Multi-Class Feature Recognition Model for Retinal Disease Classification from Retinal Scan Images (Computer Vision)

Name: Nicolas Friley
SUNet ID: nfriley
nfriley@stanford.edu

Name: Xin-Yi Pan
SUNet ID: xinyipan
xinyipan@stanford.edu

1 Motivation & Problem Statement

According to the World Health Organization [1], there is a general shift towards population ageing particularly in high-income countries. Common health conditions associated with ageing include diabetes, cardiovascular diseases, neuro-degenerative diseases, and stroke, which can be detected by a retinal scan [2]. Early detection often offers the best chance of cure [3], increasing the quality of life and longevity of the population. Current methods deploy computer-aided diagnosis systems to help doctors make quick diagnosis, or employ Deep Learning, particularly the Convolutional Neural Networks (CNN) and their variants, including Residual Networks (ResNet) [4][5]. However, these methods are often limited to the detection of single disease.

Our project has 2 main focus. First, we aim at training a feature recognition model to recognize multiple classes of health issues detectable by a retinal scan which is not commonly done. This could increase the efficiency and accuracy of diagnosis, and allow for a more affordable alternative to current health tests, encouraging regular health monitoring. Second, we explore a newer model for retinal image classification – Vision Transformer, and compare it with the performance of state of the art ResNet model which we use as our baseline model.

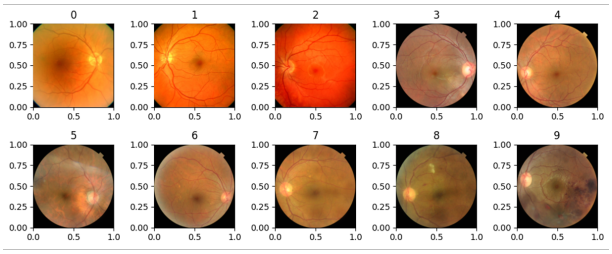


Figure 1: Example of retinal scans

2 Description of Data

The dataset used in our final report is an open-source dataset from Mendeley Data [6]. With the goal of performing multi-class classification, we wanted a dataset with both the highest number of classes and sufficient number of data within each class. Our initial plan of combining datasets from multiple sources was made complicated by several factors. First, most datasets only have 1-2 overlapping diseases. Second, datasets with more classes contain very small number samples diseases not found in other datasets. This could lead to class imbalance. Third, there is inconsistency in the labelling of datasets where some sources has one disease to an image while others allowed for multiple diseases to be associated to a single image. This made it challenging to achieve consistency across datasets. Thus, we chose a dataset with 20 medical conditions, with RGB image files of varying resolutions ranging from 744x575 to 1430x1393.

2.1 Data Augmentation

Figure 2: Types of image transformations used for data augmentation

To increase the total number of examples to approximately 10000, we implemented data augmentation on our dataset before using it to train our models [7] [8]. During the initial training of our baseline model, we recognised the detrimental impact of

unequal distribution of data across classes in our model's performance. To limit the extent of possible bias in the distribution of classes and achieve relatively equal distribution across classes, we implemented data augmentation on all classes, and performed more data augmentation on classes with fewer examples. Particularly, we augmented our dataset by flipping our images vertically, horizontally and diagonally (i.e. horizontal then vertical flip), and rotating them in intervals of 90 degrees. This was decided with consideration of what a realistic retinal image would look like. The type of image transformation we performed is illustrated in figure 2.

	DR	N	MH	ODC	TSLN	ARMD	DN	MYA	BRVO	ODP	CRVO	CNV	RS	ODE	LS	CSR	HTR	ASR	CRS	O
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Bef	495	493	157	202	66	125	126	44	64	29	54	14	56	56	2	32	19	6	20	148
Aft	495	493	529	555	529	501	490	529	513	465	433	393	449	449	481	513	533	572	546	593

Figure 3: Distribution of data before and after augmentation.

The distribution of the original and augmented data are shown in figure 3, where 'Bef' and 'Aft' refer to the dataset before and after data augmentation. The acronyms in the header correspond to the classes 'Diabetic Retinopathy', 'Normal Retina', 'Media Haze', Optic Disc Cupping', Tessellation', 'Age-Related Macular Degeneration', 'Drusen', 'Myopia', 'Branch Retinal Vein Occlusion', 'Optic Disc Pallor', 'Central Retina Vein Oclussion', 'Choroidal Neovascularization', 'Retinities', 'Optic Disc Edma', 'Laser Scars', 'Central Serous Reinopathy', 'Hypertensive Retinopathy', 'Arteriosclerotic Retinopathy', 'Chorioretinitis' and 'Other Diseases', respectively. Each of those class label is mapped to an integer and subsequently converted to one-hot embeddings in our pre-processing step. Our augmented data has a mean of 503 images per class with a standard deviation of ±49 images.

