CS230

Diagnosing diabetic retinopathy from images of the eye fundus

Lucy Wang
Department of Mechanical Engineering
Stanford University
lwang8@stanford.edu

Amelie Schaefer
Department of Mechanical Engineering
Stanford University
amesch@stanford.edu

Abstract

Applying deep learning to automate screening of eye fundus images for diabetic retinopathy can significantly improve patient outcomes by facilitating early diagnoses in areas with low ophthalmologist to patient ratios. To address this problem, we developed a model taking a 3-channel eye fundus image as input and outputting the severity of diabetic retinopathy. Since the data set for this project was relatively small and imbalanced, we applied transfer learning to a pretrained MobileNetV2 and used a weighted loss function. Unfreezing the pretrained weights and incorporating data augmentation proved to be essential for our model's performance, resulting in a final quadratic weighted kappa score of 0.8123 on our test set.

1 Introduction

Diabetic retinopathy (DR) is a common vascular complication of chronic diabetes mellitus in which the retina undergoes several characteristic stages of damage. DR has been found to be the worldwide leading cause of blindness in working aged adults (age 20-65 years) [1]. Usually, the clinical symptoms of the disease progress from microaneurysm formation, small capillary leakages and lipid exudates in the retina to infarctions of the retinal nerve fiber layer (cotton wool spots) followed by extensive retinal fibrovascular proliferation which eventually may lead to retinal detachment [2].In developed countries, diabetic patients are regularly screened for DR and referred to specialists who can initiate medical interventions to avoid progression of the disease. However, in countries or areas with very low ratio of ophthalmologists to patients, regular screening by a specialist is not feasible, thus early symptoms are ignored and treatment is often only provided at late disease stages when invasive methods become necessary. Automatic screening for DR from images of the eye fundus may help with this issue by providing access to regular screenings without the need to consult a specialist. The goal of our project is thus to develop a model based on a convolutional neural network which takes a retina image as input and outputs the correct stage of diabetic retinopathy as seen in the image.

2 Related work

Identifying diabetic retinopathy from fundus images is a fairly well explored task in deep learning. Other groups have achieved excellent accuracy for binary classifiers to identify the presence or absence of DR [3, 4]. A few groups also attempted to apply their model to a multi-class task, in which the aim is to generate a grade of disease severity, ranging from 4 [5] to 5 different classes [6, 7].
Transfer learning from models trained on ImageNet is a common method used to overcome the small amount of training data. Another commonly seen method is data augmentation with random rotations, flips, translations, brightness and contrast changes. Recently published models mainly differ in the type of employed network architecture. Gargeya et al. achieved an AUC of 0.97 on a

CS230: Deep Learning, Winter 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

binary classification task using a residual convolutional net with batchnorm and ReLU activation without transfer learning [3]. On the same binary task, Liu et al. achieved an AUC of 0.9823 using a multipath CNN architecture, in which the activations from each path are weighted and then averaged in order to reduce feature redundancy [4]. They addressed the problem of imbalanced data distribution with targeted data augmentation to balance out the class prevalence in the training data. Zeng et al. developed a siamese-like network structure to grade two fundus images corresponding to the left and right eye of the patient according to 5 classes. The network based on a pretrained Inception V3 uses the correlation between the binocular images to assist the prediction [7] and led to a a weighted kappa value of 0.829. Compared to our task, related recent publications generally had access to more training data with datasets ranging from 13,767 [5] to over 75,000 [3] images.

3 Dataset and Features

We acquired a labeled dataset of a total of 3662 3-channel images from the APTOS 2019 Blindness detection challenge on kaggle [8]. The images were labeled for 5 different categories representing increasingly severe disease stages. Labels and categories as well as the number of examples for each category are given in Table 1. The imbalanced data distribution across the different categories is illustrated in Figure 1.

