DIABETIC RETINOPATHY DETECTION

A PREPRINT

Shuaike Liu
Institute of Signal Processing and System Theory
University of Stuttgart
Stuttgart, DE
st180457@stud.uni-stuttgart.de

Zhengyu Bao
Institute of Signal Processing and System Theory
University of Stuttgart
Stuttgart, DE
st186295@stud.uni-stuttgart.de

February 13, 2024

ABSTRACT

Diabetic retinopathy is a severe complication of eye disease that is caused by diabetes. This project aims to develop a solution which can detect early symptoms affecting the retina, such as microaneurysms, soft exudates, hemorrhages and/or hard exudates. To achieve a cheap and reliable way to do so, we will be using deep learning models to classify retina images, whether a patient has non-referable or referable retinopathy. With the model VGG (Visual geometry group) and ResNet (Residual neural network) we could finally achieve a test accuracy of 82.5% and 86,4%. Since the dataset is small and imbalanced, we will have to use various kinds of augmentation as well as hyperparameter tuning methods to prevent overfitting, stable training and to achieve higher test accuracies. Lastly, we will visualize the result using Grad-CAM for a better understanding of the machine reasoning.

1 Introduction

In modern day it is becoming increasingly more common to develop diabetes.[1] One of its complications is diabetic retinopathy. It affects eyesight and severe cases can even lead to permanent vision loss for the patient. Since there are no early warning signs and can only be treated if it has been detected in its early stages, a regular retina image examination from a doctor is very crucial. However, the downsides of it are the cost and time complexity. To compensate these issues, a deep learning method shall be developed.

To train the neural network we will use the Indian Diabetic Retinopathy Image Dataset (IDRID). It has a label range from 0 (no apparent diabetic retinopathy) to 4 (proliferative diabetic retinopathy). We will divide the labels into two classes, non-referable (labels 0 and 1) and referable diabetic retinopathy (labels 2 and up). The dataset contains 413 images for training and 103 images for testing, each image has a resolution of 4288 x 2848 pixels. Example images are displayed in Figure 1.

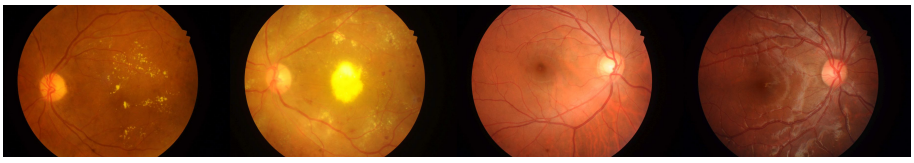


Figure 1: Example images from dataset

2 Preprocessing and Augmentation

In this project, a range of preprocessing and augmentation methods were implemented. Both the training and test datasets were initially loaded, followed by random selection of images from the training dataset to form a validation dataset, the training dataset has a label distribution as shown in Figure 2. This selection was done by using a designated random seed, which also serves as a hyperparameter in the tuning process. The preprocessing step involved resizing each image to dimensions of 256×256 pixels and normalizing the pixel values of all images to fall within the range [0, 1]. This process was applied uniformly across all datasets. Further, each dataset was converted into a TensorFlow dataset for subsequent stages.

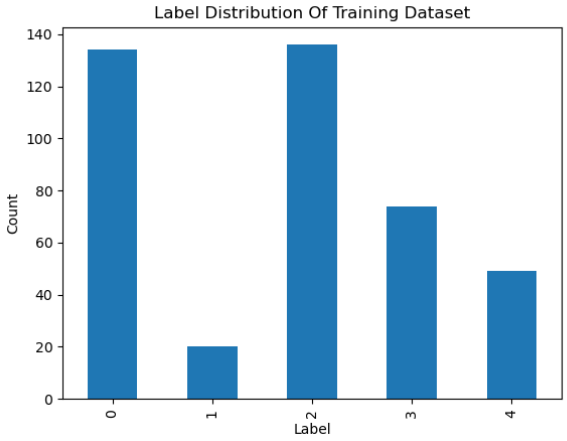


Figure 2: Class distribution across training dataset with a total number of 413 images

Given the limited size of our train dataset, consisting of only 330 images, augmentation becomes a crucial step to prevent overfitting. For the augmentation on the train dataset we mainly used the tf.image module. It incorporates 5 types of augmentation as shown in Table 1. Given the sufficient quality and clinical relevance of the images, it was important to maintain their integrity at a plausible level throughout the augmentation process. That is why we kept the augmentation range as narrow as possible. Following augmentation, the train dataset undergoes shuffling, batching, and prefetching operations to optimize its readiness for model input.