Our dataset is made up of various sources of data combined. To achieve consistency in the size of images across datasets, we resized all images to a 256 x 256 pixel resolution. This size is chosen based on prior work with the same dataset which showed a general performance improvement for smaller image sizes when sizes between 384 to 700 pixels were explored [6]. Non-square images were cropped based on the length of the shortest size so as to center the retinal scan in the resulting square image. The images were resized using OpenCv's INTER_AREA, where a shrinkage used 'true area' method. This was chosen to minimize the extent of image distortion in the resizing process. To help our model learn meaningful features from the dataset, the mean RGB value was computed over the training set and subsequently subtracted from each pixel. This ensures the image arrays fed into our algorithm are consistent, with pixel values centered around zero [9].

3 Hyper-parameter and architecture choices

3.1 Baseline Model - ResNet V.2

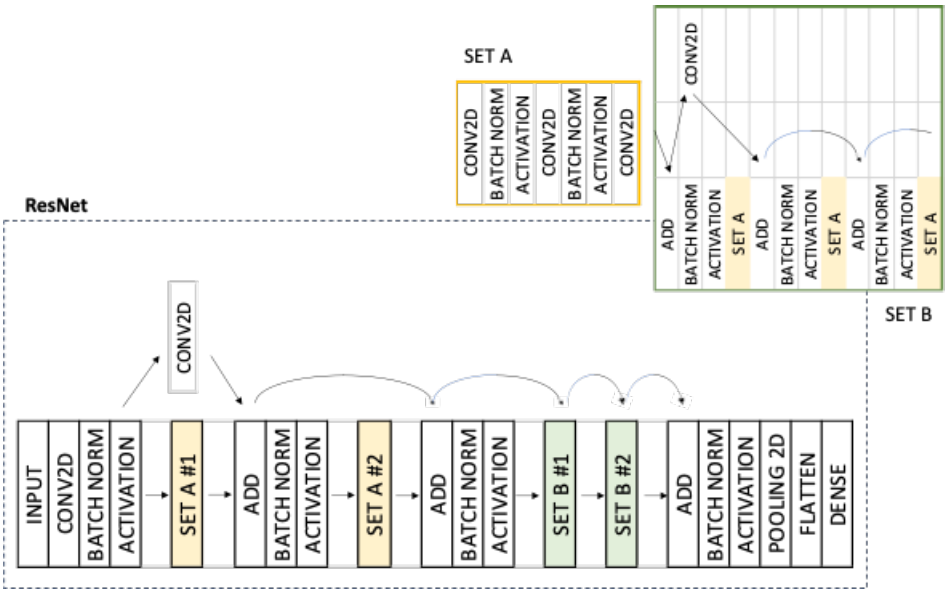


Figure 4: Schematic of ResNet Model

We chose to use ResNet as our baseline model as it is commonly used for deep learning applications and are known to avoid the problem of 'Vanishing Gradient' associated with the classic Convolutional Neural Networks (CNN) [10][11]. In particular, we selected the ResNet V.2 as proposed in 2016 He's "Identity Mappings in Deep Residual Networks" [12]. This choice was motivated by the ability of this model to learn complex representations in the input data and outperform ResNet V.1. As our data consists of different patterns in the veins at the back of the ocular globe (eye fundus), there is a need to capture the nuances for accurate diagnosis. This makes ResNet V.2 a strong candidate.

Each building block in our baseline consists of 16 filters, a kernel size of 3 and a stride of 1. The activation function is ReLU and we use batch normalization. Considering the tendency for the model to plateau a little before the 85th epoch, we decided to train our model on 85 epochs. The learning rate used is set to decrease with the number of epochs, from 1E-3 for epoch 1 to 80, to 1E-4 for epoch 81 to 120. We employ one-hot encoding and chose the categorical_crossentropy loss with Adam optimizer and a 80/20 training to test data split.

