DIABETIC RETINOPATHY DETECTION BASED ON DEEP LEARNING

A PREPRINT

Wenwu Tang [*]
Institute of Signal Processing and System Theory
University of Stuttgart
Stuttgart, Universitaetstrasse
st180408@stud.uni-stuttgart.de

Xiangyuan Meng
Institute of Signal Processing and System Theory
University of Stuttgart
Stuttgart, Allmandring
st176701@stud.uni-stuttgart.de

July 24, 2023

ABSTRACT

This paper employs deep learning methods for the classification of Diabetic Retinopathy (DR). The objective of this project is to train a deep neural network to accurately categorize input fundus images. The Indian Diabetic Retinopathy Image dataset is utilized for training and testing of deep learning models. The methodology can be divided into four parts: First, Pre-processing of images and preparation of the input pipeline, including the implementation of the Ben Graham method for image enhancement and augmentation of the data. Second, Construction of deep learning models, such as ResNet-18, VGGNet, and models based on transfer learning, for disease severity prediction, Third, Training of the models using the input data. Fourth, Evaluation of the model training effects through analysis of the results. The results demonstrate the efficacy of the proposed models, with VGGNet demonstrating the best performance among all models. To further address class imbalance and improve performance, Focal Loss was introduced, followed by the application of ensemble learning. The study culminated in an accuracy of 86.4% on the small data set, and the use of Grad-CAM and dimensionality reduction techniques to visualize and analyze the models and results.

1 Introduction

Diabetic retinopathy (DR) is a medical disease in which the retina of the human eye is smashed because of damage to the tiny retinal blood vessels in the retina. The goal is to predict whether the patient has non-referable (NRDR) or referable diabetic retinopathy (RDR) based on the images. NRDR is considered mild, while RDR is considered moderate to severe. The paper is organized into several sections including data analysis, model architectures, evaluation results, interpretation of predictions, and conclusion. The input pipeline, image pre-processing techniques, and model training and testing details are also discussed.

2 Input pipeline

The input pipeline involves processing fundus images, including normalization and data augmentation, to prepare them for analysis by the deep learning model. It includes: 1) Applying Ben Graham method to process and resize images in both train and test datasets. 2) Balancing the training data set by resampling the underrepresented class. 3) Augmenting the training dataset by rotating, translating, and cropping images, resulting in 2000 images per class. 4) Serializing images and labels into the TFRecord format.

[*] Code is avaiable https://github.com/TWWinde/Diabetic_Retinopathy.

The IDRID contains 413 training images and 103 test images rated for diabetic retinopathy by clinicians. The training set is divided into validation set with 8:2 ratio, but the labels are imbalanced with 154 instances of non-proliferative diabetic retinopathy (NRDR) and 259 instances of proliferative diabetic retinopathy (RDR) as shown in Figure 1. To balance the data, the paper first oversamples the minority class and then uses focal loss.

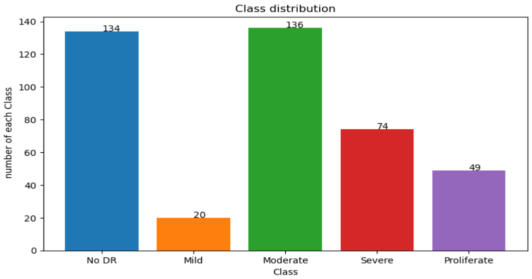


Figure 1: Class distribution

2.1 Fundus image pre-processing using Ben Graham method

The heterogeneous nature of fundus imaging equipment leads to differences in resolution, breadth, and color reproduction among images. To address this, image processing is performed in two stages: 1) Image enhancement and 2) Size normalization. The latter stage involves resizing the images to 256 × 256 pixels without distorting them.

