Diabetic Retinopathy Detection via CNN

Michael Dermksian
Carnegie Mellon University
mdermksi@andrew.cmu.edu

Harsh Dhruva
Carnegie Mellon University
hdhruva@andrew.cmu.edu

Michael Turski
Carnegie Mellon University
mturski@andrew.cmu.edu

Eric Rasmussen
Carnegie Mellon University
erasmuss@andrew.cmu.edu

Abstract

Diabetic Retinopathy (DR) is a complication resulting from diabetes in which lesions on the retina form, affecting vision. In this paper we apply several existing image classification convolutional neural network (CNN) architectures to the detection of DR to evaluate their capability. We further evaluate breaking the categorical classification problem into multiple binary classification problems in hopes of increasing the overall model performance. We also incorporate image preprocessing techniques that aid the CNN's classification ability.

Our preliminary results conclude that state of the art CNNs, such as VGG16, fall short of being able to fully classify all levels of DR, achieving 80% accuracy. To combat this, we used binary classification to detect the presence of DR, resulting in 97% accuracy. Due to this result, we are confident that CNNs can accurately detect DR. Finally, we attempted to break the classification down further into binary classification of early stage DR versus late stage DR, and binary classifications of each subset of the early stage and late stage labels. Our results showed that while this binary approach improved classification of the healthy images, overall, this technique did not improve the overall categorical classification of the varying levels of DR.

1. Introduction

Diabetic Retinopathy (DR) is a complication resulting from diabetes in which long term high blood sugar causes blood vessels on the retina to become damaged, affecting vision. In its early stages, DR has only a mild effect on vision, but if allowed to proliferate, DR can eventually lead to blindness. Diabetic individuals with either type 1 or type 2 diabetes are at risk for the development of DR.

The long-term negative effects can be effectively managed and often prevented if presence of the disease is detected early. To this end, it is common for diabetic individuals to have regular eye examinations at which their physician will photograph the rear of the eye (termed the fundus). These fundus images can be examined for features such as microaneurysms, hemorrhages, hard exudates, and soft exudates. Each of these lesions present as discolored regions in the fundus image. Examples of these discolored regions can be found in Figure 1.

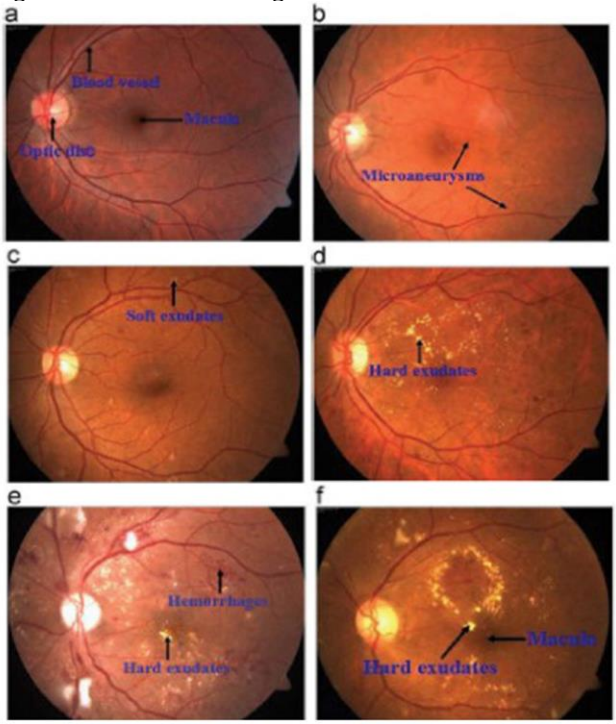

Figure 1: Six fundus images indicating the various image features associated with DR. Accessed from [1].

Convolutional neural networks (CNN) have been shown through many experimental findings to be highly capable tools for classifying images. CNNs present those interested in imaging-based disease diagnosis with a tool for automating this task. If a machine learning method can be shown to reliably detect the disease, or detect it more reliably and at an earlier stage than human clinicians, it has the potential to introduce tremendous value to the field of diagnostics. In the case of DR, early detection is key to preventing the complications that occur as the disease proliferates. Additionally, the indicators of the disease require trained clinicians to detect, and human error is a concern with image-based diagnoses. Highly repeatable

machine learning techniques could simultaneously reduce the risk of human error and remove the need for high-paid skilled labor. This may indirectly reduce the cost of care.

In this paper, we demonstrate our experiments with various CNN models tasked to classify the fundus images to identify the level of DR. Through hundreds of trained models and various techniques explored as described throughout the paper, we achieved successful results in the detection of DR as binary classification, and substantial results in classifying the images into each specific category, or levels of DR. We further explored results of breaking down the categorical classification to several binary classification problems, to understand the behavior of the network in identifying specific levels of DR.

2. Related Work

The DR detection task via CNN has been attempted by many in the past with varying results. Our work is an attempt to replicate their individual results and synthesize the techniques of several independent teams. Additionally, we attempt to introduce some novel efforts by distributing this multi-class classification task across several CNNs.

Prior efforts for classification of fundus images were largely discovered in the meta-analysis presented by Alyoubi et al. [1]. This analysis and the works it references form the foundational body of work for our experiments.

