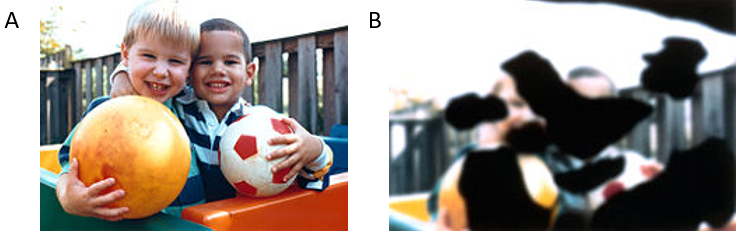
**Using Deep Learning to Detect Diabetic Retinopathy**

**Introduction**

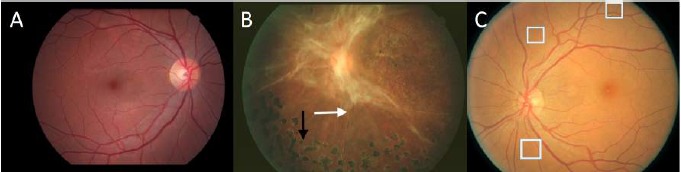
Diabetic retinopathy (DR) is the damage to retinal blood vessels that eventually leads to blindness if not treated. DR is a complication associated with diabetes and the longer someone has diabetes, the higher the risk of them getting DR. DR is a leading cause of vision-loss globally. Of an estimated 285 million people with diabetes mellitus worldwide, approximately one third have signs of DR and of these, a further one third of DR is vision-threatening DR. Initially, DR can cause no symptoms or only mild vision problems but can eventually lead to complete blindness. Figure 1 shows a representation of how DR affects vision. [1]



**Figure 1:** (A) Normal vision (B) Vision affected by severe DR

Early detection of DR, critical for good prognosis, relies on professionals and can be labor and time-intensive [2]. This puts certain demographics at a severe disadvantage. DR screening methods are also dependent on the subjective reading of fundus images, which leads to inconsistency in diagnosis [2]. Thus, automatic computer based techniques to diagnose DR are critical in the modern world.

**Previous literature**

Current diagnosis of DR from fundus images depends on detecting micro-aneurysms, small saccular bulges of capillaries, retinal hemorrhages, and ruptured blood vessels. On the other hand, severe DR fundus images show large scale features like cotton wool and flame-shaped hemorrhages (Figure 2) [3].

**Figure 2:** (A) Normal fundus photograph (B) Severe DR with white arrows pointing towards flame shaped hemorrhages (C) Early stage DR with white boxes showing micro-aneurysms

Generally speaking, previous literature models have been able to perform binary classification of diabetic retinopathy at accuracies upwards of 90% [3]. However, previous methods have not shown impressive results in the graded classification of DR, especially in terms with early diagnosis. The early diagnosis is extremely difficult because it involves detecting micro-aneurysms, which can also prove to be difficult for a trained professional (Figure 2). The first neural network that tried to differentiate between normal patches of retina from patches with micro-aneurysms achieved an accuracy of 74% [4]. Some recent studies that classify early and mild DR report accuracies between 0% and 41% [5, 6].

**Study goals**

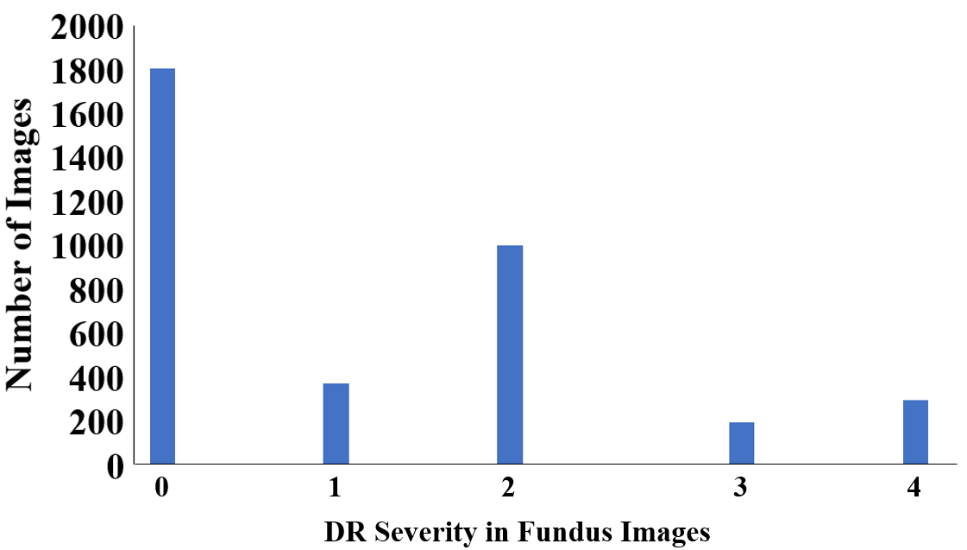
The primary goal of this study was to develop a single convolution network that would be able to perform (1) binary classification (2) early detection (3) severity detection in DR fundus images. Performing binary classification would allow comparison to previous benchmark models in literature. The early detection and severity detection models will be compared to previous literature models in terms of model architecture and model accuracy.