Label	Category	Examples
0	No DR (normal)	1805
1	Mild DR	370
2	Moderate DR	999
3	Severe DR	193
4	Proliferate DR	295

Table 1: Explanation of labels and class prevalence

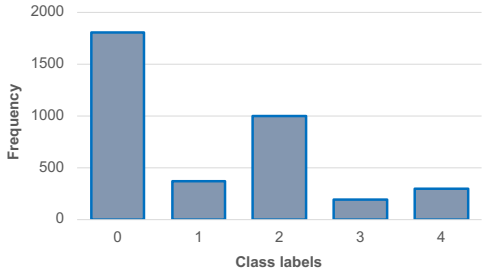

Figure 1: Illustration of class prevalence

We split the data into a training set, a development set and a test set. Due to the overall small amount of data, we decided for a 70/15/15 split, resulting in a training set of 2564 images and development and test sets of 550 images each. The images of each category were shuffled before sorting them into the different datasets, but the distribution of class prevalence was preserved for all three sets. Since the images in the dataset were taken under various lighting conditions with varying camera equipment, the set contains a wide range of image quality and resolutions. In order to have the option of different input resolutions for our model, we resized all raw images to 128x128 and to 224x224 pixels and scaled the channel values to be between 0 and 1. In a further attempt to improve our model performance, we later changed our input to images from the same challenge, that were already preprocessed to account for different lighting conditions [9]. Since the dataset is small for a deep learning task, we used data augmentation to generate additional training examples. Figure 2 shows examples of preprocessed images from different categories, from normal to proliferate diabetic retinopathy.

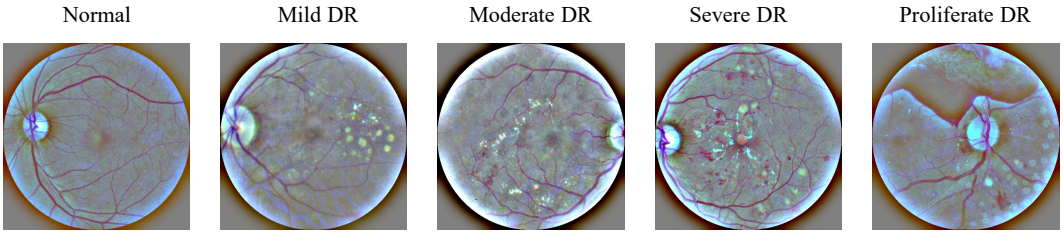

Figure 2: Examples of preprocessed images from the different categories, increasing disease severity.

4 Methods

In order to address the limited amount of labeled data for our application, we used transfer learning with a pretrained MobileNetV2 [10] which is designed to run efficiently on mobile devices. We chose this model, because employment on a mobile device could be one way to bring our application to the user. A key feature of the MobileNet base model is the use of depthwise separable convolutions, in which the full convolutional operation is split into two separate layers, one depthwise convolution with a 3x3 kernel, followed by a 1x1 pointwise convolution. This strategy allows for a reduction in computational cost of a factor 8 to 9 compared to standard convolutions, making it affordable for deployment on a mobile device. Each of the convolution operations is followed by a batch normalization and ReLU6 activation. In MobileNetV2 the building blocks are transformed into so called bottleneck residual blocks. These blocks contain an expansion layer, a depthwise convolution and a projection layer (see Figure 3). The block is called bottleneck because the projection layer at the end of the block reduces the number of channels. The connection skipping the layers has a similar function as in a residual net.