Process	Augmentation Range
random_brightness	max_delta 0.1
random_contrast	0.8 - 1.2 delta
random_saturation	0.8 - 1.2 delta
random_flip_left_right	N/A
random_flip_up_down	N/A

Table 1: Augmentation used in this project

Train dataset	330 images
Validation dataset	83 images
Test dataset	103 images

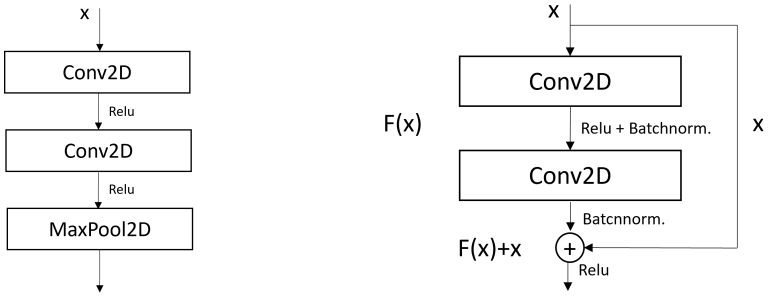
Table 2: Datasets after preparation

3 Model Architecture

In this project, two models were experimented with. The first model is **VGG**, a deep convolutional neural network architecture characterized by multiple convolutional layers. To enhance scalability, the convolutional layers are organized into blocks, each consisting of two convolutional layers followed by a max pooling layer, as illustrated in Figure 3a. Renowned for its simplicity and robust performance, VGG is a widely recognized model architecture in the field of deep learning.

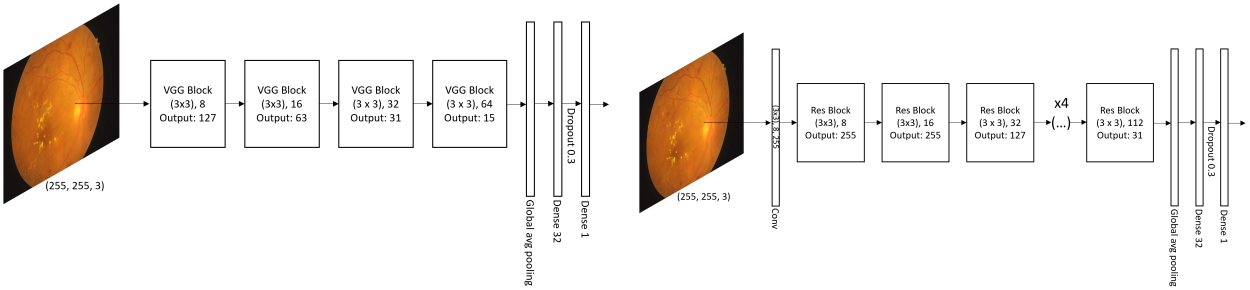
The second model is **ResNet**[2]. We have choosen this model because our initial thought was, that it may be useful to have a deeper neural network, hence this may capture intricate features and patterns from medical images, which is particularly beneficial in tasks like diabetic retinopathy detection where subtle variations in retinal features may

indicate disease progression. ResNet introduces residual connections, or skip connections in the residual block. These connections allow the network to circumvent specific layers during training Figure 3b, thereby addressing the challenge of vanishing gradients during backpropagation by adding the input of the previous layer to the output of a subsequent layer, which is commonly encountered in deep architectures. In the context of this project diabetic retinopathy detection, where large datasets is not available, the inclusion of residual connections may aid in training more effective models with limited data. In Figure 4 both of our model are shown.



(a) VGG block (b) Residual block

Figure 3: Block architecture



(a) VGG architecture (b) ResNet architecture

Figure 4: Our model architecture

4 Hyperparameter Tuning

Hyperparameter tuning is a critical aspect of optimizing deep learning models. We divided our adjustable hyperparameters into 3 groups, as illustrated in Figure 5. These groups are the architecture of a VGG model, the input pipeline configuration, and the architecture of a ResNet model.