Inspiration for the usage of AlexNet and VGG architectures comes from Wang et al. [2], Wan et al. [3], and Mobeen-ur-Rehman et al. [4]. Inspiration for the usage of ResNet architectures comes from Wan et al. [3] and Zhang et al. [5]. Image preprocessing techniques were also inspired by literature. Our usage of Gaussian filtering and CLAHE contrast enhancement was inspired by Yan et al. [6] and Wu et al. [7].

3. Data

While there are many possible sources of data for fundus images, we relied primarily on the Messidor-2 and Kaggle datasets, described herein. An acknowledgement must be made that the results described can only be as good the quality of the dataset on which they are trained. Particularly in the case of DR, where fundus images are classified by clinicians, any biases that exist in the human-generated labels will be transferred to the machine learning model.

3.1. Messidor-2 [8] [9]

The Messidor-2 dataset is a collection of 1,748 images corresponding to right and left fundus images from 874 examinations. It is a combination of two groups of images, one set from 529 examinations, which comprised the original Messidor dataset, and another set of 345 examinations. This dataset is comprised of very high-resolution images taken with consistent equipment and field of view. These images are almost entirely 2240 x 1488 pixels each and are full RGB. Labels for the dataset are generated by retina specialists and accessed via Kaggle.

3.2. Kaggle [10]

The Kaggle dataset is comprised of 88,704 images representing a combination of images from the 2015 *Diabetic Retinopathy Detection* competition with images provided by EyePACS, and images from the *APTOS 2019 Blindness Detection* competition. Fundus images within this dataset are significantly less consistent in quality and framing. These images are also full RGB, but much less consistent in size and aspect ratio than Messidor-2, ranging from 2612 x 1964 pixels to 221 x 205 pixels. However, the overall size of the dataset is attractive as CNNs often require large datasets to effectively train.

A Gaussian filtered subset of this dataset was accessed via Kaggle [11]. This dataset consists of 3,632 images with a Gaussian filter applied. Labels match the original Kaggle dataset.

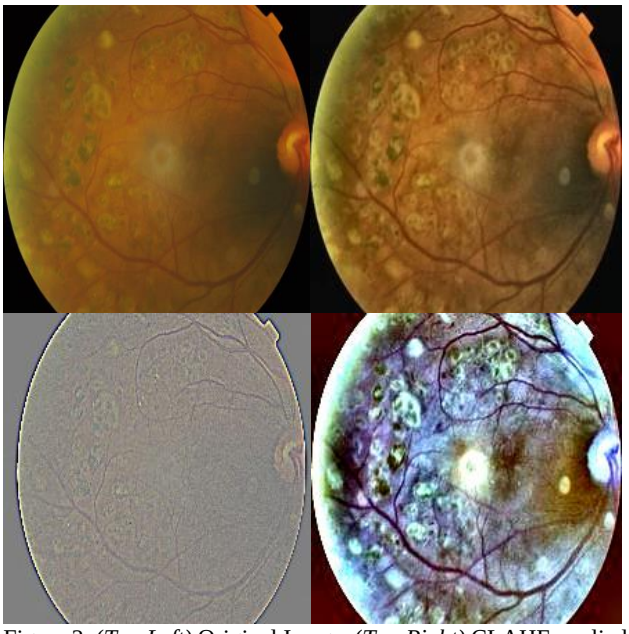

Figure 2: (*Top Left)* Original Image, (*Top Right)* CLAHE applied, (*Bottom Left*) Gaussian Filter applied, (*Bottom Right*) CLAHE + Gaussian Filter with Color.

3.3. Image Preprocessing

The images from the Messidor-2 dataset are similarly sized, but the irregularity of the Kaggle datasets forced us to preprocess all the images to be the same size. We chose to resize all images to be 224 x 224 pixels, as this allowed us to use ImageNet weights. The ImageNet weights are sets of layer weights used to initialize the training of a new CNN, which have been previously generated by training various CNN architectures on ImageNet, a large dataset

containing over 30,000 classes of images. Also, at this resolution, the features associated with varying levels of DR are still visible to the human eye.

After resizing, we applied various filters to the images for testing. These include Gaussian filters and contrast limited adaptive histogram equalization (CLAHE). These filtered images were treated as separate datasets to use for training the CNNs, which would be used in comparison with the original, non-filtered images. Figure 2 shows the effects of applying various image preprocessing techniques to the images.

3.4. Image Augmentations

Image augmenting is a common strategy in image classification to prevent overfitting based on location of edges, colors, or other generic features within the picture. To this end, images within the training set are flipped, rotated, skewed, etc. during the training of the network. Throughout our testing, we utilized width shifts, height shifts, rotations, brightness alterations, standardization, zoom alterations, horizontal flips, and vertical flips.

4. Methods

Our team approached this DR classification problem by using well-researched image classification networks such as AlexNet, VGG16, and ResNet50. These CNNs were tested on various datasets (filtered and non-filtered) and observations were made. Using these architectures, we experienced overfitting and biasing towards specific labels. To combat this, we explored the use of various image augmentations and image resizing strategies. We also attempted to model our own CNN, based on the architectures of the above-mentioned networks, by modifying various layers and hyperparameters. However, we did not achieve any success using our own CNN.