**Dataset**

The APTOS 2019 blindness dataset was used to train all of the models in this study [7]. A clinician has rated each image for the severity of diabetic retinopathy on a scale of 0 to 4:

* 0: No DR
* 1: Mild DR
* 2: Moderate DR
* 3: Severe DR
* 4: Proliferative DR

The dataset contains approximately 3,680 images with the following distribution of labels (Figure 3). Both datasets consist of color photographs that vary in height and width between the high hundreds to couple thousands of pixels.



**Figure 3:** Distribution of image classification in APTOS blindness dataset

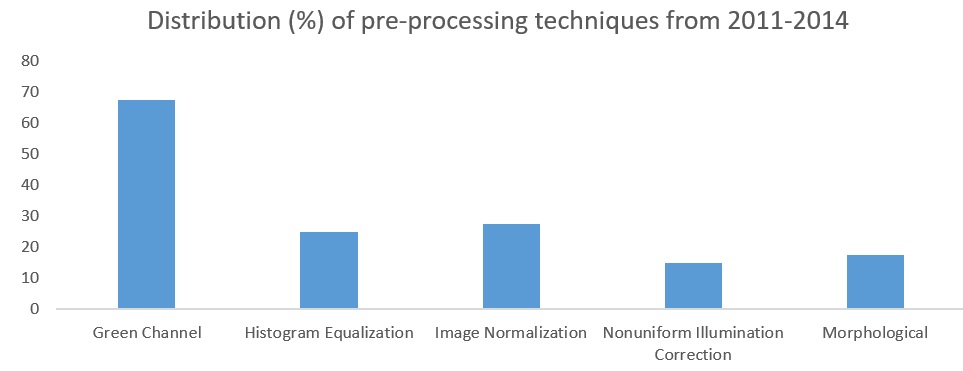
**Method**

**Dataset Modifications**

The APTOS dataset was used to run three experiments:

1. Binary classification
2. Early detection
3. Severity classification

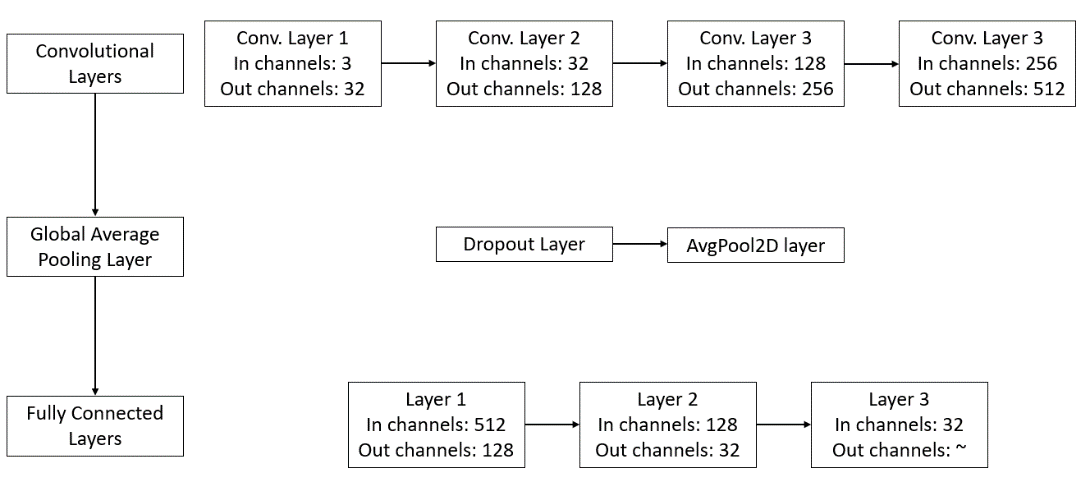
For all models, images were resized to 256x256 pixels and normalized. All three color channels were used for model training, even though previous studies have largely focused on using the green channel (Figure 4) [8]. For the first model (binary classifier), the image labels were changed to “0” if the image had no DR, or “1” otherwise. For the second model (early detector), images with no DR were labelled “0,” “1” if they had early DR (corresponding to “1” and “2” from original dataset), or “2” if they had severe DR (corresponding to “3” and “4” from original dataset). For the final model (severity classifier), the original labels were maintained.