For the tuning we followed through a systematic process for these three groups. Initially, the batch size was optimized, based on GPU memory constraints and ensuring tolerable levels of training noise. Subsequently, a range of hyperparameters for constructing a VGG model were systematically explored. Based on the result from the best-performing VGG model, hyperparameters related to the input pipeline were then tuned. Finally, the architecture of the ResNet model underwent the tuning process. This stepwise approach ensured a comprehensive optimization of the model's hyperparameters while considering practical constraints and architectural dependencies. (Both results are shown in Figure 6)

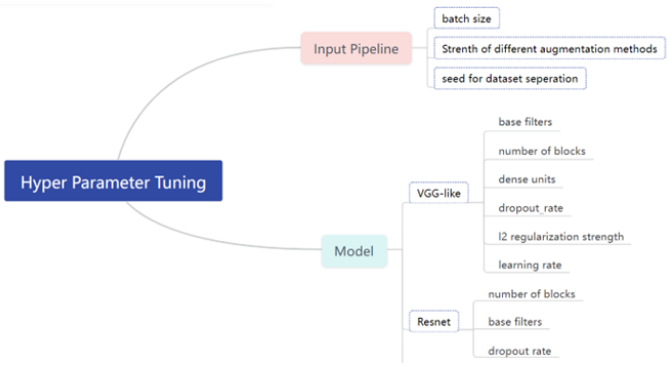
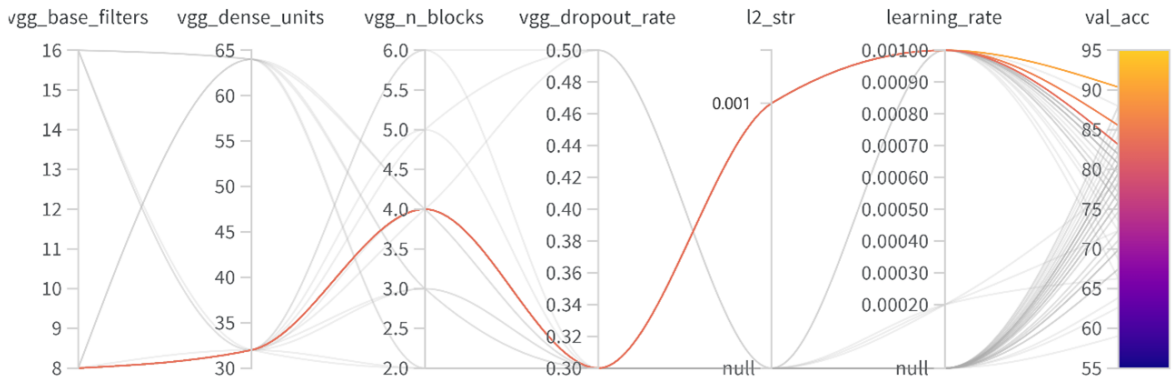
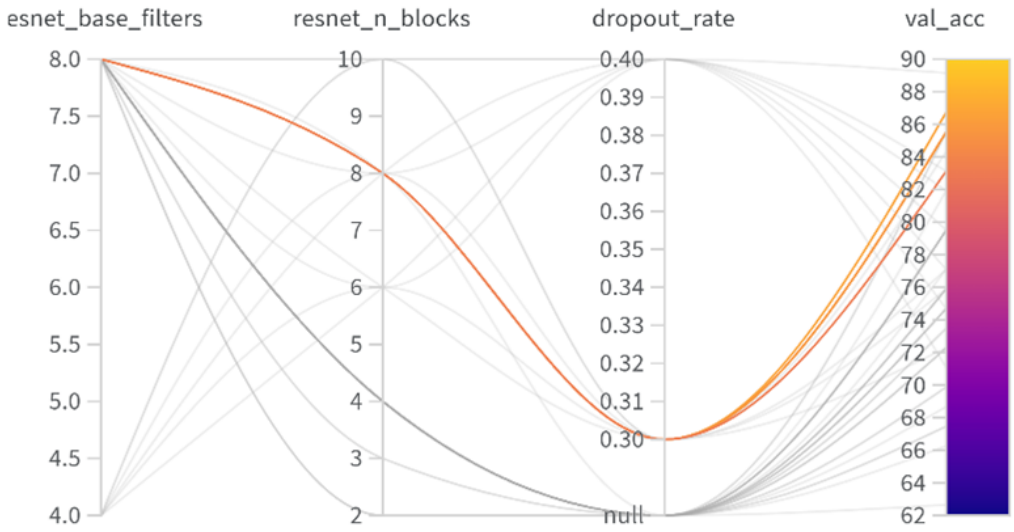


