# CSC4080 Midterm Project

Yuancheng Wang

April 4, 2022

**Abstract**

Deep learning algorithms, in particular convolutional networks, have rapidly become a methodology of choice for analyzing medical images. And image classification was one of the first areas in which deep learning made a major contribution to medical image analysis [LKB+17]. However, there are still many challenges in medical image classification, such as data scarcity, class imbalance, explicability, etc. In this project, We designed a an automatic diagnosis software for detecting the degree of diabetic retinopathy by using a subset of the APTOS 2019 dataset. We tried to use some methods to improve the performance of our model from the perspective of algorithm, model and data, and tried to explain our deep learning model.

## 1 Introduction

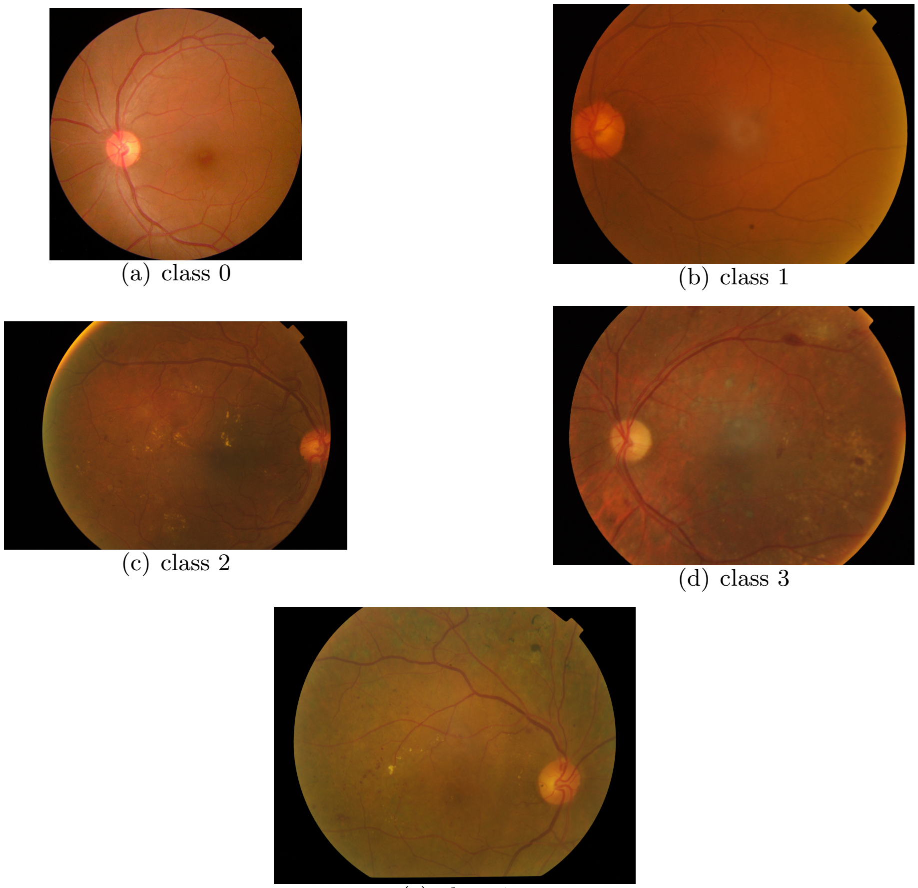
Deep convolutional neural networks [KSH12] have made great achievements in the field of image classification. The application of deep learning in medical image analysis has grown rapidly over the past decade, especially in the field of image classification [LKB+17]. However, compared to traditional image classification problems, medical image classification has some challenges, such as data scarcity, class imbalance, explicability, etc. In addition, how to choose an appropriate model and training techniques also profoundly affects the performance of the model.

Around the above points, we built a classification model based on the APTOS 2019 Blindness Detection dataset. We mainly focus on the following questions: (1) How to choose an appropriate model? Will deeper and larger models show better performance? (2) How to solve the problem of data scarcity? (3) How to solve the problem of unbalanced sample size? (4) How to make the diagnosis more reliable?

The rest of the paper is organized as follows. Section 2 introduces the APTOS 2019 Blindness Detection dataset. Section 3 introduces the baseline model we use and some ways to improve the performance of our model, as well as some training tricks. Section 4 is about experimental hyperparameters and results. Section 5 concludes the project.