**Figure 4:** Distribution of preprocessing techniques used from 2011-2014 [8]

**Model Architecture**

The single model architecture used for all three experiments is described in Figure 5. The model included four convolutional layers, a global average pooling layer, and three fully connected layers. The final layer had different number of nodes for each experiment (shown by ‘~’ in Figure 5).



**Figure 5:** General model architecture used for all three experiments. ‘~’ in the final fully connected layer signifies that the final number of nodes changed depending on the experiment

As mentioned before the input image size to the first convolutional layer was 256x256 for each experiment. Each convolution layer also included a kernel of size 5x5, stride of size 1, and padding of size 2. After each convolutional layer, a ReLU activation was applied. Finally, a max pool of kernel size 2 and stride of size 2 was used to extract the most important features from each convolutional layer. The model was converted to cuda in order to use the system GPU. Adam optimizer was utilized. Total number of trainable parameters are 4,271,586. . Cross entropy loss criterion was used in all of the experiments.

**Determining hyperparameters**

Multiple batch sizes and learning rates were tried on the dataset for binary classification and the validation accuracy was tracked (Table 1).

**Table 1:** Varying hyperparameters and seeing the effect on validation accuracy

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of train images** | **Number of test images** | **Epochs** | **Batch size** | **Learning rate** | **Validation accuracy** |
| 256 | 256 | 10 | 32 | 0.0001 | 88% |
| 256 | 256 | 30 | 32 | 0.0001 | 90% |
| 256 | 256 | 30 | 32 | 0.0001 | 94% |
| 256 | 256 | 30 | 16 | 0.00001 | 68% |
| 256 | 256 | 30 | 16 | 0.001 | 91% |
| 256 | 256 | 30 | 8 | 0.0001 | 93% |

After these experiments, the batch size for all the experiments was set to 8 and learning rate to 0.0001. The binary classification experiment ran for 25 epochs, but the early detection and severity classification experiments ran for 40 epochs each.

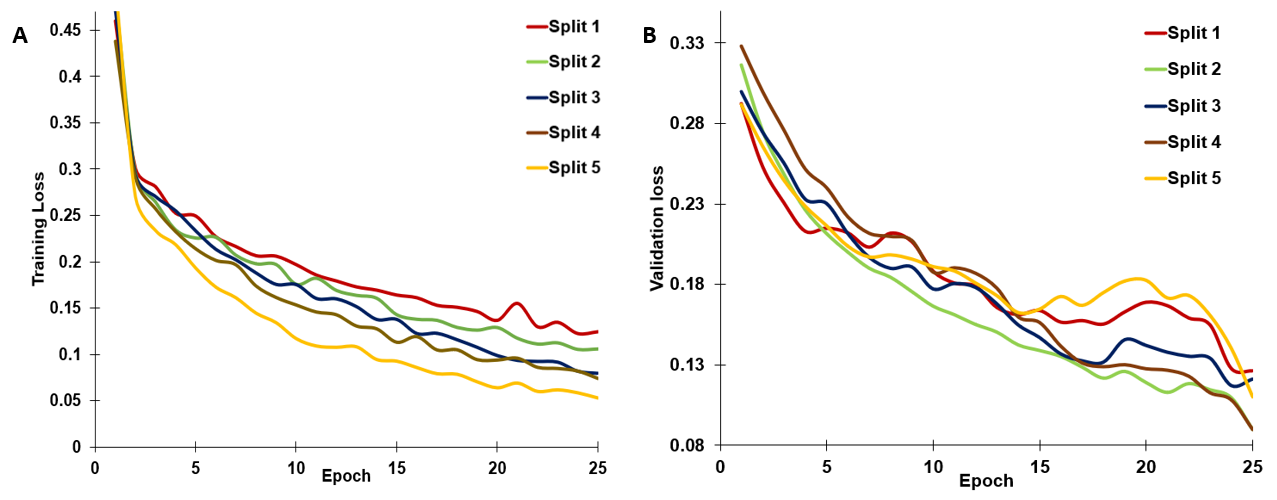
**Training and testing protocol**

A five fold cross validation study was performed for each experiment. Four splits were used for model training and the final split was used for model validation. After each epoch of training, the model was immediately validated. Training loss, training accuracy, validation loss, and validation accuracy were tracked for all epochs. The actual and predicted labels were saved. For binary classification, the true positive and false positive rates were calculated for each split. For the early detection and severity classifying experiments, confusion matrices were created for each split.