4.1. AlexNet

As one of the first highly successful CNNs, AlexNet inspired many of the more recent architectures. AlexNet contains 5 convolutional layers and 3 fully connected layers [12]. ReLU activation functions are used to introduce nonlinearity into the system, with the inclusion of dropout layers instead of the commonly used L2 regularization to prevent overfitting.

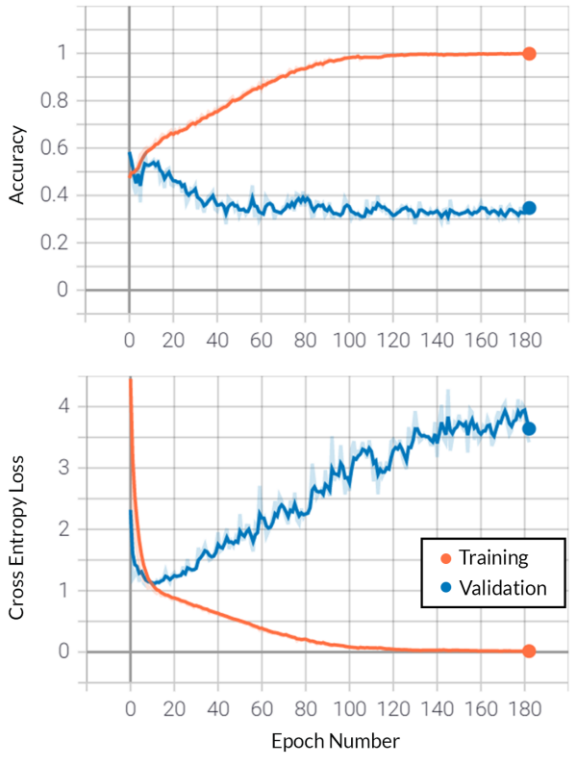

Figure 3: Accuracy and loss of the training and validation data for AlexNet trained on the Messidor-2 dataset. Despite initially regressing together, the train accuracy drives to 100%, while the validation accuracy falls to around 35%.

During our experimentation, we trained a generic AlexNet model on both the Kaggle and Messidor-2 dataset. Loss and accuracy during training can be seen in Figure 3. We experienced extreme overfitting with this model. Instead of learning the necessary filters and features, AlexNet seemed to memorize the images, resulting in a significant gap between training and validation accuracies and losses.

We attempted to mitigate this overfitting using image augmentation during training. Additionally, different batch sizes, loss functions, regularization techniques, and adjustable learning rates were utilized to attempt to improve accuracy. Even with these measures, we either experienced overfitting, or biasing to the 0 label. In the latter case, due to our datasets primarily being composed of healthy images (label 0), we found that AlexNet would predict the majority of each class as 0 to easily maximize the accuracy. This result can be seen in the confusion matrix for training runs, seen in Figure 4. Unfortunately, this issue proved hard to overcome in AlexNet.

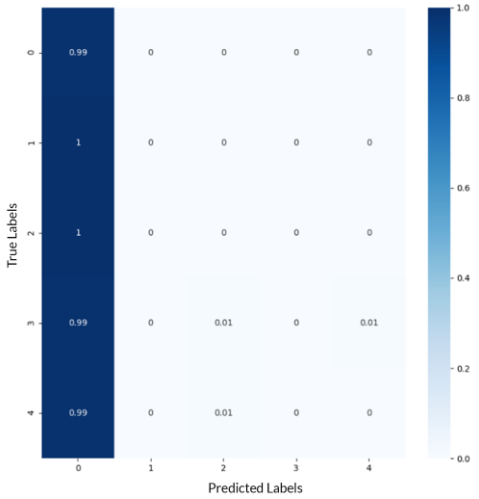

Figure 4: AlexNet confusion matrix illustrating the problem of biasing towards the 0 label. In this case, AlexNet has been training for 30 epochs, with training and validation accuracies of around 73.5%.

4.2. ResNet

ResNet [13] introduces residual learning in the form of shortcut connections and adds more layers to improve performance. The shortcut connections perform identity mapping to prevent additional parameters from being introduced into the model, cutting away computational time and increasing accuracy in the process. ResNet models range from 18 to 152 layers. Our team implemented the 50-layer model. ResNet50 consists of a series of convolutional layers with the shortcut connections, ending with an average pooling and a fully connected layer with a softmax function for classification.

Our results with ResNet were mixed. Like AlexNet, the model often overfit to our dataset, albeit with slightly higher validation accuracies. About 80% accuracy was achieved on our validation set with over 95% train accuracy.

4.3. VGG

VGG is another successor to AlexNet created by the Visual Geometry Group of Oxford University. Like AlexNet, it uses multiple convolutional layers with max pooling [14]. The architecture ends with three fully connected dense layers with a softmax activation function.

