0.162307297	88.64%
4	8	3	32	0.502084817	78.02%
5	8	2	16	0.502084817	68.13%

Table 3.2: Comparison between ResNet18 models with different initializers

	he-normal	glorot-normal	lecun-normal	orginial
test accuracy	**86.41%**	78.64%	82.52%	78.64%

3.5 Transfer learning(Shengbo Wang)

By transfer learning, we can try different models more easily and faster, here we try MobileNet, InceptionV3, InceptionResNetV2 with the original weight trained on ImageNet. The top layers are replaced by a small model, which consists of a global average pooling, a dropout, and a dense layer. For DenseNet we unfreeze some of the top layers and add a more complicated model at the top.

4 Experiments

4.1 Metrics(Shengbo Wang)

We select sparse categorical accuracy, confusion matrix, ROC/AUC, sensitivity, specificity, precision and F1 score.

4.2 Details of learning(Shengbo Wang)

For training, we use a batch size of 32, for the ResNet models the batch size is reduced to 16 since it requires more GPU RAM, the optimizer we use is an Adam optimizer. Furthermore, we use sparse categorical cross-entropy or Huber loss as the loss function for classification or regression problems. The platform we use to conduct the experiments are the GPU server from ISS with one GeForce RTX 2080 Ti GPU and the Google Colab.

4.3 Results(Shengbo Wang)

The evaluation results for each model are listed in Table 4.1. The transfer models can achieve accuracy at about 80%. Among them, Densenet with some unfreeze layers and a bigger model on the top has the best performance. For VGG, we tune the hyperparameter, for ResNet we used he-normal initializer and L2 kernel regularizer, which improve the result a lot. The ensemble voting and averaging models from our best VGG and ResNet models have accidentally the same result and surpass all other models. The final accuracy is 89.32%. The AUC of ensemble averaging is 0.94.

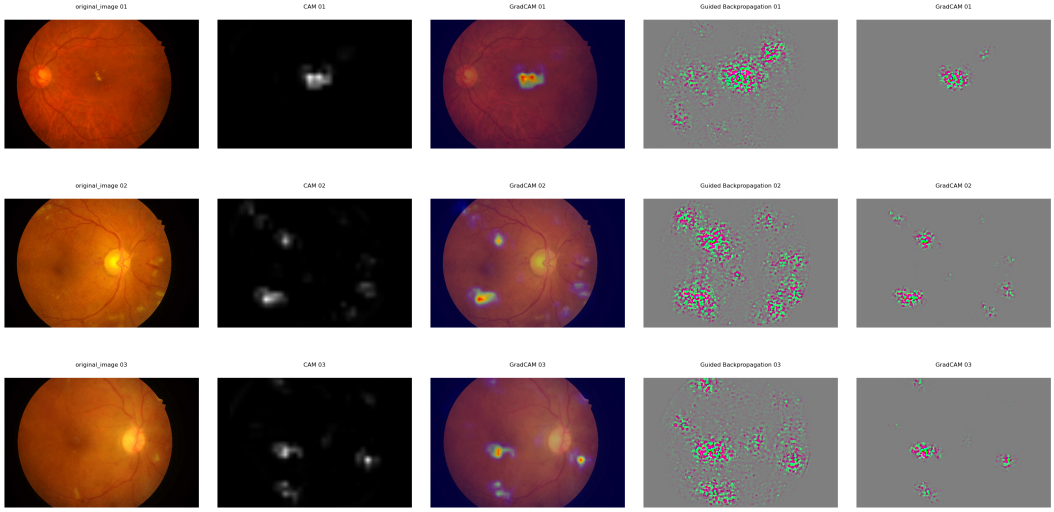
Table 4.1: Evaluation of all models

	Accuracy	Sensitivity/Recall	Specificity	Precision	F1 score
VGG	88.34%	0.875	0.897	0.933	0.903
ResNet18	86.41%	0.828	0.9231	0.946	0.883
ResNet34	86.41%	0.797	0.974	0.981	0.879
MobileNet	73.44%	0.821	0.821	0.870	0.797
InceptionV3	80.58%	0.875	0.692	0.824	0.849
InceptionResNetV2	80.58%	0.797	0.821	0.879	0.836
DenseNet	80.58%	0.797	0.821	0.879	0.836
Voting	**89.32%**	**0.844**	**0.974**	**0.982**	**0.908**
Averaging	**89.32%**	**0.844**	**0.974**	**0.982**	**0.908**

4.4 Visualization(Shengbo Wang)

The lack of decomposability into intuitive and understandable components makes CNN hard to interpret. However, interpretability matters, we must build transparent models that explain why they predict what they predict. This is where deep visualization methods come into play. Some of the most common visualization approaches are Grad-CAM, Guided Backpropagation and Guided Grad-CAM. Some typical visualization outputs are shown in Figure 4.1(on the best VGG model). We can see clearly that the typical symptoms associated with diabetic retinopathy such as microaneurysms, soft and hard exudates and hemorrhages are detected.

Figure 4.1: Deep visualization



5 Further experiments

5.1 Multi-class classification(Shengbo Wang)

To further considering the multi-class classification problem, namely try to classify all stages according to the International Clinical Diabetic Retinopathy Scale(5 labels). We omit the step of redefining the label and select the ResNet18 model which has good performance in the binary task, and change the neurons of the output layer from 2 to 5. The sparse categorical accuracy on test set is 59.22%, and corresponding confusion matrix is shown in Figure 5.1.

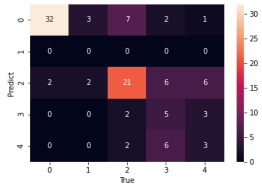


Figure 5.1: Confusion matrix

5.2 Regression(Shengbo Wang)

Sometimes for medical applications, the boundaries between these 5 stages could not be so explicit, so we try regression after the 5-class classification task. We use Huber loss instead of sparse categorical cross entropy.

6 Conclusion

Overall, we performed an efficient binary classifier(NRDR or RDR) based on fundus retinal images. We tried various transfer models from keras with a small model on the top. We combined a VGG model with optimized hyperparameters, a ResNet18 and a ResNet34 model with l2 regularizer and he-normal kernel initializer by averaging predictions and voting. The best test accuracy of this ensemble model can be 89.32%, which proves the good generalization ability. For a five-class classification problem, we could reach 59.22% test accuracy with a ResNet18 model.

References

[1] R. Klein, B.E.K. Klein and S.E. Moss. Visual impairment in diabetes. In Ophthalmology 91, pages 1-9, 1994.

[2] Kaji Y. Diabetic eye disease. In *diabetes and aging-related complications. Springer*,pages 19-29, 2018.

[3] Zhentao Gao, Jie Li, Jiexiang Guo, Yuanyuan Chen, Zhang Yi, and Jie Zhong. Diagnosis of Diabetic Retinopathy Using Deep Neural Networks. *IEEE Access, pages 3360-3370*, 2019.

[4] Goodfellow Ian, Bengio Yoshua, Courville Aaron. Deep Learning, MIT Press. *http://www.deeplearningbook.org*, 2016.