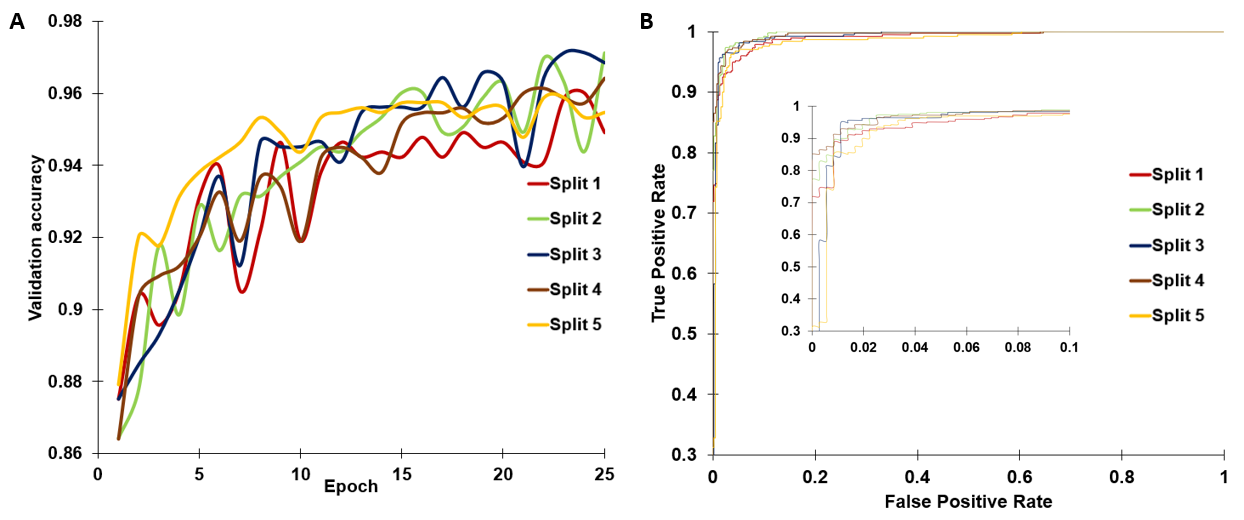
**Results**

**Binary Classifier**

Figure 6A shows that the training loss decreases approximately 89% over 25 epochs. While the training loss decreased, the validation loss also decreased (Figure 6B) for all of the splits. After epoch 15, however, there seems a slight increase in validation loss. Since training loss is decreasing past epoch 15, but validation loss shows small increase, the model might be overfitting. In the future, model training would be stopped if training loss decreases, but validation loss increases.

**Figure 6:** (A)Training loss (B) Validation loss for model for all five splits over 25 epochs for the binary classifier experiment

The binary classifier model gave very strong validation accuracies (Figure 7A). The ROC curves for all the splits (Figure 7B) also suggest the very robust results of the binary classifier. Table 2 classifies the split-wise validation accuracy and area under the curve.