3.2 Chosen Model - Vision Transformer

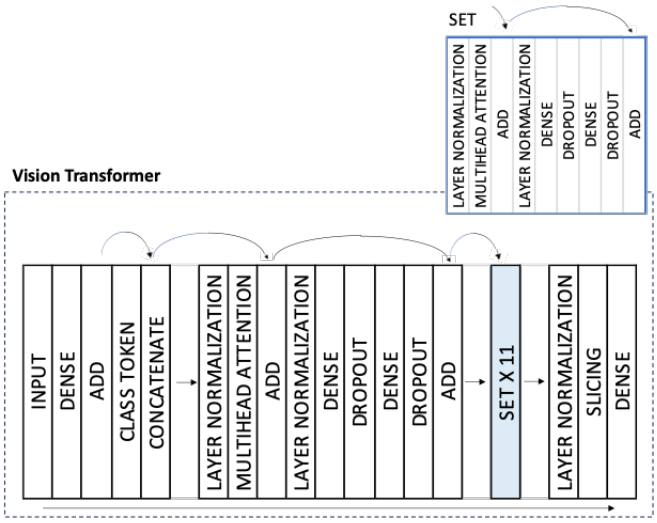


Figure 5: Schematic of Vision Transformer Model

We implemented a Vision Transformer (ViT) model to perform multiclass feature recognition on retinal scans. Our choice of ViT took into consideration the drawbacks of Convolutional Neural Networks (CNN) and the efficiency and accuracy of ViT [13]. Unlike conventional feature / image recognition algorithms which rely on some features of CNN, the transformer method can be directly applied to image classification tasks efficiently [14][15]. Particularly, we felt it would be an interesting comparison between state of the art ResNet (as our baseline model) with ViT that captures global features in lower layers [16]. Moreover, our survey of methods employed on Ocular Disease Recognition found commonly used methods such as CNN and its variants. While ViT was introduced in 2020 and has been used in retinal image diagnosis [17], the number of classes used in the predictions was limited [18]. Thus, we wanted to explore an implementation of the ViT with an efficient data augmentation approach across a wider number of classes.

Our model is trained on a 60/20/20 split, with a learning rate of 1E-4, and a batch size of 32. Images were split into 64 patches of size 25, and both the patches and their positional information were represented by tokens within the learned embeddings. The tokens pass through the transformer block to generate the classification prediction [19]. While a 2% improvement in accuracy can be achieved when the number of epochs increase from 85 to 300, considering the drastic slow-down in performance improvements after 85 epochs, and the need to balance the number of epochs and training iterations, we chose to limit our training to 85 epochs. Our model has 12 layers, with the intermediate embedding of dimension of 768, and that of the intermediate layer of the multi-layer perceptron (MLP) of 3072. The number of attention heads used in the multi-head self-attention mechanism of the transformer block is 12.

While it would be ideal to optimize for the dimension of MLP and the number of attention heads to improve the model's ability to recognize complex non-linear interactions and the subtle dependencies between the input tokens, these would be at the expense of computational cost. Hence, we decided not to change vary them in the interest of our project. With more compute units and time, we would consider comparing the model's performance for different number of attention heads particularly those between

8 and 16, with an MLP dimension of 2048 versus 4096, and for different dimensions of intermediate embeddings between 64 and 768, to find the optimal values.

4 Results & Evaluation

4.1 Quantitative Analysis

We used accuracy, precision, recall and F1 score as the metrics to measure the performance of both models. The equations for each are presented below.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \qquad precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN} \qquad F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Using the values of hyper-parameters for the ResNet V.2 model from the paper, we obtained a training accuracy of above 99% and a test accuracy under 44%. Our first hypothesis for this huge variance was that the model was too complex and was over-fitting the training data. Thus, we explored different sets of hyper-parameters using a grid search, varying the number of filters in each residual block and the total number of residual blocks. To use our compute credits efficiently, we randomly chose 7 out of 12 combinations of hyper-parameters in the grid search. The results of this performance benchmark are presented in figure 7. According to these results, the ResNet model performs best with 3 residual blocks and 16 residual layers.

Random Grid Search		Number of Filters			
		4	8	16	32
Number of residual blocks	1	train: 0.5125 test: 0.4212	not applicable	train: 0.5721 test: 0.4234	not applicable
	2	train: 0.4382 test:4302	train: 0.5278 test: 0.4257	not applicable	not applicable
	3	not applicable	train: 0.5737 test: 0.4392	train: 0.5915 test: 0.4504	train: 0.4915 test: 0.4302

Figure 7: Random Grid Search

While the combination with the highest accuracy was retained, the improvement in accuracy as a whole was minimal. To achieve better performance, we explored our second hypothesis – class imbalance, through data augmentation as mentioned in section2.1. The comparison between the performance of both models before and after data augmentation is displayed in figure 8. The stark contrast in performance confirmed that the class imbalance was our main issue.