The VGG architecture is the model we achieved our highest success with. We attempted to use VGG19 but were not successful in achieving better results than VGG16. Using VGG16 initialized with ImageNet weights on the Gaussian filtered dataset, we achieved a training and validation accuracy of 95+ and 80% respectively for the 0-4 classification. The training of this VGG model can be seen in Figure 5, and the resulting confusion matrix can be seen in Figure 6. This specific trial resulted in a CNN that was clearly able to discern between label 0 (healthy) and all other labels. However, this VGG struggled to differentiate between classes 1-4 (varying levels of DR). When determining class 0 from the other classes, VGG16 had a precision of 96.2%, recall of 98.6%, and F1 score of 97.45%.

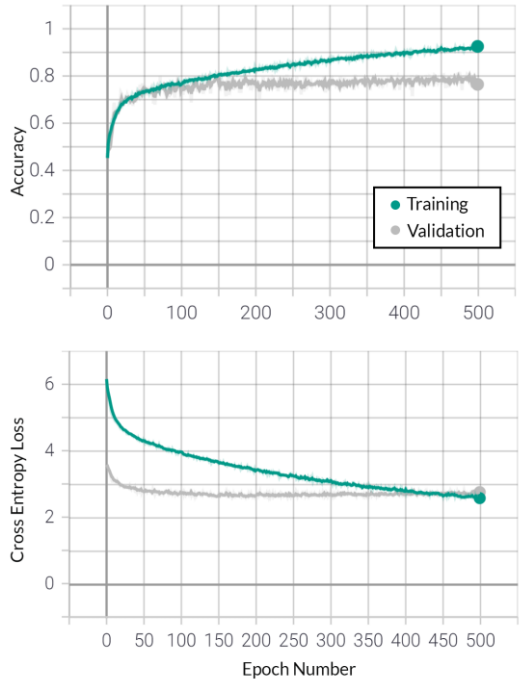

Figure 5: *(Top)* Test and train accuracy of the VGG16 network versus epoch number *(Bottom)* Test and train loss versus epoch number.

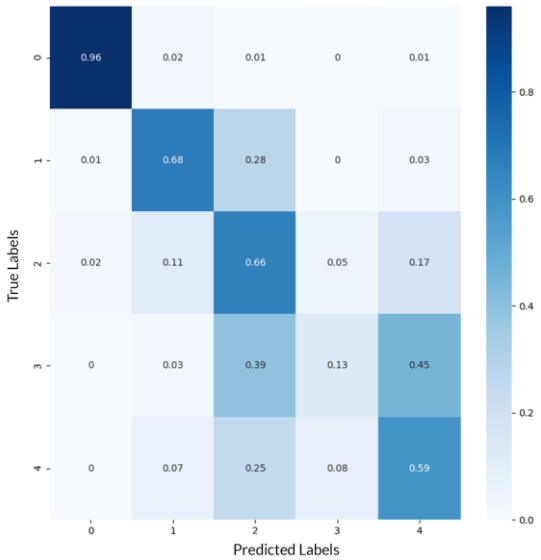

Figure 6: Confusion matrix for VGG16 trained on all labels 0 – 4.

4.4. Class Separation with Binary Classification

This VGG result proved that a CNN would potentially be able to correctly classify a healthy image, despite struggling to discern between the various severities of DR. This result also follows those of other groups who have attempted to classify DR. Upon further research, we found that others had had some success with breaking the dataset into a binary classification problem [1].

In similar fashion, we proceeded to further split this 0-4 classification problem into multiple sub-classification problems using binary classification. Our initial thinking was that we would be able to achieve a higher overall accuracy.

To do so, we first split and altered the labels of our data to separate into a binary classification. All afflicted cases, 1-4, were given a new label of 1, while the healthy cases remained a 0 label. Using VGG16 and the previously mentioned Gaussian filter, we equally split the data points between classes and began testing again with all previously mentioned augmentations and adaptable variables. Within 80 epochs, we were able to record a testing and validation accuracy of over 96% and 97% respectively. Full depictions of the run can be seen below in Figure 7. Given this great result, we felt confident that the model alone would be suffice when trained professionals were in a pinch.

This high accuracy received on this test alone shows promise to be used as an automated retinopathy detection program. Catching this disease early for treatment is the most important factor for the real-life application of retinopathy detection.

5. Experimental Results

As discussed in the Methods section, we propose that DR can be fully classified using binary classifications to differentiate between classes that may be difficult in one-shot categorical classification. This section will provide a look into the various experiments we conducted to reach our resulting full binary classification.

5.1. DR Binary Detection

The first step in this classification of DR is to perform binary classification to determine if the image is healthy versus afflicted. We trained this model by combining labels 1-4 (afflicted) and comparing them to label 0 (healthy), as discussed earlier. Using VGG16 initialized with ImageNet weights, updated class weights to accommodate for imbalance in the data, various data augmentations, and L2 regularization, trained on the Kaggle gaussian filtered dataset, we achieved training and validation accuracies of 99.4% and 98% respectively. We further identify that our binary classification has 98.1% precision, 99% recall, and a 98.6% F1 score. This is a marginal improvement over the prior result when classifying with the full set of labels. These results are displayed in Figure 7 and Figure 8.