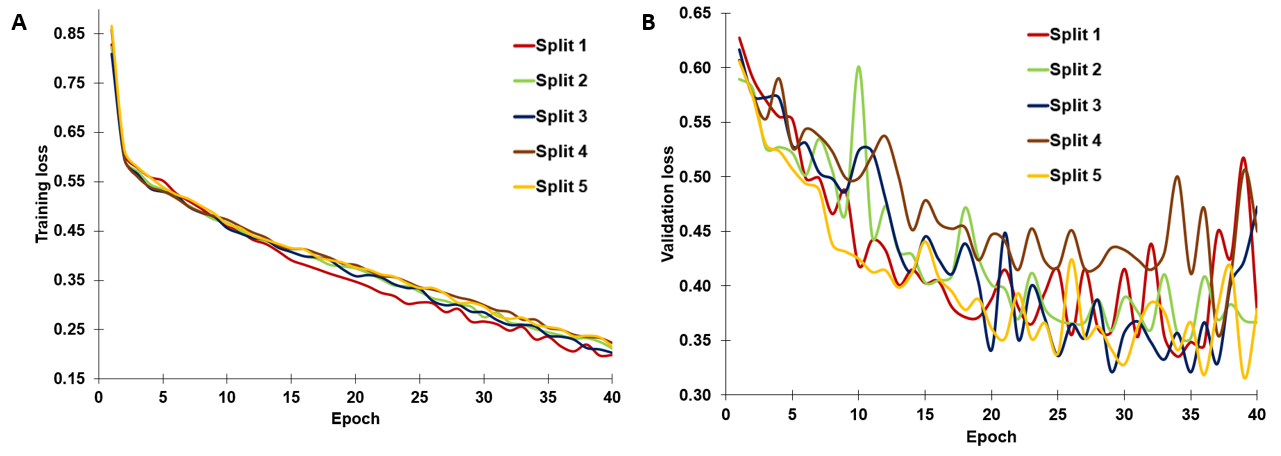
**Figure 7:** (A) Validation accuracy for all five splits for binary classifier (B) ROC curves for all for splits

**Table 2:** Accuracies and area under curve for each split

|  |  |  |
| --- | --- | --- |
| **Split** | **Validation Accuracy** | **Area Under Curve** |
| 1 | 0.949 | 0.990 |
| 2 | 0.971 | 0.996 |
| 3 | 0.968 | 0.992 |
| 4 | 0.964 | 0.994 |
| 5 | 0.954 | 0.985 |
| **Average** | 0.961 | 0.992 |

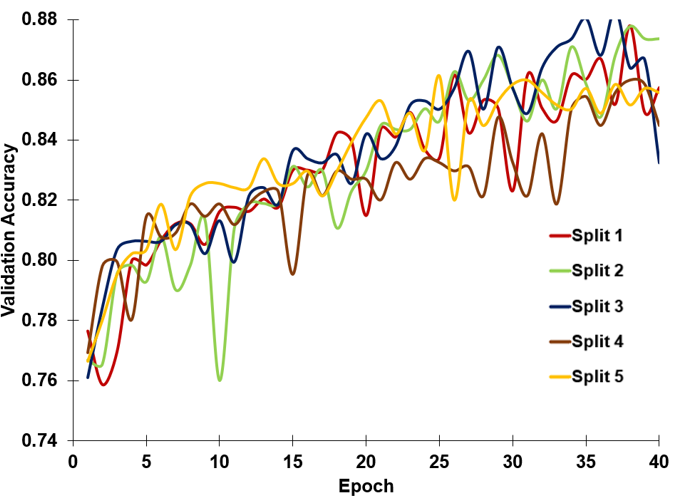
**Early Detector**

Figure 8A shows that the training loss decreases approximately 76% over 40 epochs. While the training loss decreased, the validation loss also decreased (Figure 8B) for all of the splits. Again, the model seems to be overfitting after epoch 30. Thus, the number of epochs need to be reduced.



**Figure 8:** (A)Training loss (B) Validation loss for model for all five splits over 40 epochs for the early detector experiment

The early detector gave moderately good accuracies over validation (Table 3, Figure 9). These results are comparable with literature models [3]. The validation accuracy seems to level off after epoch 30, suggesting overfitting once again.

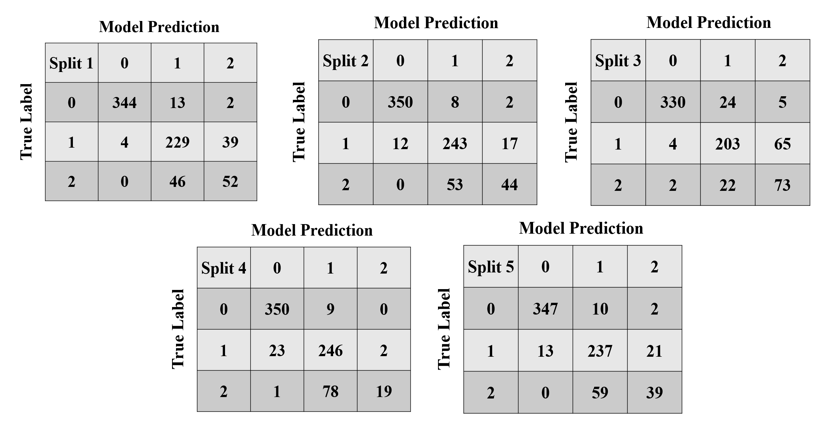


|  |  |
| --- | --- |
| Split | Validation Accuracy |
| 1 | 0.86 |
| 2 | 0.87 |
| 3 | 0.83 |
| 4 | 0.84 |
| 5 | 0.86 |
| Average | 0.852 |