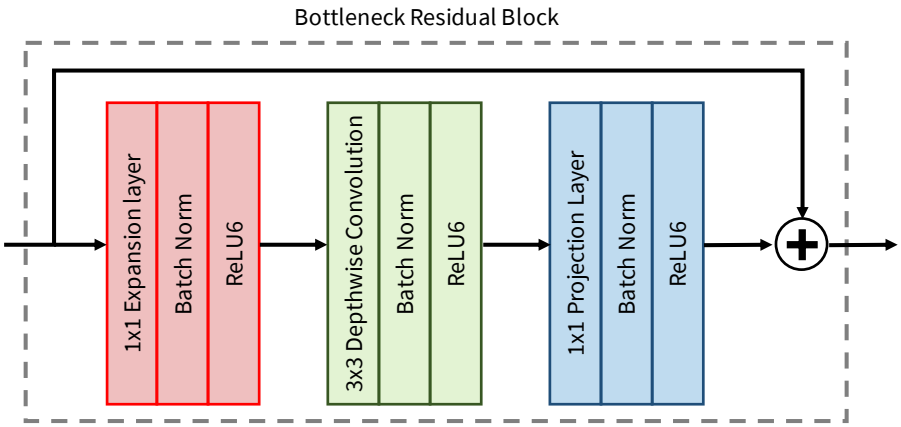

Figure 3: Illustration of bottleneck residual block in MobileNetV2 (diagram adapted from https://machinethink.net/blog/mobilenet-v2/).

In order to adapt the pretrained model to our task, we added to the base model a global average pooling layer followed by a fully connected layer with 5 neurons for our 5 categories and softmax activation. The architecture of the modified model is shown in Figure 4.

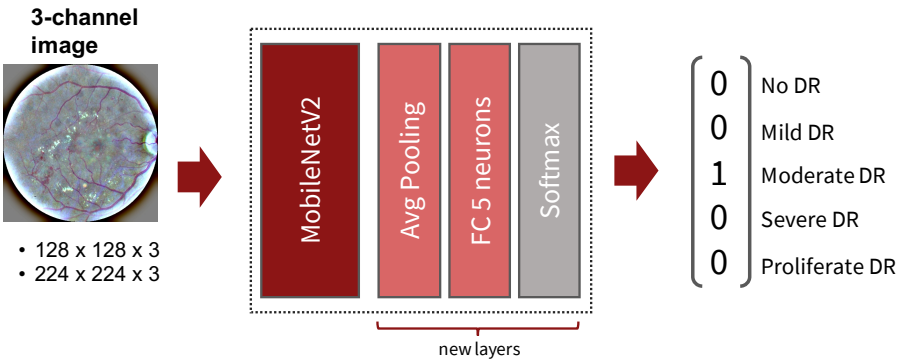

Figure 4: Illustration of modified model architecture with input and output.

For training our model, We chose to use the categorical cross entropy loss function and Adam optimization with a batch size of 32. To evaluate the performance of our models, We used categorical accuracy as a metric. Categorical accuracy evaluates how often predictions have the maximum at the same index as the ground truth label. We made these choices based on observations of what had been used in previous work as well as ease of implementation using built-in Keras functions.

5 Experiments/Results/Discussion

5.1 Experiments

At first, we decided to freeze all of the parameters in the pretrained model, training only the parameters of the last fully connected layer. In the first attempt, we trained a model on images of size 128x128x3 using a learning rate of 1E-04 for 10 epochs. This first attempt resulted in a training accuracy of 0.7168 and a dev accuracy of 0.6158. To address the high bias, we decided to increase the image resolution to 224x224x3. This increased the training accuracy, but only slightly, to 0.7267. Keeping the higher image resolution, we then compared three different learning rates: 1E-04, 1E-05, and 1E-03. The model trained using the slowest rate (1E-05) was trained for 50 epochs instead of 10. While the 1E-03 learning rate produced the highest training accuracy, the validation loss consistently increased during training, so we decided in favor of our original rate of 1E-04. The next step we took in hopes of decreasing bias was to unfreeze the weights of the pretrained model. This successfully increased the training accuracy from 0.7267 to 0.9767. Variance, however, remained high with the dev accuracy increasing only from .6158 to .6324.