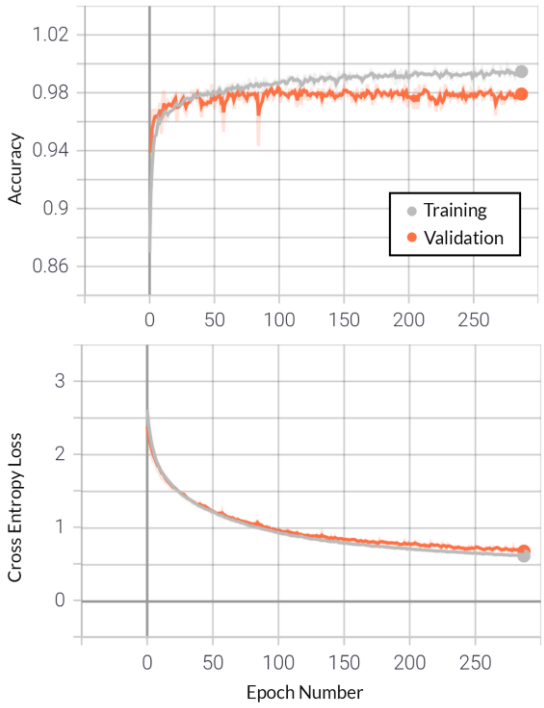

Figure 7: Accuracy and loss vs epoch for binary classification between afflicted and unafflicted groups.

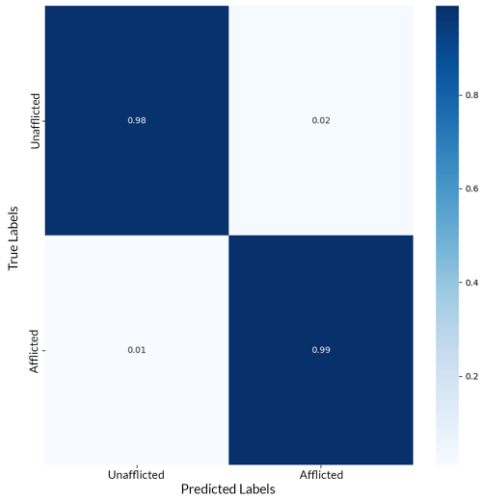

Figure 8: Confusion matrix for binary classification between afflicted and unafflicted groups.

With these preliminary results, we are confident that we can discern between healthy and afflicted images. Now, we can exclude the healthy labels from further trials, as we can

identify them with this model. This leaves the afflicted images, labels 1-4, to be classified.

5.2. Afflicted Categorical Classification

Given the high accuracy of the binary separation, we can now focus on the separation of the four remaining unhealthy groups. During this classification between all the afflicted cases, 1-4, our team introduced the CLAHE filter along with the normal Gaussian filter to enhance visible features within the eyes. Given the previous success with the VGG architecture, we used it as our main model architecture. Unfortunately, the training and validation accuracy only resulted in around 75% and 65% respectively. The confusion matrix of the validation set, seen in Figure 9, gave us intuition on how to proceed.

Only looking at the training and validation accuracies, our network was not very successful at this one-shot categorical classification of the afflicted images. However, looking at the confusion matrix in Figure 9, it appears the network was able to discern between the image either being in class 1 or 2, and the image being in class 3 or 4. It struggles more when trying to differentiate between 1 and 2, and between 3 and 4. This could imply that the differences of features between cases 1&2 and 3&4 respectively might not be large or distinct, and thus cause the training of the network to suffer. We further investigate this classification challenge by using binary classification between these sub-groups.

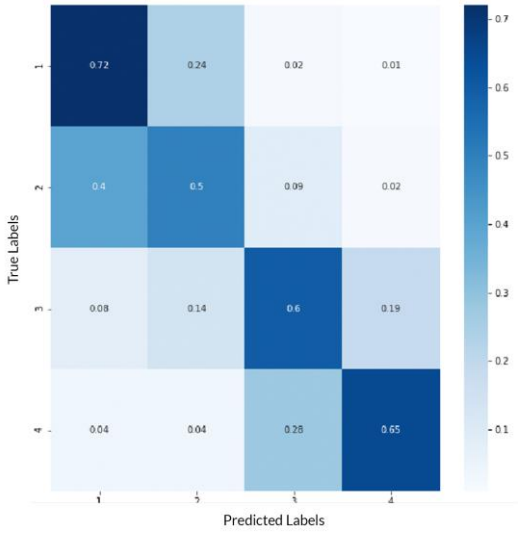

Figure 9: Confusion matrix for categorical classification of the afflicted images. In this trial, our VGG model struggled overall to classify these images correctly, however, it was able to discern between classes 1 and 2, and classes 3 and 4, and only struggled on the minor differences.

5.3. Afflicted Binary Classification

Following our results from categorical classification on the afflicted images, we saw that we were able to discern between the image either being in class 1 or 2, and the image being in class 3 or 4 well, however, we struggled to differentiate between 1 and 2, and between 3 and 4. Therefore, we will break these all into separate binary classification subproblems.