**Table 3:** Validation Accuracy for early detector

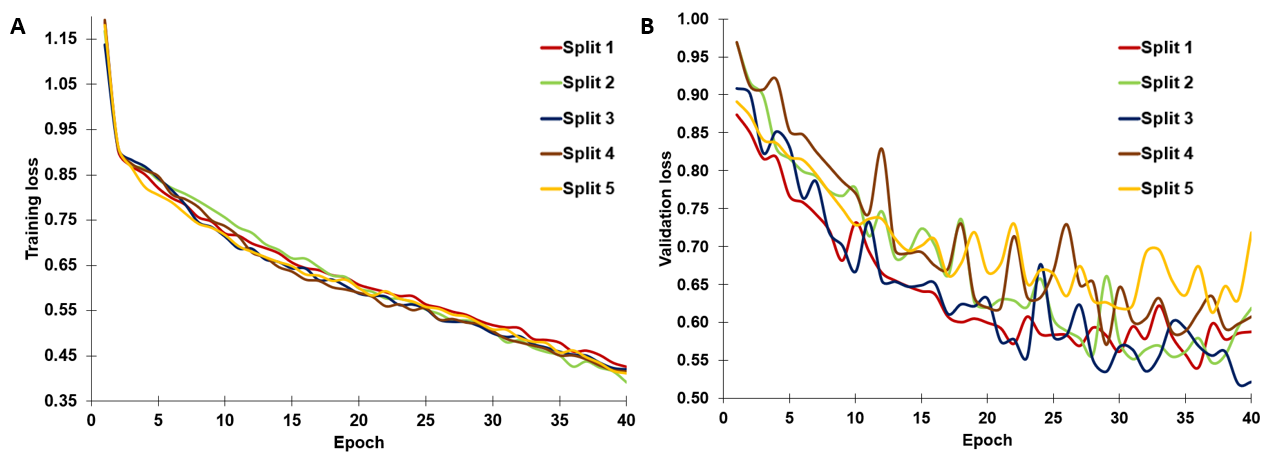
**Figure 9:** Validation accuracy for early detector for all five splits

Figure 10 shows the confusion matrices for all five splits. We see that the model can accurately separate healthy subjects from mild DR patients. However, the model has trouble predicting severe and proliferative DR (Figure 10).

**Figure 10:** Confusion matrix for all splits of early detector

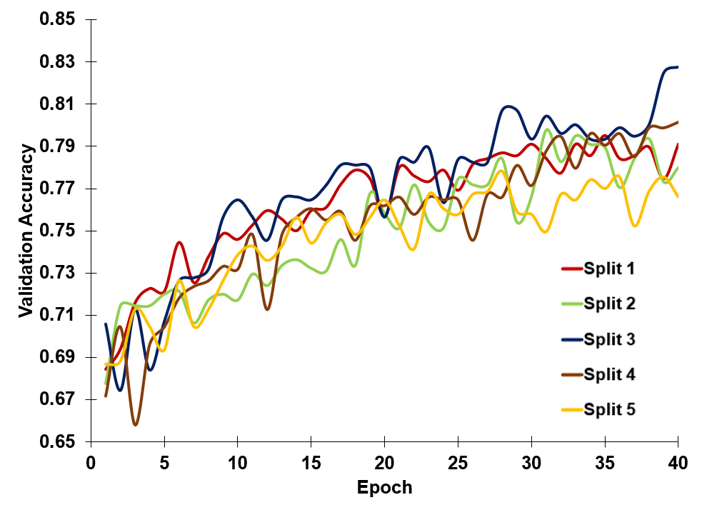
**Severity Classifier**

Figure 11A shows that the training loss decreases approximately 76% over 40 epochs. While the training loss decreased, the validation loss also decreased (Figure 8B) for all of the splits. Again, the model seems to be overfitting after epoch 30. Thus, the number of epochs need to be reduced.