Here, the Ben Graham method is utilized. This methodology employs a weighted combination of two images through the utilization of the OpenCV library. The method inputs two images and performs a weighted sum of the two images, potentially enhancing the visual details (such as microaneurysms, soft exudates, haemorrhages, and hard exudates) in the original image by reducing the presence of noise and other distractions. The specific image processing and cropping procedures are as follows:

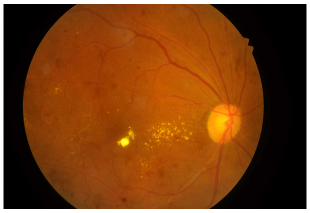


Figure 2: Input Fundus image

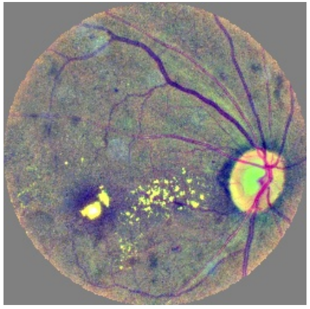


Figure 3: Processed fundus image

The described methodology pertains to the pre-processing of retinal fundus images in order to enhance the specificity of a machine learning model. The initial step involves the determination of the centroid of the functional fundus region in the image as Figure 2. This is achieved by utilizing the height of the image as the length of a square, centered at the determined geometric center. Subsequently, the Ben Graham method is applied to the image. Due to the presence of fully black regions outside the eyes, boundary effects occur following Gaussian blurring, which negatively impacts the training process. To mitigate this issue, a mask with a gray value of 0.5 is added to the image, thereby effectively removing the boundary effects and transforming the region of interest into a circular shape. Finally, the images are resized to 256 × 256, leading to a significant improvement in the quality of information, as demonstrated by the increase in specificity from 0.46 to 0.82 in the VGGNet model. An example of processed image is shown in Figure 3

3 Model Architectures

This section presents different deep learning models like basic CNN, ResNet-18 and VGGNet and pre-trained models like XceptionV3, InceptionV3 and Inception-resnet, which are modified by transfer learning method. The information about the models is as Table 1. We first train these models with the initial parameters and then perform hyperparameter optimization on the models with good performance.

Table 1: Model information

Model name	Number of params	Input size	Batch size	Kernel size	Dropout rate	Optimizer	Loss
Basic-CNN	397,986	32×256×256×3	32	3×3	0.2	Adams	CE
ResNet-18	11,559,970	32×256×256×3	32	3×3	0.2	Adams	CE
VGGNet	4,736,402	32×256×256×3	32	3×3	0.2	Adams	CE

3.1 Self-trained Models

First, a basic CNN model, which is only consist of 4 convolutional layers is built to check the correctness of input pipeline, trainer and other configurations. Data augmentaion is not applied in this model. Every convolutional layer is follow by a max-pooling layer and a batch normalization in order to get a fast convergence. In this model, underfitting appears. Due to the class imbalance, the true negativ rate is very low. When the number of convolutional layers is less than 3, model is not capable to extract enough features. Without any regularization, model tends to overfitting when the number of convolutional layers is more than 4.

Then, We try to train a ResNet-18, which has 18 layers, including 16 convolutional layers, pooling layers, and 2 fully connected layers, and is designed to address the problem of vanishing gradients in very deep networks[1]. Due to the small dataset and the large model capacity, the overfitting problem is very serious. Resnet-18 is not suitable for this small dataset.

After that, a VGGNet[2] with 5 VGG-blocks is trained. The results turned out pretty good. a deep model can be constructed by reusing the basic block (Block). Each block is consist of two convolutional layer followed by batch normalization and ReLU activations, with a number of filters that is doubled every 2 layers, starting with 16. Then they are followed by 2 dense layers, the first with dense units 64 and the last one with softmax activation.

3.2 Models based on transfer learning

Due to the limited amount of images, it's extremly hard to train a high performance models with a large amount of parameters. Transfer learning is often used in deep learning where a pre-trained model on a large dataset is fine-tuned for a specific task. The pre-trained model already have learned high-level features such as edges, textures, and shapes. When fine-tuning this model for a new image classification task, only the final dense layers for classification that correspond to the new task need to be trained, as the lower-level features will already have been learned from the pre-trained model.

XceptionV3, InceptionV3 and Inception-resnet are trained through transfer learning method and achieved a high accuracy. See Table 2.

Table 2: Information of models from transfer learning

Model name	Total parameters	Trainable parameters	Accuracy
XceptionV3	22,335,618	532,834	0.7961
InceptionV3	21,394,314	532,834	0.8155
Inception-resnet	54,738,498	401,762	0.8058

4 Train and Evaluation

We first train models with the initial parameters and find the model with best performance. After first training of VGGNet and Resnet, an ensemble learning is implemented. We combine the VGGNet, Resnet and three models from transfer learning to perform ensemble learning. The accuracy increased by 3%, but the AUC remains the same. After that, change the loss of VGGNet to focal loss, and do hyperparameter optimization for both models and loss parameters.