## 2 Dataset

Our dataset comes from the APTOS 2019 Blindness Detection dataset. We randomly selected 2000 samples from the training set of the original dataset as our dataset, then randomly chose a training set of size 1600 and a test set of 400. The dataset consists 5 class and each image is rated for the severity of diabetic retinopathy on a scale of 0 to 4. This dataset has the characteristics of most medical image classification datasets: moderate or small amount of data and class imbalance



(a) class 0
(b) class 1
(c) class 2
(d) class 3
(e) class 4

**Figure 1:** Examples of images

problem. The Table 1 shows some details of our dataset and the Figure 1 show some samples for the dataset.

# 3 Method

This section introduces the models and methods we use. We used a convolutional neural network as our baseline model, compared training the network from scratch and fine-tuning with pre-trained weights, and used reweighting and data augmentation to try to overcome the problem of sample imbalance. Finally, we also introduced some training tricks.

## 3.1 ResNet

Deeper neural networks are more difficult to train. He et al. introduced a residual learning framework to ease the training of networks that are substantially deeper than those used previously [HZRS16]. The ResNet architecture consisted of so-called ResNet-blocks which learns the residual. Since ResNets can be stacked by different numbers of residual blocks, we can always easily obtain ResNets of different depths. Residual network models of different depths are suitable for analyzing

**Table 1:** APTOS 2019 dataset information

| **Class Number** | **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| description | No DR | Mild | Moderate | Severe | Proliferative DR |
| Number of train set | 778 (48.6%) | 173 (10.8%) | 434 (27.1%) | 87 (5.4%) | 128 (8.0%) |
| Number of test set | 188 (47.0%) | 34 (8.5%) | 122 (30.5%) | 20 (5.0%) | 36 (9.0%) |

whether our dataset should choose a "deeper" or "larger" model. In this project, we used ResNet18 as our benchmark model and compared 3 different residual network models: ResNet18, ResNet34 and ResNet50.

### 3.2 Transfer learning

Transfer learning is essentially the use of pre-trained networks to try to work around the requirement of large data sets for deep network training [WKW16]. Transfer learning can be briefly summarized as using a pre-trained network as a feature extractor and or fine-tuning a pre-trained network. Since most medical imaging datasets suffer from data scarcity, transfer learning is naturally applied to medical imaging. The present standard is to take an existing architecture designed for natural image datasets such as ImageNet [DDS+09], together with corresponding pretrained weights, and then fine-tune the model on the medical imaging data [RZKB19]. Our project explored whether fine-tuning with pre-trained networks can improve the performance of our models, more precisely, we compared the performance of ResNet18, ResNet34, and ResNet50 with or without pre-trained weights.

### 3.3 Reweighting

Deep neural networks have been shown to be very powerful modeling tools for image classification. However, they can also get bad results if the numbers of samples of some classes in the datasets are very small, or the numbers of samples of different classes are not evenly distributed [RZYU18]. A very natural idea is to modify the weights of different classes in the loss function:

$$w_j = n_{samples}/(n_{classes} \times n_{samples_j}). \tag{1}$$

In our project, We used the data from the training set to modify the weights in the loss function and compared the results.

### 3.4 Data augmentation

Deep neural networks are heavily reliant on big data to avoid overftting. Unfortunately, many application domains do not have access to big data, such as medical image analysis [SK19]. Data augmentation [SLJ+15] is a technique commonly used to address sample scarcity and class imbalance. The image augmentation algorithms discussed in this survey include geometric transformations, color space augmentations, kernel flters, mixing images, random erasing, feature space augmentation, adversarial training, generative adversarial networks, neural style transfer, and meta-learning [SK19]. Limited by time, computing resources, and the characteristics of the medical imaging task itself (we hope that the samples generated by data augmentation are still real images), in our project, we only considered using horizontal flip, vertical flip, and random rotation to implement data augmentation.

### 3.5 Train tricks

Deeper neural networks are more difficult to train. Some network architectures and optimization algorithms can speed up neural network training, reduce overfitting, and improve model performance. For the model, the ResNet uses components such as residual blocks and batch normalization [IS15] to accelerates training and enhances the generalization ability of the network [STIM18]. For the optimizer, we use Adam [KB14] as optimizer, which is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments.

