A Dual-Phase Deep Learning Pipeline for Comprehensive Retinal Image Analysis

Mathew Kuruvilla

Abstract:

Using a publicly available dataset of retinal images, I have been working on creating a convolution neural network that can accurately predict disease conditions in retinal images. The project consists of two parts: simply identifying images between normal and abnormal conditions to raise flags on images that might require further analysis, and to accurately determine conditions in images using 45 different disease conditions that have been carefully annotated by experts. Used hand in hand, both tools could be used in conjunction to produce accurate predictions on possible disease conditions a retinal image could have.

Introduction:

Machine learning can be implemented to make analysis of patient records much more streamlined and efficient for doctors by doing a preliminary analysis of data received and providing an accurate prediction. Human involvement cannot be replaced in this manner as this will likely not reach 100% accuracy anytime soon, but it will make it so that there is a safety net in the event there is an anomaly that the human eye does not catch. It can warn the reviewer to have a more careful eye when perusing the data if the predictive model shows that the patient may have some deviation from the norm. Essentially, if we properly train a model to analyze medical records autonomously, we would be able to create an AI medical assistant.

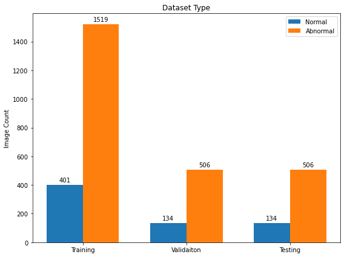
In our specific case we will be analyzing retinal images to predict if the patient will have any anomalies that could indicate a risk for diseases. Work has already been done in the field, but only for specific conditions such as diabetic retinopathy and age-related macular degeneration. The accuracy for these methods ranges from 80 to around 90% depending on the data type being used and how disease specific the model being used it (Jeong et al). Utilizing a robust enough database of annotated retinal images, models could be created that could produce much more descriptive and accurate information on disease risks present in retinal images that could be a large step in creating a set of tools that could be used in a clinical setting to wade through retinal image data.

There is a publicly available dataset of 3200 retinal images that have been expertly annotated for 45 different disease conditions that perfectly fit our needs (Pachade et al). Using this dataset, my goal for this project is to create two tools that would be used in tandem with each other. The first tool would be able to analyze retinal images and determine whether there are abnormalities present in an image. For a cursory sweep, this would provide scientists with a quick way to see what images need to be analyzed further for any underlying conditions in the patient. The second tool would be used with the first tool to present a list of possible conditions that could match up with the flagged retinal images. Essentially, this would function as a medical assistant to filter through large swathes of retinal data quickly with high accuracy.

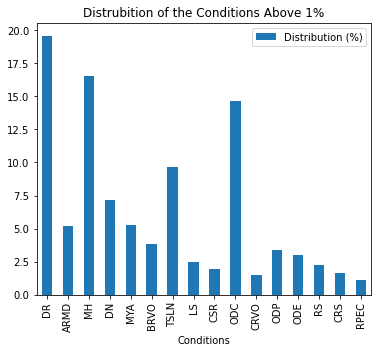
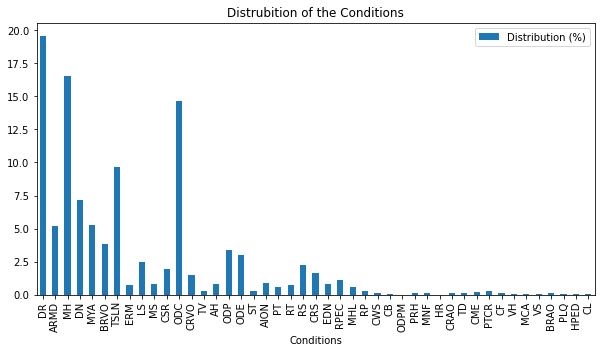
Methodology:

The two parts of this project will both be done using convolutional networks. For the first part of the project, I will be building my architecture to process the images with a high degree of accuracy to distinguish between normal and abnormal images. This will be the brunt of the experimentation for this project with adjusting hyperparameters to obtain the highest accuracy possible for these images. Nine hyperparameters were finetuned via a series of experiments, but the project’s true beginning is with the exploratory data analysis of the dataset and the preprocessing of the data it called for.