Figure 5: Hyperparameters used for tuning



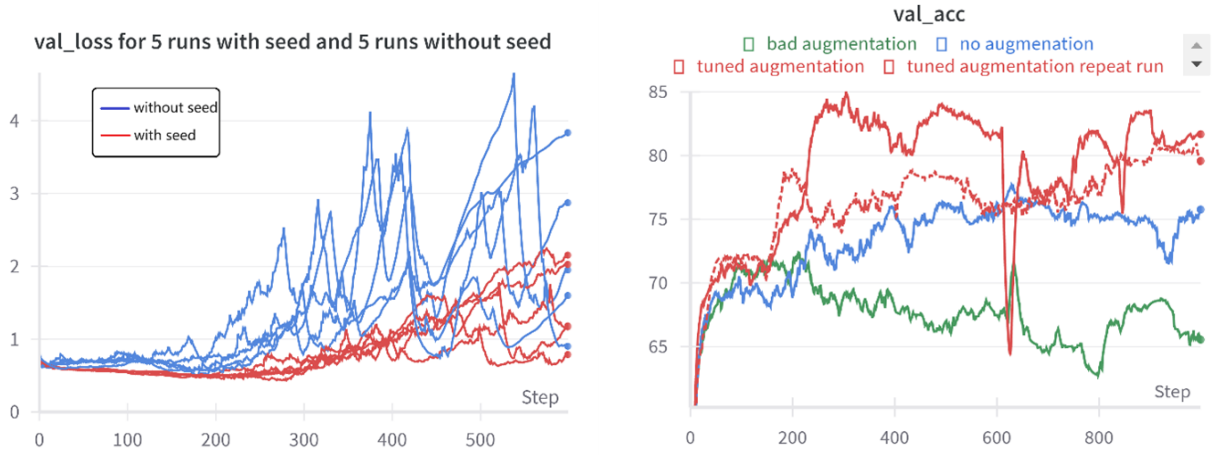
(a) Tuning of VGG, best result marked in red



(b) Tuning of ResNet, best result marked in red

Figure 6: Hyperparameter Tuning

Illustrated in Figure 2, the label distribution shows significant imbalance when considering sub-labels in training dataset. This disparity poses a risk of unstable training outcome caused by random dataset splitting, as depicted in Figure 7a. To address this concern, we devised a strategy outlined in the Chapter 2. By fine-tuning the random seed, we ensure that each label is evenly represented during training. This approach mitigates the impact of imbalanced label distributions, tending more stable and reliable training outcomes. To validate this method, we will be using the test dataset to check the outcome of the training.



(a) Validation loss of VGG for 5 runs with our chosen seed and 5 runs without seed

(b) Validation accuracy of VGG without augmentation, with bad augmentation and with tuned augmentation

Figure 7: Performance comparison

Furthermore, six distinct augmentation techniques were evaluated. Parameters such as the implementation of flipping and adjustments to brightness, contrast, saturation, and picture quality were individually calibrated. Figure 7b shows the validation accuracy trends during training of a VGG model under three conditions: without augmentation, with suboptimal augmentation, and with tuned augmentation.

5 Visualization

Visualization techniques play a crucial role in understanding and interpreting the working mechanism of a deep learning model, particularly convolutional neural networks, since they are often regarded as “black boxes” due to their complex architecture and high-dimensional feature representations.

Visualization provide interpretability by highlighting which part of an input contribute most to the model’s decision-making. Additionally, visualization techniques help diagnose model errors and biases. By visualizing activations and attention maps, we can identify patterns of misclassification, leading to targeted improvements in model design and training strategies. Furthermore, visualizations help the comparison of different model architectures. By visualizing the feature maps, we can tell which model architecture focuses on which feature map.