## 4 Experiments

This section is about the details of our experiments and the results.

### 4.1 Setup

Limited by time and computing resources, all our experiments are trained only once under the same random seed, and each experiment trains for 50 epochs. We trained ResNet18, ResNet34 and ResNet50 with and without pre-trained weights (training on ImageNet). In addition, we also trained ResNet18 in reweighting mode with pretrained weights, and ResNet18 and ResNet50 with pre-trained weights in data augmentation mode (without reweighting mode). Table 2 shows hyperparameters used in our experiments (all the experiments use the same hyperparameters).

Our experiments implement data augmentation in the data preprocessing stage. Due to time and resource constraints, our training samples are increased to about 1.5 times the original (1600 to 2463), and the number of test set samples remains unchanged. It is worth noting that the original images of the samples added by data augmentation in the training set are all from the training set. Through data augmentation, we alleviate the class imbalance to some extent. Table 3 shows the class distribution after using data augmentation.

**Table 2:** Hyperparameters used in the experiments

| **Hyperparameter** | **Value** |
|---|---|
| random seed | 42 |
| epoch | 50 |
| minibatch size | 64 |
| learning rate | 0.001 |
| betas (Adam) | (0.9, 0.999) |
| weight decay (Adam) | 0.001 |

### 4.2 Results

We first tested the performance of ResNet18, ResNet34, and ResNet50 with or without pre-trained weights (without using data augmentation and reweighting). The results show that the use of

Table 3: Class distribution with using data augmentation

| Class Number | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Number of train set | 778 (31.6%) | 519 (21.1%) | 434 (17.6%) | 348 (14.1%) | 384 (15.6%) |

pre-training can greatly improve the performance of the model. In addition, without considering pre-training, the performance of the "larger" model is not better than that of the "smaller" model, however, if pre-trained weights are used, the performance of the "deeper" model will be slightly higher than that of the "smaller" model. The experimental results are shown in Table 4.

**Table 4:** The results of ResNet18, ResNet34, and ResNet50 with or without pre-training

| Model | Test accuracy |
|---|---|
| ResNet18 | 75.25% |
| Pre-trained ResNet18 | 80.75% |
| ResNet34 | 74.75% |
| Pre-trained ResNet34 | 80.50% |
| ResNet50 | 75.00% |
| Pre-trained ResNet50 | 81.25% |

The experimental results show that the simple reweighting defined in (1) does not significantly help improve the accuracy of the test set. However, due to the limitation of resources and time, we did not do a comparative experiment of multiple groups of different models, therefore, this conclusion needs to be further explored. In addition, we find that the recall rate of classes with a smaller number of samples is higher in the case of reweighting. The results are shown in the Table 5.

**Table 5:** The results of pre-trained ResNet18 with or without reweighting

| Model | Test accuracy |
|---|---|
| ResNet18 | 80.75% |
| ResNet18 with reweighting | 80.25% |

Finally, we found that data augmentation techniques were helpful for model performance improvement. Although we did not generate more data through data augmentation, according to past research, we have reason to believe that if more samples are generated through data augmentation, the performance of our model can be further improved. The results are shown in the Table 6.

## 5 Conclusion

Finally, we make a brief summary of the content of the project. We designed a an automatic diagnosis software for detecting the degree of diabetic retinopathy by using a subset of the APTOS 2019 dataset. We investigated how to properly select an appropriate model, and our results show that the "larger" model does not There is no clear advantage over the "smaller" model, so the

**Table 6:** The results of pre-trained ResNet18 with or without data augmentation

| **Model** | **Test accuracy** |
|---|---|
| ResNet18 | 80.75% |
| ResNet18 with data augmentation | 82.75% |
| ResNet50 | 81.25% |
| ResNet50 with data augmentation | 83.00% |

simple ResNet18 model is suitable for our task. Then, we explore how pretraining, reweighting, and data augmentation affect model performance. Using pretrained weights can significantly improve model performance, and data augmentation can improve model generalization. However, simple reweighting has a negative impact on overall accuracy. The improvement effect is not obvious, and the recall rate of classes with a small sample size can be improved.

## References


[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[LKB+17] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[RZKB19] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

[RZYU18] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

[SK19] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[STIM18] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[WKW16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.