Table 3: Training results

Model name	Accuracy	AUC
Basic-CNN	0.7282	0.82
ResNet-18	0.7864	0.86
VGGNet-19	0.8350	0.90
Ensemble learning	0.8641	0.90

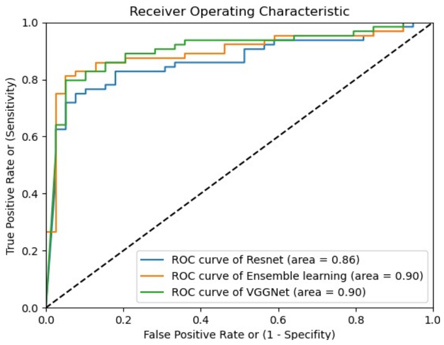


Figure 4: ROC curves

4.1 Hyperparameter optimization for VGGNet

To enhance the performance of VGGNet, we conducted hyperparameter optimization via grid search. The results are illustrated in the Figure 5. As evident from the figure, with a fixed dataset size, the objective of hyperparameter optimization is to strike an optimal balance between model capacity and dataset size. The number of base filters exhibits the greatest impact on the results, as it significantly affects the number of trainable parameters.

Table 4: Results of hyperparameter optimization

Base filters	Dense units	Blocks	Dropout rate	Val acc
16	32	7	0.215	0.979
16	64	7	0.467	0.976
16	32	8	0.358	0.975
16	32	7	0.3137	0.968
16	32	6	0.886	0.966
8	64	6	0.641	0.954
16	64	8	0.002	0.950
8	32	8	0.133	0.948
8	64	8	0.83	0.944
8	64	6	0.484	0.940
8	64	8	0.146	0.923
8	64	7	0.796	0.911

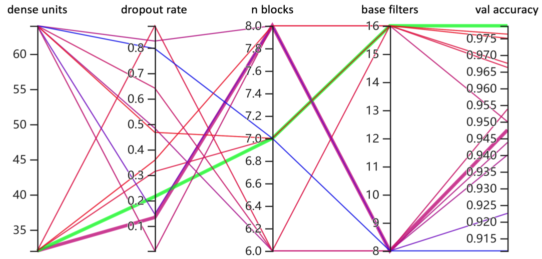


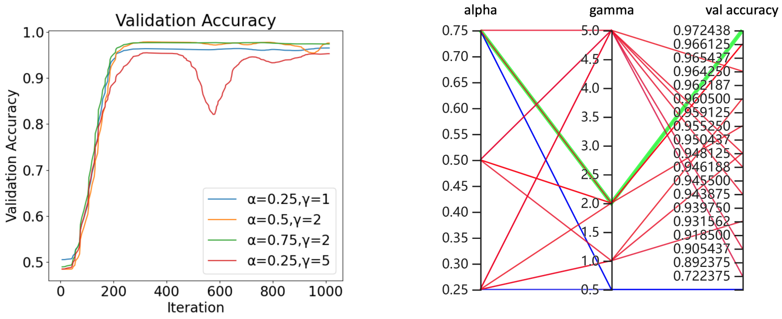
Figure 5: Visualization of hyperparameter optimization

4.2 Focal Loss

The Focal Loss was introduced in the 2017 paper "Focal Loss for Dense Object Detection" by Tsung-Yi Lin et al[3]. The authors showed that this loss function improves the performance of object detection models on imbalanced data sets. In such scenarios, the Focal Loss helps to focus the model's attention on the hard examples that are more likely to be misclassified. The focal loss is calculated as the Equation 1 below. α is the weighting factor, and its role is to balance the contribution of positive and negative examples in the loss calculation. γ is the focusing parameter, and its role is to control the degree of focus on hard examples.

$$FL(p_t) = -\alpha(1 - p_t)^{\gamma} \log(p_t) \tag{1}$$

In order to assess the impact of focal loss, a new dataset was constructed without oversampling, while all other operations were maintained. Using the same loss parameters as outlined in the paper (γ=2, α=0.25) and a consistent random seed, an accuracy of 84.47% was obtained, demonstrating the efficacy of focal loss in addressing imbalanced datasets. The result shows, that the focal loss has a better effect than oversampling in solving imblance problem, as evidenced by the ROC curves as shown in Figure 7. According to the results, γ has a bigger influence on the validation accuracy. The best results appear, when γ is equal to 2. The best combination of γ and α is γ=2, α=0.75 in this problem.