**Figure 11:** (A)Training loss (B) Validation loss for model for all five splits over 40 epochs for the severity classification

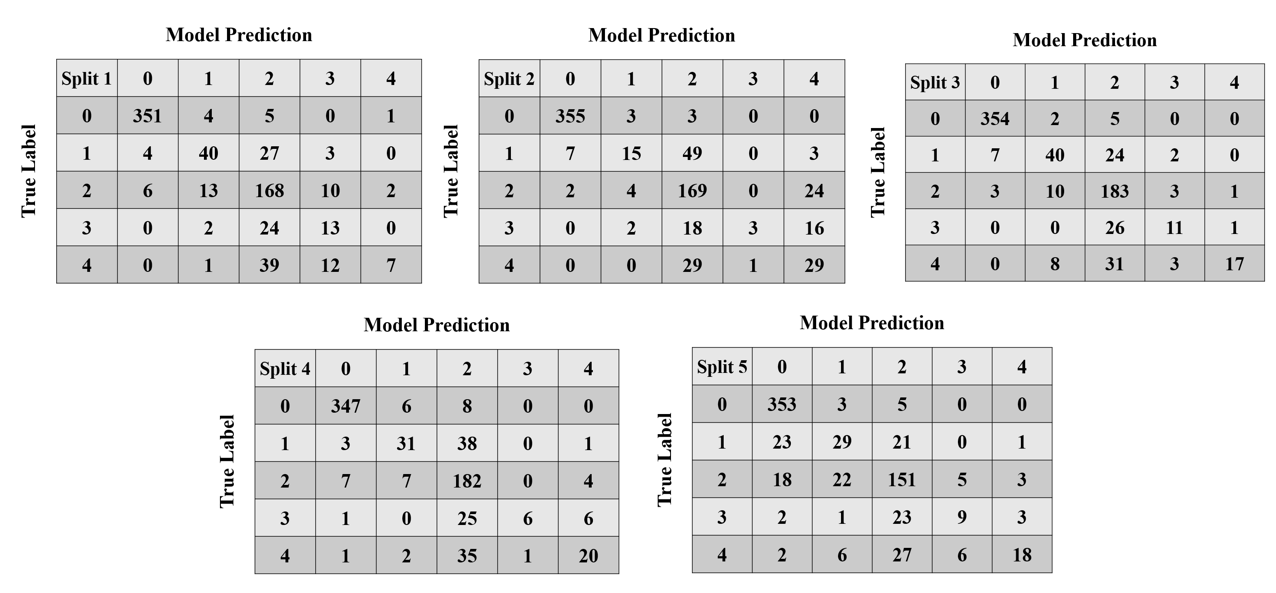
The early detector gave moderately good accuracies over validation (Table 4, Figure 12). This could happen because literature models are much more complex, with residual networks and intricate algorithms like K-Nearest Neighbor.



|  |  |
| --- | --- |
| Split | Validation Accuracy |
| 1 | 0.79 |
| 2 | 0.78 |
| 3 | 0.83 |
| 4 | 0.80 |
| 5 | 0.77 |
| Average | 0.794 |

**Table 3:** Validation Accuracy for early detector

**Figure 12:** Validation accuracy for severity classification for all five splits

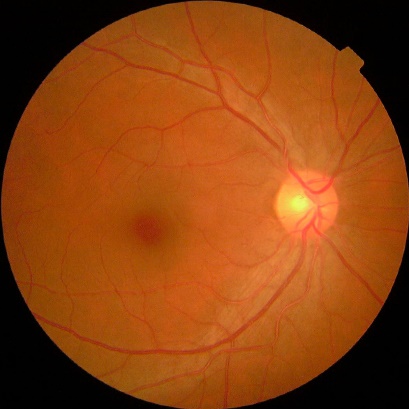
Figure 13 shows the confusion matrices for all five splits. We see that the model can accurately separate healthy subjects from mild DR patients. However, the model has trouble predicting severe and proliferative DR (Figure 13).

**Figure 13:** Confusion matrix for all splits of severity classifier

**Discussion**

The binary classifier model gave very strong results. The average area under the curve score is higher than the state-of-the-art literature models [5]. The binary classifier’s sensitivity was calculated as 0.932 and the specificity as 0.989, which are comparable to literature models [3, 5]. The specificity being higher than sensitivity means that the model is better at predicting healthier cases than DR cases. On the other hand, the accuracies presented by the early detector and severity classifier are higher than those presented by [3]. However, more recent literature models as well as the leaderboard on Kaggle for the APTOS challenge present clinical level of accuracy for the graded classification of DR [12]. Nevertheless, these models much more complex than the ones presented here. Another reason could be the size of the dataset. The higher accuracy models [3] have been trained on private datasets with over 120,000 images, which is extremely large compared to the current dataset.