	ResNet	ResNet	ViT	ViT
Data augmentation	Without	With	Without	With
Accuracy	0.4302	0.7419	0.4995	0.8612
Precision	0.4564	0.7542	0.5123	0.8775
Recall	0.4229	0.734	0.4807	0.8517
F1 score	0.439012	0.743963	0.4959972	0.864408

Figure 8: Comparison of performance without and with data augmentation in ResNet and ViT

We trained the ViT model with different values of the dropout rate to see how the regularization would affect the performance on the test data. The results are displayed in figure 9.

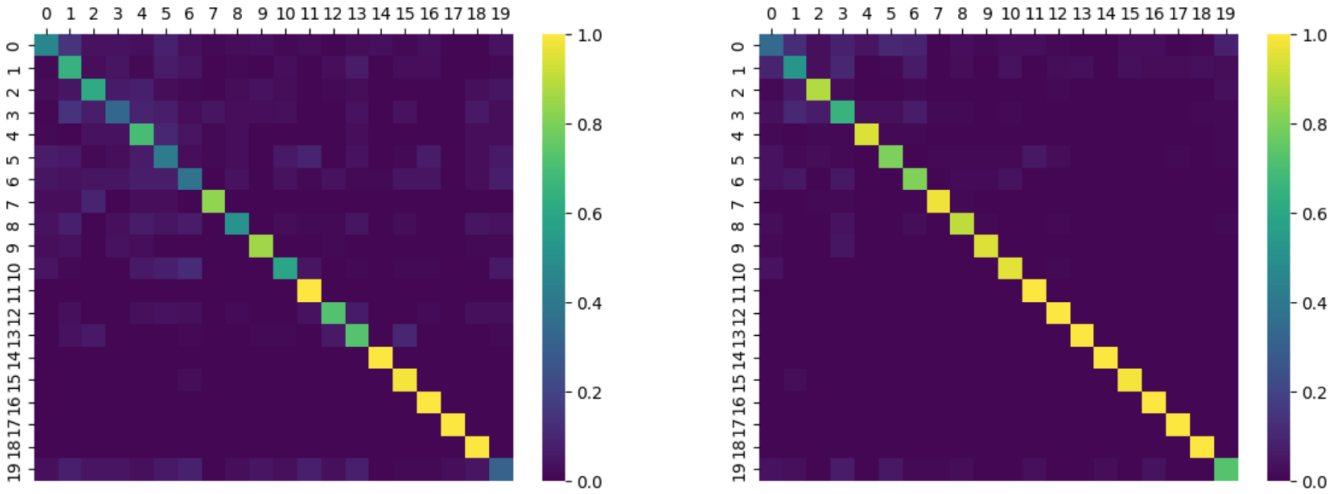
	ViT		
Dropout rate	0.0	0.1	0.2
Accuracy	0.6249	0.8612	0.8413
Precision	0.646	0.8975	0.8747
Recall	0.6119	0.8417	0.8229
F1 score	0.628488	0.868705	0.848009696

Figure 9: ViT Performance for various dropout rates

The best F1-score was achieved with a dropout rate of 0.1, with a value of 0.8687, indicating that moderate regularization has a strong positive impact on the performance of the ViT.

The best results for the ResNet model was achieved with 3 residual blocks and 16 residual layers. The test accuracy achieved was 0.7419 and the F1-score was 0.7439. The best results for the ViT model were achieved with a dropout rate of 0.1. The test accuracy achieved was 0.8612 and the F1-score was 0.8687. Within the limits of the fine-tuning performed on both models, the ViT model greatly outperformed the ResNet for this multi-class classification problem. It is possible that a much deeper fine-tuning would eventually bring the performance of the ResNet model on par with that of the ViT model. However, the extent to which the ViT model can be fine-tuned is greater and this is very likely to allow the ViT model to achieve even better performance.