5.3.1 Early Stage DR vs. Late Stage DR

For this trial, we will group classes 1 and 2 as label 0 (early stage DR), and classes 3 and 4 as label 1 (late stage DR). With the same VGG network, preprocessed images, and data augmentation, we were able to train a model to training and validation accuracies of 98.1% and 80.0%, respectively. The training of this model can be seen in Figure 10 and the respective confusion matrix can be seen in Figure 11. For this binary sub-classification, while accuracy is 80%, precision falls to 61.1%, recall falls to 66%, and the F1 score is 63.5%.

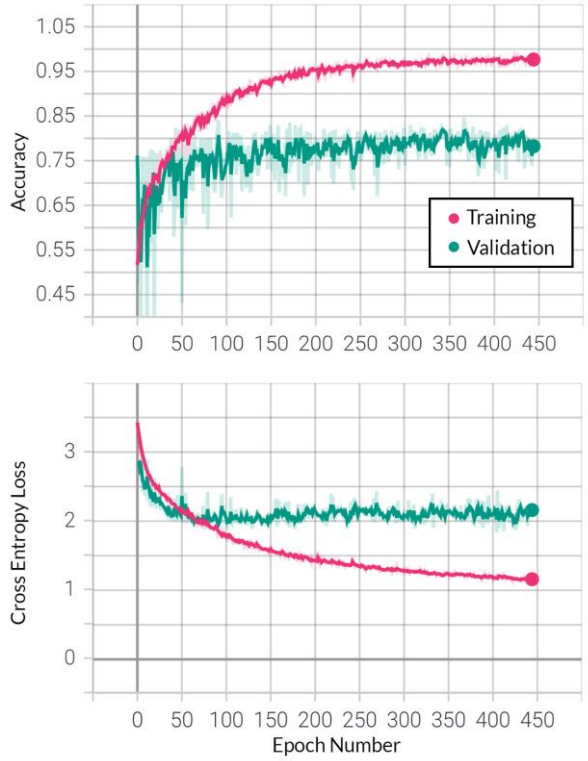

Figure 10: Accuracy and loss vs epoch for binary classification between early stage and late stage DR groups.

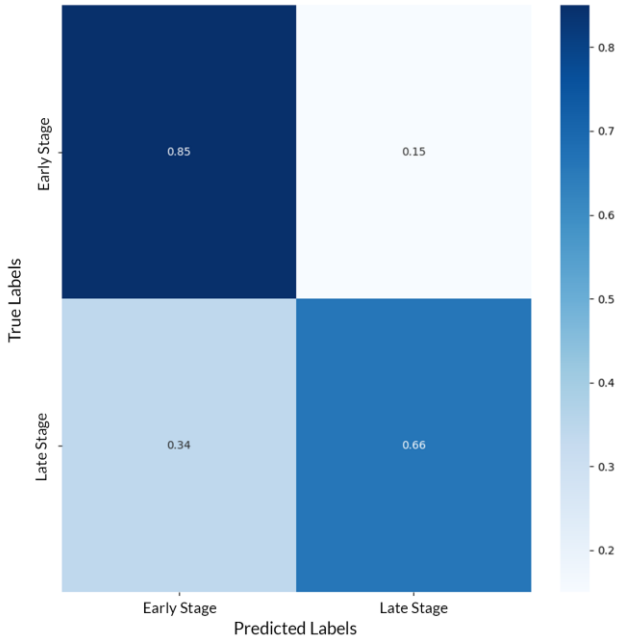

Figure 11: Confusion matrix for binary classification between early stage and late stage DR groups.

This binary classification trial is very important. We have already determined that the image contains signs of DR, however, this step will now classify the image between early stage DR and late stage DR. Our results show that we can classify the majority of images correctly but struggle on around 20% of the images. This is a preliminary result, and we expect that improvements could be made with further tuning of the model.

5.3.2 Early Stage DR Subclassification

For this trial, we will be directly comparing classes 1 and 2 (early stage DR) to increase the accuracy of our full categorical classification of DR. From the previous result, we assume that we know that we either have early stage, or late stage DR. Using the same model training setup as before, we achieved training and validation accuracies of 96% and 82%, respectively. The model training and confusion matrices can be seen in Figure 12 and Figure 13, respectively.

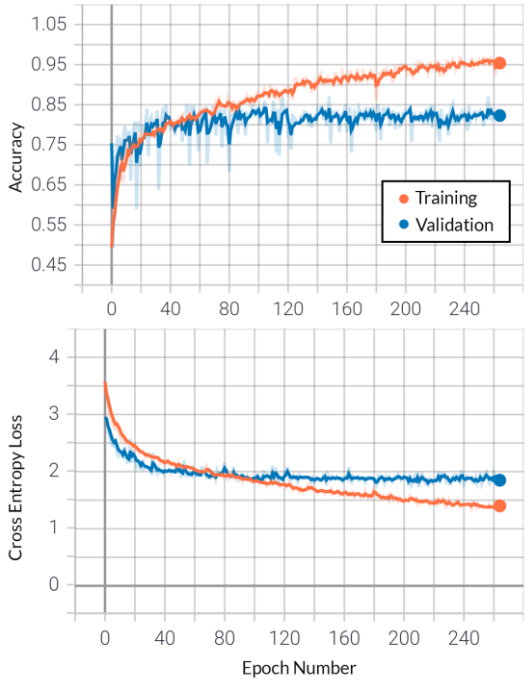

Figure 12: Accuracy and loss vs epoch for binary classification between group 1 and group 2.