(a)

(b)

Figure 6: The results of different α and γ

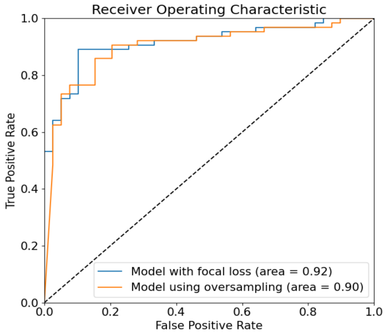


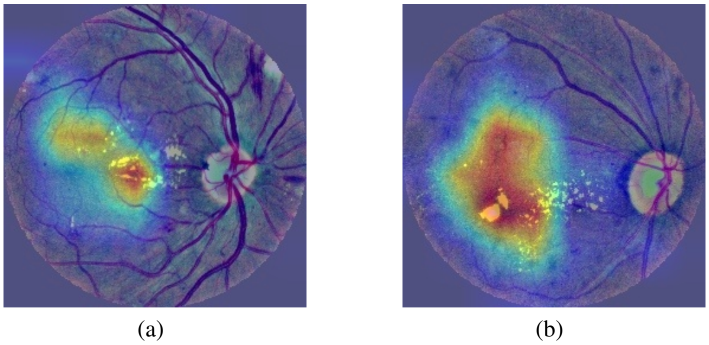
Figure 7: ROC comparation of two models with focal loss and oversampling.

5 Deep Visualization

After training, deep visualization methods such as Grad-Cam and dimensionality reduction are applied to interpret the trained model.

5.1 Grad-CAM

Grad-CAM [4] is a visualization technique that helps interpret CNN predictions. It highlights important regions of an image used by the network to make predictions by computing gradients and weighting feature maps of the last convolutional layer. It is useful for understanding the behavior of CNNs, debugging, fine-tuning, and generating explanations. Grad-CAM results are shown as a heatmap. The red areas overlap with the bright spots of the original image as shown in Figure 8(a) and 8(b), which seems very logical and intuitive.



(a)

(b)

Figure 8: Visualizing VGG based model

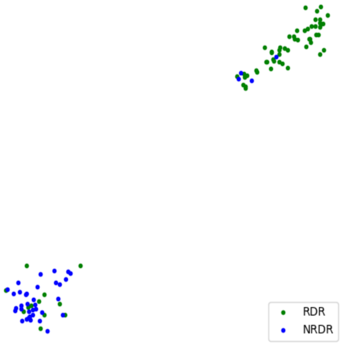


Figure 9: Scatterplot of two classes

5.2 Dimentionality reduction

T-SNE is used to visualize high-dimensional data in a 2D space, showing similar data points close and dissimilar ones far apart. T-SNE has advantages over other Dimensionality Reduction techniques and the scatterplot suggests the ability to separate the two classes with low error rate and separable clusters, with outliers representing hard-to-classify images. See Figure 9.

6 Summary

In this paper, we embarked on a comprehensive approach to construct deep learning models for the purpose of predicting disease severity. As an initial step, we carried out image preprocessing and established an input pipeline, utilizing the Ben Graham method to enhance image quality, which proved to be effective. Subsequently, we conducted a model selection process and determined that VGGNet was the most appropriate model for our task. To address the issue of imbalance in the data, we introduced the focal loss function, which improved the accuracy to 84.47%. In an effort to further optimize performance, we conducted a hyperparameter optimization procedure. Finally, we leveraged ensemble learning to attain an accuracy of 86.4%.

References

[1] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. 2016.

[2] Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556, 2014.*

[3] Lin, Tsung-Yi and Goyal, Priya and Girshick, Ross and He, Kaiming and Doll. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988. IEEE, 2017.

[4] Selvaraju, Ramprasaath R and Cogswell, Michael and Das, Abhishek and Vedantam, Ramakrishna and Parikh, Devi and Batra, Dhruv. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626. IEEE, 2017.