Among all the visualization techniques, we have chosen Grad-CAM[3] (Gradient-weighted Class Activation Mapping) as our method for visualizing the regions of an input image that contribute most to the model’s prediction. It leverages the gradient information flowing into the final convolutional layer of the network to localize the most salient regions for a specific class prediction. By computing the gradients of the target class score with respect to the feature maps of the last convolutional layer, Grad-CAM generates a class activation heatmap that highlights regions in the input image relevant to the predicted class. These heatmaps provide intuitive visual explanations for the model’s decisions, offering valuable insights into its decision-making process.

The comparison of results between VGG and ResNet architecture is shown in Figure 8 a) - d). Notably, ResNet tends to outline more features distinctly, likely attributable to its deeper neural network architecture. Despite the model’s

proficiency to reliably detect exudates, hemorrhages present a challenge (Figure 8 e) - f), possibly due to the varying brightness of this area across different samples. To address this issue, further preprocessing steps such as the Graham preprocessing method could be considered.

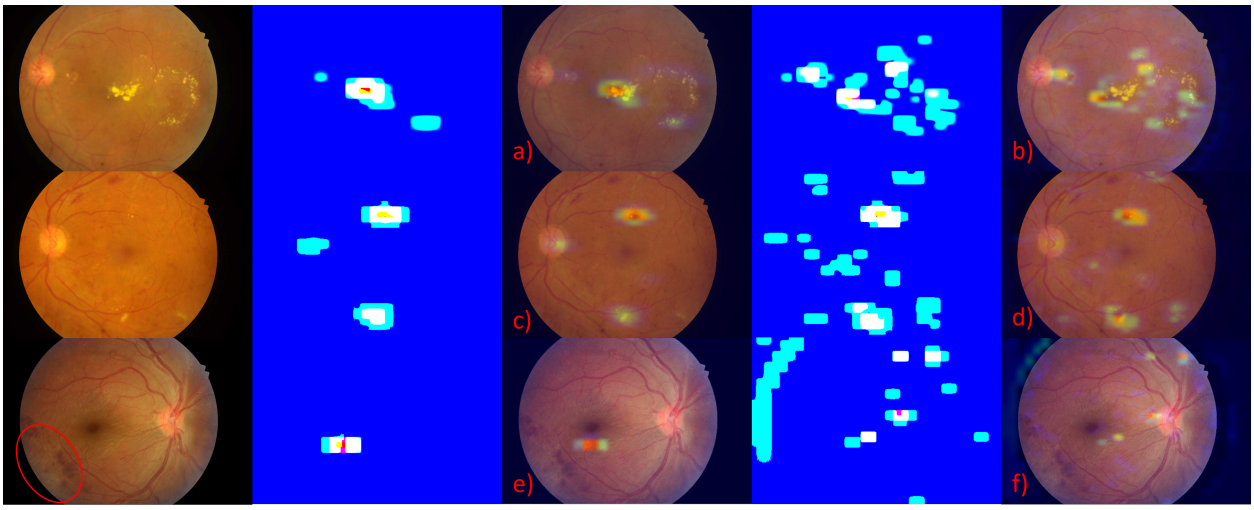


Figure 8: Grad-CAM output from a), c), e) VGG and b), d), f) ResNet. Exudates and even soft exudates could be detected. However, hemorrhages(highlighted in red) could not be detected, both model have predicted 0.

6 Conclusion

In this paper we have observed a variety of augmentation, model architecture, hyperparameter tuning and visualizations for diabetic retinopathy, ultimately selected two models based on our observations (see Table 3 and 4 based on three runs average). We found out the significance of preprocessing, augmentation and model choice, particularly when dealing with small datasets. Additionally, we implemented an idea to optimize the random seed, crucial for achieving a balanced distribution among all subclasses when splitting the dataset.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	58	6
	Negative	12	27

Table 3: Confusion Matrix for VGG with an overall accuracy of 82.5% based on three runs

		Predicted Class	
		Positive	Negative
Actual Class	Positive	55	9
	Negative	5	34

Table 4: Confusion Matrix for ResNet with an overall accuracy of 86.4% based on three runs

Moreover, as our initial thought anticipated, employing a deeper architecture has proven to be more effective in extracting comprehensive information from the data, as demonstrated by the outcomes presented in Chapter 4, 5 and Table, 3, 4. Additionally, the decision to utilize ResNet appears rightful, particularly in combination with a limited small dataset, where the risk of overfitting is typically high.

References

[1] International journal of health sciences. Is diabetes becoming the biggest epidemic of the twenty-first century?, 2007.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[3] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019.