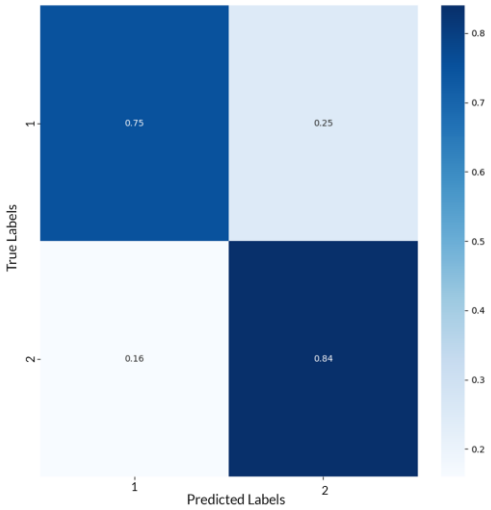

Figure 13: Confusion matrix for binary classification between group 1 and group 2.

5.3.3 Late Stage DR Subclassification

For this trial, we will be repeating 5.3.2, however, comparing classes 3 and 4 (late stage DR. From 5.3.1, we assume that we know that we either have early stage, or late stage DR. Once again, using the same model training setup as before, we achieved training and validation accuracies of 97% and 57%, respectively. The model training and confusion matrices can be seen below in Figure 14 and Figure 15, respectively.

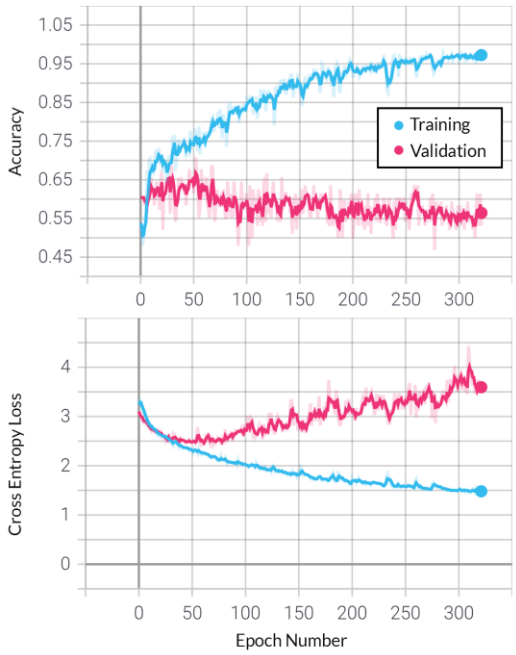

Figure 14: Accuracy and loss vs epoch for binary classification between group 3 and group 4.

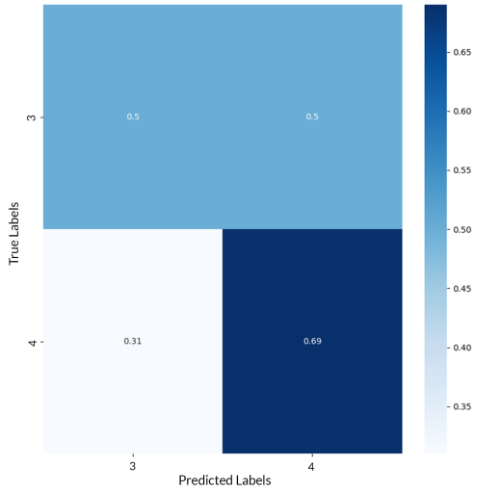

Figure 15: Confusion matrix for binary classification between group 3 and group 4.

6. Conclusion

The long-term negative effects of Diabetic Retinopathy (DR) can be effectively managed and often prevented if presence of the disease is detected early. In this paper we applied several existing image classification CNN architectures to the detection of DR and evaluated their capability, the results of which led us to attempt various binary classifications. We also incorporate image preprocessing techniques to demonstrate their effectiveness in aiding the CNN's classification ability.

During our survey of current CNN architectures, we struggled with issues such as overfitting and biasing to the 0 label. Our best result for 0-4 classification came from using the VGG16 network, with data augmentation and L2 regularization, trained on the Gaussian filtered Kaggle dataset. We achieved overall training and validation accuracies of 98.1% and 80.0% respectively for categorical classification. By looking at the confusion matrix of this result, we discovered that our model was able to correctly predict 96% of healthy images, with errors from the separation between the varying levels of DR.

This discovery led us to approach this categorical classification problem through multiple binary classifications. This allowed us to achieve training and validation accuracies of 99.4% and 98% respectively for identifying whether a person has some level DR, that is classifying between 0 and 1-4. We further identify that this binary classification has 99% precision, 96% recall, and a 97.5% F1 score. Due to these promising results with binary classification, we attempted to set up three more binary classifications: early stage versus late stage DR, label 1 versus label 2 (a subset of early stage DR), and label 3 versus label 4 (a subset of late stage DR). Unfortunately, our classification accuracies for each level of DR were not significantly changed from our original one-shot categorical classification of all labels (0-4).