Figure 14 shows a case where the binary classifier wrongly predicted a fundus image to have DR. It is not apparent why this prediction was made and creating a heatmap that shows the areas of DR detected by the algorithm (like in [3]) could help determine why this wrong classification was done. However, it is likely that the model could not learn to differentiate between the micro-scale features of mild DR that distinguish it from no DR. Figure 15 shows a fundus image that the binary classifier predicted no DR, but there is DR present in the image. This is possible because of data set fidelity. The fundus images are obtained from all over the world and from various instruments. Even though the pixels of the images were normalized, this approach may not be enough. Some previous studies [3, 11] have found that contrast limited adaptive histogram equalization protocols can help enhance the DR features [10]. This is a preprocessing approach that can be further explored.



**Figure 14:** An example of false positive prediction by binary classifier



**Figure 15:** An example of false negative prediction by binary classifier

Taking a look at the model architecture can also possibly explain why the early detector and severity classifier experiments did not yield very high accuracies. From Figures 10 and 13, we see that the model is able to identity healthy subjects very accurately but cannot classify the more severe cases. This may happen because deeper in the model, the aperture of the convolutional layers is larger than the minute features distinguishing early onset DR from severe DR. This problem can be potentially solved by using residual networks in which channels from initial convolutional layers are fed to deeper layers in the model to better predict the early onset of DR and severity of DR.

Looking at the distribution of the labels in the current dataset could also explain the low accuracy achieved in early onset and severity classifier experiments. Majority of images are of the label “0” followed by “2”. Thus, Figures 10 and 13 show that these labels are very accurately predicted. The other labels do not have a lot of images compared to labels “0” and “2” (Figure 3), and therefore the model is not very well trained on predicting these labels. This problem could be solved by moving to a larger dataset, like [11] or employed data augmentation techniques.

**Future work**

The first change I would like to make is move to the larger Kaggle dataset. This would ensure that there is a more equal distribution of images for the labels. Secondly, I would like to explore techniques to augment the data. Once these changes are made, I am interested in looking into implementing residual networks and dense nets to better predict DR. Finally, I would like to make two changes to the training-testing protocol. Firstly, I want to decide on number of epochs more objectively. This can be done by terminating training when validation loss and training loss start to diverge. This solution can also help with the overfitting problem evident in all three of the experiments. Secondly, instead of doing a cross validation study, I would like to move to a 3-1-1 approach, where three folds are used for training, one fold is used for testing, and the final fold for validation. This would make our experiments more comparable with literature results.

**References**

[1] Lee, Ryan, Tien Y. Wong, and Charumathi Sabanayagam. "Epidemiology of diabetic retinopathy, diabetic macular edema and related vision loss." Eye and vision 2.1 (2015): 17.

[2] Goh J. K. H, Cheung C. Y., Sim S. S., Tan P. C., Tan G. S. W, Wong T. Y. Retinal imaging techniques for diabetic retinopathy screening. Journal of diabetes science and technology, 2016;10(2):282–294.

[3] Lam, Carson, et al. "Automated detection of diabetic retinopathy using deep learning." AMIA Summits on Translational Science Proceedings 2018 (2018): 147.

[4] Gardner G., Keating D., Williamson T., Elliott A. Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool. British journal of Ophthalmology. 1996;80(11):940–944.

[5] Gargeya R, Leng T. Automated identification of diabetic retinopathy using deep learning. Elsevier. 2017

[6] Gulshan V., Peng L., Coram M., Stumpe M. C., Wu D., Narayanaswamy A., Venugopalan S., Widner K., Madams T., Cuadros J., et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA, 2016;316(22):2402–2410.

[7] <https://www.kaggle.com/c/aptos2019-blindness-detection/data>

[8] Ahmad, Arslan, et al. "Image processing and classification in diabetic retinopathy: A review." 2014 5th European Workshop on Visual Information Processing (EUVIP). IEEE, 2014.

[9] Padhy, Srikanta Kumar, et al. "Artificial intelligence in diabetic retinopathy: A natural step to the future." *Indian journal of ophthalmology* 67.7 (2019): 1004.

[10] Shome, Saikat Kumar, and Siva Ram Krishna Vadali. "Enhancement of diabetic retinopathy imagery using contrast limited adaptive histogram equalization." *International Journal of Computer Science and Information Technologies* 2.6 (2011): 2694-2699.

[11] <https://www.kaggle.com/c/diabetic-retinopathy-detection>