At this point, we took a look at the model's predictions and noticed that the model was predicting only class 0 and class 2. Thinking this was due to the higher number of examples in these classes compared to the other three, we introduced class weights to our loss function. After incorporating the weighted losses, the model began to predict all five classes. The dev accuracy, however, dropped from 0.6324 to 0.5827. Looking at the accuracy, it seems the weights negatively impacted the model's performance. However, reflecting on these results, it seems that accuracy was likely not the best choice of metric for this problem. Comparing, instead, using the quadratic weighted kappa, we see that the kappa score increases from 0.4180 to 0.6833. The kappa score, unlike the categorical accuracy, takes into account both the probability of agreeing by chance and the fact that some categories are more similar to each other than others.

Turning now to address the high variance, we first tried using the preprocessed data set described in Section 3 but saw no change in model performance. We then tried applying data augmentation through horizontal and vertical flipping and random rotations of up to 360 degrees. This successfully decreased the variance, resulting in a final training accuracy of 0.8327 and dev accuracy of 0.7309.

5.2 Results

Our final model achieved training, dev, and test accuracy of 0.8327, 0.7309, and 0.7847, respectively. Quadratic weighted kappa scores for the three data sets were 0.8468, 0.7761, and 0.8123, respectively. Looking at the confusion matrix in Figure 5 and the metrics in Table 2, it is clear that the model performs the best on classes 0 and 2, which are the classes with the most training examples.

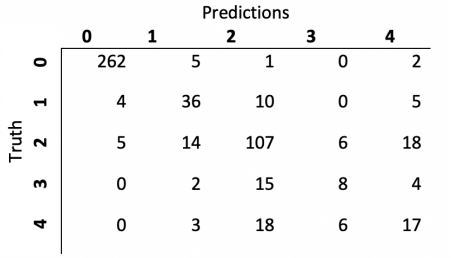

Figure 5: Confusion matrix for the final model using the test data set.

Label	Precision	Recall
0	0.9668	0.9704
1	0.6000	0.6545
2	0.7086	0.7133
3	0.5714	0.2759
4	0.5862	0.3864

Table 2: Precision and recall of final model on the test set.

The saliency maps we used to visualize our results show that the model sometimes correctly uses pathological features to perform the classification (Figure 6, left). In several other cases, however, the model is giving importance to the corners of the image and the circular border (Figure 6, center). These areas do not actually contain information relevant to disease classification. Furthermore, the pathological features present in diabetic retinopathy vary widely from capillary leakage to cotton wool spots. Our model successfully recognizes some features like lipid exudates (Figure 6, left) but fails to recognize others such as hemorrhages (Figure 6, right).

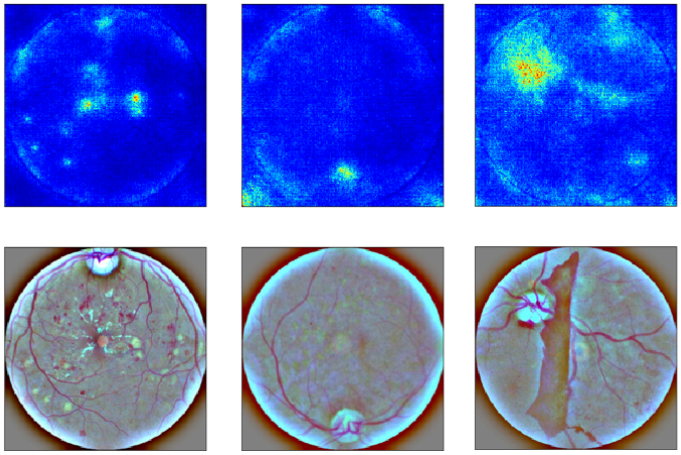

Figure 6: Saliency maps showing desired behavior of the model (left), undesired attention in irrelevant portions of the image (center), and ignored pathological features (right).