From these preliminary results, we confirm that it is possible to use binary classification to accurately detect the presence of DR in images. Furthermore, we conclude that further dividing the classes into sub-classification problems did not lead to noticeable performance gains. Due to the time constraints of this project, we were unable to spend much time tuning these binary classification models and believe that further tuning may lead to performance improvements. To determine the true performance gain of adding these subclassification steps, we would need to run cross-validation studies on the various combinations.

Through these difficulties classifying various images in the datasets, we have learned some things about DR. Our CNNs can detect with a high accuracy whether DR is present, however struggle to differentiate between the various level of DR (seen in Figure 5). This leads us to believe that the presence of DR is not as subtle as the variation between levels. Furthermore, in Figure 8, we see that we generally can tell the difference between early stage DR and late stage DR, but struggle discerning between neighboring levels of DR. For further study of classifying DR, we believe that setting up a network to look for minor differences in features or the number of features may perform better than these CNNs like VGG16 that are able to classify thousands of vastly different images.

7. Individual Contributions

Over the course of this project, each team member contributed in a timely manner for whatever tasks needed to get done at that time. Given that project consisted of many different datasets and information, everyone did a little of everything. Predominately, these tasks consisted of research, testing, model improvement, writing, and generating figures. Further specification is given below

7.1. Michael Dermksian

Contributed a large amount on the research and model improvement side of the project. Implemented augmentations for improved validation and different ways for data import. Created various codes for data sorting and preprocessing. Contributed largely on the presentation and final project writing. Contributed to testing and modeling.

7.2. Harsh Dhruva

Contributed largely to the investigation and testing of different well researched models. Expanded the number of well-known CNNs that we tried and researched by creating working models. Researched largely into different data set and possibilities to overcome overfitting within our models. Generated many models, contributed to presentation creation and final draft writing.

7.3. Michael Turski

Contributed largely to model testing, creation, and research for good practices on DR in CNN. Worked heavily on comparing between results between modeling, created code to supply confusion matrices for each trial, generating valuable data on each trial. Contributed to presentation and final draft writing

7.4. Eric Rasmussen

Contributed largely to data preprocessing and augmentation. Created codes for cropping and filtering datasets to attempt to increase validation accuracy. Contributed to DR research along with best ways to split the data in CNN. Contributed to presentation creation and final draft writing.

8. References

[1] W. L. Alyoubi, W. M. Shalash and M. F. Abulkhair, "Diabetic retinopathy detection through deep learning techniques: A review," *Informatics in Medicine Unlocked*, 2020.

[2] X. Wang, Y. Lu, Y. Wang and W.-B. Chen, "Diabetic Retinopathy Stage Classification Using Convolutional Neural Networks," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, Salt Lake City, 2018.

[3] S. Wan, Y. Liang and Y. Zhang, "Deep convolutional neural networks for diabetic retinopathy detection by image classification," *Computers & Electrical Engineering*, vol. 72, pp. 274-282, 2018.

[4] Mobeen-ur-Rehman, S. H. Khan, Z. Abbas and S. D. Rizvi, "Classification of Diabetic Retinopathy Images Based on Customised CNN Architecture," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Dubai, 2019.

[5] W. Zhang, J. Zhong, S. Yang, Z. Gao, J. Hu, Y. Chen and Z. Yi, "Automated identification and grading system of diabetic retinopathy using deep neural networks," *Knowledge-Based Systems*, vol. 175, pp. 12-25, 2019.

[6] Y. Yan, J. Gong and Y. Liu, "A Novel Deep Learning Method for Red Lesions Detection Using Hybrid Feature," in *2019 Chinese Control And Decision Conference (CCDC)*, Nanchang, 2019.

[7] Y. Wu, Y. Xia, Y. Song, Y. Zhang and W. Cai, "NFN + : A novel network followed network for retinal vessel segmentation," *Neural Networks*, vol. 126, pp. 153-162, 2020.

[8] E. Decencière, X. Zhang, G. Cazuguel, B. Lay, B. Cochener, C. Trone, P. Gain, R. Ordonez, P. Massin, A. Erginay, B. Charton and J.-C. Klein, "Feedback on a publicly distributed database: the Messidor database.," *Image Analysis & Stereology*, vol. 33, no. 3, pp. 231-234, 2014.

[9] M. D. Abràmoff, J. C. Folk, D. P. Han, J. D. Walker, D. F. Williams, S. R. Russell, P. Massin, B. Cochener, P. Gain, L. Tang and M. Lamard, "Automated Analysis of Retinal Images for Detection of Referable Diabetic Retinopathy," *JAMA Ophthalmol*, vol. 131, no. 3, pp. 351-357, 2013.

[10] J. Cuadros and G. Bresnick, "EyePACS: an adaptable telemedicine system for diabetic retinopathy screening," *Journal of diabetes science and technology*, vol. 3, no. 3, p. 509–516, 2009.

[11] S. R. Rath, "Diabetic Retinopathy 224x224 Gaussian Filtered," February 2020. [Online]. Available: https://www.kaggle.com/sovitrath/diabetic-retinopathy-224x224-gaussian-filtered. [Accessed November 2020].

[12] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[13] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016.

[14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2015.