The 3200 images I was working with were split into three subsets: training, validation, and testing. These subsets had a 3:1:1 split for the data, but the robustness of the data had to be accounted for as well. As seen in Figure 1, there was a 1:3 ratio of normal images to abnormal images (Figure 3) in all the subsets meaning there was enough data to train the model properly. However, when it came to the second part of the project, there was a lack of data in some of the categories. Each condition has a binary set of results: either the image will have a risk for the condition, or it will not. In Figure 2A I take the lowest of the results (either positive or negative) in the training subset and look at it as a percentage over the whole. Most of the 45 categories do not have sufficient data for training, so I decided to cull any categories that did not provide at least a 1:99 ratio (1% of the smallest set of results). This ended up cutting down the number of categories from the formidable 45 disease categories to a much smaller 16 which can be seen in Figure 2B. However, this is still a lot more categories than have been considered by most other papers on the topic, so it works for my purposes as a proof of concept.

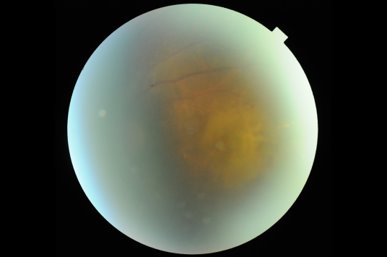
[[1]](#endnote-1)

**Figure 1**. The distribution of the image data subsets between normal and abnormal images. Showcases a 1:3 ratio in all subsets.



**Figure 2**. A. The percentage of the lowest attribute (positive or negative for each condition) present for each category. Showcases the lack of diverse data for many of the categories. B. The percentage of the lowest attribute (positive or negative for each condition) is present for only the chosen categories. Showcases that each category used for testing contained at least a 1:99 distribution of the data allowing for sufficient training data for each disease.

Once the initial exploratory data analysis was completed, then came the processing of the images directly. The images themselves were of decent quality, so no upscaling or refining of the images needed to be done. However, technological limitations limited how much data we could store at once. The project required that we store all the images as arrays as we were training the model and storing data for 3200 images each with approximately 1400 by 2400 pixels was a heavy burden on my computer in terms of available space. For that reason, and because the images weren't all perfectly the same number of pixels, the images were all resized to 200 by 200 pixels. Of course, this would result in some loss of data, but hopefully it wouldn’t be anything significant enough to cause issues.



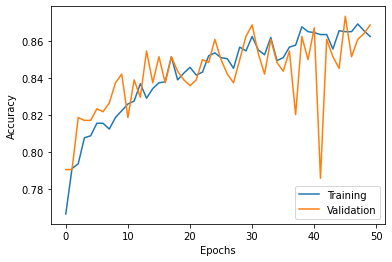
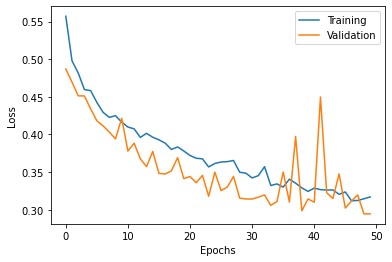
**Figure 3**. A. Normal retinal image. B. Abnormal retinal image (characterized by media haze and macular degeneration)

With the initial preprocessing out of the way, then came the actual experimentation. The nine hyperparameters tweaked included the following parts of the architecture: the number of epochs, the batch size used in each epoch, the number of convolutional filters, the size of the window in the convolutional layer, the size of the window for the max pooling filter, the activation functions used between each layer, the final activation function going into the output layer, and the optimizer used in conjunction with the loss function in the model. Rigorous testing was done with these parameters until the final model was chosen and the data was run against the test data to get the final accuracy results.