5.3 Discussion

Looking at the results from our model, the effects of a small training data set are evident. The precision and recall scores directly correlate with the amount of training data available for each class. Obtaining more data to balance out the classes would improve model performance. In gathering this new data, it would also be important to obtain greater representation of certain pathological features such as hemorrhages, which the model currently does not recognize as being important for classifying the images. Despite relatively low performance on the multiclass task, our model could perform quite well at a binary classification task of healthy vs. diseased. Looking at the confusion matrix, the diseased categories (1-4) are most often misclassified amongst each other rather than as the healthy class 0.
Reflecting back, it would have been better to use the quadratic weighted kappa as the metric because the unbalanced data set led to misleading accuracy values. Unfortunately, we did not realize this at the beginning. As a result, the hyperparameters we explored first using only the categorical accuracy may not be optimized for our desired model performance.

6 Conclusion/Future Work

The main challenges in this project were the small amount of data and imbalanced classes. To address the small data set, we chose to apply transfer learning to a pretrained MobileNetV2. Incorporating a weighted loss function turned out to be essential for addressing the imbalance. Our initial attempts at training resulted in both high bias and high variance, but we found that unfreezing the MobileNetV2 weights and including data augmentation were the most effective strategies for reducing bias and variance, respectively. Looking forward, if we had more time and resources, we would implement L2 regularization and add more data augmentation approaches to bring the variance down further. We would also analyze the saliency maps more closely as an error analysis strategy to brainstorm ways to improve the model. For example, in some saliency maps, it is apparent that the model is paying too much attention to irrelevant portions of the images. One possible way to improve the model would be to crop out the irrelevant corners of the input images. Finally, our model would greatly benefit from additional data to balance out the classes and to increase representation of rarer pathological features.

7 Contributions

Both team members collaborated on developing and adjusting the code and writing the project report. Lucy Wang focused on analysis of results and documentation throughout the model optimization. Amelie Schaefer focused on literature review and design and recording of the poster.

8 Code

Github URL: https://github.com/wangLucyM/CS230-Project

References

[1] R Klein, BEK Klein, SE Moss, MD Davis, DL DeMets, Wisconsin Epidemiologic Study of Diabetic Retinopathy, et al. Prevalence and risk of diabetic retinopathy when age at diagnosis is less than 30 years. *Arch Ophthalmol*, 102(4):520–6, 1984.

[2] Daniel E Singer, David M Nathan, Howard A Fogel, and Andrew P Schachat. Screening for diabetic retinopathy. *Annals of Internal Medicine*, 116(8):660–671, 1992.

[3] Rishab Gargeya and Theodore Leng. Automated identification of diabetic retinopathy using deep learning. *Ophthalmology*, 124(7):962–969, 2017.

[4] Yi-Peng Liu, Zhanqing Li, Cong Xu, Jing Li, and Ronghua Liang. Referable diabetic retinopathy identification from eye fundus images with weighted path for convolutional neural network. *Artificial intelligence in medicine*, 99:101694, 2019.

[5] Wei Zhang, Jie Zhong, Shijun Yang, Zhentao Gao, Junjie Hu, Yuanyuan Chen, and Zhang Yi. Automated identification and grading system of diabetic retinopathy using deep neural networks. *Knowledge-Based Systems*, 175:12–25, 2019.

[6] Feng Li, Zheng Liu, Hua Chen, Minshan Jiang, Xuedian Zhang, and Zhizheng Wu. Automatic detection of diabetic retinopathy in retinal fundus photographs based on deep learning algorithm. *Translational vision science & technology*, 8(6):4–4, 2019.

[7] Xianglong Zeng, Haiquan Chen, Yuan Luo, and Wenbin Ye. Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network. *IEEE Access*, 7:30744–30753, 2019.

[8] kaggle. *APTOS 2019 Blindness Detection*. Available at: `https://www.kaggle.com/c/aptos2019-blindness-detection`.

[9] Ratthachat. *APTOS : Eye Preprocessing in Diabetic Retinopathy*, Sep 2019. Available at: `https://www.kaggle.com/ratthachat/aptos-eye-preprocessing-in-diabetic-retinopathy`.

[10] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.