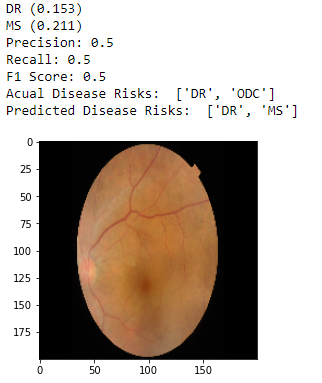
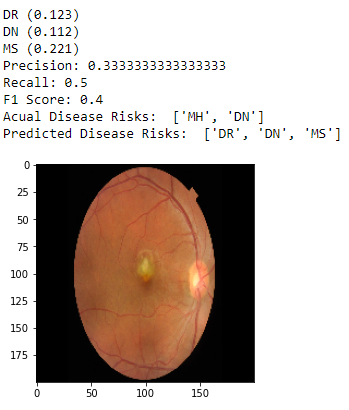
Once the first part of the project was completed, the second part of the project was underway. This part of the project was much more straightforward. The same architecture from part one would be used here for the image processing, but here instead of simple binary classification we’re doing multi-label classification, so the output layer had to be adjusted for the number of categories we were considering. Here, the main testing that needed to be done was to determine the ideal threshold to consider disease risk predictions for each of the images. Once that was completed, part two of the project was completed as well.

Results:

In part one, over the course of approximately 18 experiments with the convolutional neural network, I managed to determine the best hyperparameters for the network. I utilized the graphs output by each run of the model to determine how the parameters did. As can be seen in Figure 4, two graphs would be output after each run showcasing the loss over the epochs and the accuracy over the number of epochs the model was run for. Some things I looked for to make sure I was obtaining a good model was to make sure the accuracy and loss didn’t plateau throughout the process as that would indicate the model wasn’t learning as it should be. Another thing I needed to watch out for was validation accuracy and the training accuracy diverging too far from each other. The training accuracy would only continue to increase as the number of epochs increased, so to prevent overfitting I had to make sure to limit the number of epochs the model was run for. A short summary for each of the experiments run can be seen in Appendix Table 1. The final model I ended up using (model 18) produced an accuracy of 90.16% which was my stretch goal for this project. I wanted to be able to be in at least the 80 to 90% range but having an accuracy above 90% exceeded my expectations. The hyperparameters used for the chosen model were then used with part two of the project.

  
**Figure 4**. Both the outputs graphs from running a model (these graphs are specifically from the model 18 run) A. Loss v. Epochs graph B. Accuracy v. Epochs graph

In part two of the project the main experimentation done was to determine the correct cutoff for the predictions so that only the most likely disease risks for each image would be shown. Changing the cutoffs from 10% to 20% then back to 15%, I was able to get an F1score 0.43 using the validation data, and then when done with the testing data I was able to get an F1score of 0.57. Of course, these are not ideal F1scores, but part of the reason is because I included normal retinal conditions as well for an added challenge which likely tanked the F1 scores with false positives for normal retinal images. The output that a clinician would see when using this tool is shown in Figure 5.



**Figure 5**. A. Retinal image prediction with a cutoff of 0.1 B. Retinal image prediction with a cutoff of 0.15 (less false positives and a higher F1 score)

Conclusion:

Both parts of the project were a success with the creation of two tools that could be used in tandem to predict retinal disease risks just from scanning an image quickly. After rigorous finetuning of the hyperparameters, I managed to obtain an impressive 90.16% accuracy on the test data using my model to predict abnormalities in the retinal images. Using the success in part one, part two was run using the same architecture and the same hyperparameters with minor adjustments. The finetuning done here was the cutoff for predictions to be considered as a disease risk for an image, and that was settled at 15%. The F1scores for this section of the project were not ideal, but that was partly due to the added challenge of mixing in normal images to see if the tool would be able to handle it and possibly be used by itself. While the tool in the second part of the project showed that it would not be ideal to use it by itself, both tools used together could provide a high degree of prediction accuracy for retinal image data.

Citations:

1. Jeong Y, Hong YJ, Han JH. Review of Machine Learning Applications Using Retinal Fundus Images. Diagnostics (Basel). 2022;12(1):134. Published 2022 Jan 6. doi:10.3390/diagnostics12010134
2. Pachade S, Porwal P, Thulkar D, Kokare M, Deshmukh G, Sahasrabuddhe V, Giancardo L, Quellec G, Mériaudeau F. Retinal Fundus Multi-Disease Image Dataset (RFMiD): A Dataset for Multi-Disease Detection Research. Data. 2021; 6(2):14. <https://doi.org/10.3390/data6020014>

1. [↑](#endnote-ref-1)