We used confusion matrices (figure 10) to illustrate the distribution of the True Positives, True Negatives, False Positives and False Negatives for easier visualization [20], with the predictions on the horizontal axis and ground truth on the vertical axis. The highlighted diagonal in stark contrast with the off-diagonal terms is reflective of the relatively high accuracy of the model. From the confusion matrices, the classes that our ResNet baseline model had more difficulty predicting were 0, 3, 5, 6, 8 and 19, with an accuracy between 35% and 60%. On the other hand, that of the ViT model are classes 0 and 1, with an accuracy between 38% and 45%, and classes 3 and 19 with an accuracy around 60%. Other classes have accuracies that exceed 80%.



(a) ResNet

(b) Vision Transformer

Figure 10: Confusion Matrices

4.2 Qualitative Analysis

We selected a random sample of 200 images amongst wrongly predicted images to attempt to visually identify potential causes for mis-classifications. We identified 4 categories that might explain this, with their percentage out of all images: distortion (18%), blur (16%), low light (6%) and overly-cropped (8%). Sample images is available in figure .1 of the appendix. We believe performing training and testing of our models without these images, or a particular category of images, would allow us to confirm whether they are indeed the reason for the lower accuracy of our models. Another method that could help identify contributions to poor performance could be saliency maps. We believe these would allow us to better visualize parts within each image that our model focuses on when making predictions. By observing saliency maps associated with correctly and incorrectly classified input, we can visualize parts of the images that our model may be misinterpreting or ignoring. The comparison between saliency maps can also provide a way to check if our model has focused on relevant features and could also provide greater insights into our model's performance.

Lastly, the accuracy of the model for each class as reported with the confusion matrix might be slightly bias due to the small quantity of images in certain classes. For instance, class 14 ("LS") and 17 ("ASR") show an accuracy close to 100%, but only contained 2 and 6 images respectively in the original dataset. This means that the 481 and 533 images used in training, validatation and testing of the model were all variations of these 2 and 6 images. This may have led the model to overfit these images. If this is the case, the model might actually perform very poorly on new images for those two particular diseases that are unseen. We would therefore recommend using all new images for any of the less-represented classes to better train the model.

5 Contributions

We actually really did the project together. From coming up with our project idea, the baseline model, the actual model, writing and debugging the code, and writing the report. The individual parts are only running the models where we split for more efficiency.

6 Code

The code for this project can be found at this GitHub repository: https://github.com/sf-nf-ai/cs230_final_project . This contains 3 python files required to run the ViT model, 1 file required to run the ResNet model, the folder containing the trained ResNet model and its weights, and the folder containing the trained ViT model and its weights.

References

[1] World Health Organization (WHO). Ageing and health, 2022. https://www.who.int/news-room/fact-sheets/detail/ageing-and-health.

[2] TJ MacGillivray, Emanuele Trucco, JR Cameron, Baljean Dhillon, JG Houston, and EJR Van Beek. Retinal imaging as a source of biomarkers for diagnosis, characterization and prognosis of chronic illness or long-term conditions. *The British journal of radiology*, 87(1040):20130832, 2014.

[3] viso.ai. Vision transformers (vit) in image recognition - 2023 guide - viso.ai., 2023. https://viso.ai/deep-learning/vision-transformer-vit/.

[4] Edward Ho, Edward Wang, Saerom Youn, Asaanth Sivajohan, Kevin Lane, Jin Chun, and Cindy ML Hutnik. Deep ensemble learning for retinal image classification. *Translational Vision Science & Technology*, 11(10):39–39, 2022.

[5] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 74–92. Springer, 2022.

[6] MA Rodriguez, H AlMarzouqi, and P Liatsis. Multi-label retinal disease classification using transformers. *IEEE Journal of Biomedical and Health Informatics*, 2022.

[7] J Brownlee. Best practices for preparing and augmenting image data for cnns. *machinelearningmastery*, 2019.

[8] TensorFlow. Data augmentation, 2023.

[9] Wenru Dong. What is opencv's inter_area actually doing?, 2018. https://medium.com/wenrudong/what-is-opencvs-inter-area-actually-doing-282a626a09b3.

[10] pawangfg. Residual networks (resnet) – deep learning. https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/.

[11] Renu Khandelwal. Visualizing deep learning model architecture, 2022. AIGuys.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[13] Bo-Kai Ruan, Hong-Han Shuai, and Wen-Huang Cheng. Vision transformers: state of the art and research challenges. *arXiv preprint arXiv:2207.03041*, 2022.

[14] Yeonwoo Jeong, Yu-Jin Hong, and Jae-Ho Han. Review of machine learning applications using retinal fundus images. *Diagnostics*, 12(1):134, 2022.

[15] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

[16] Title = AI-Scholar.

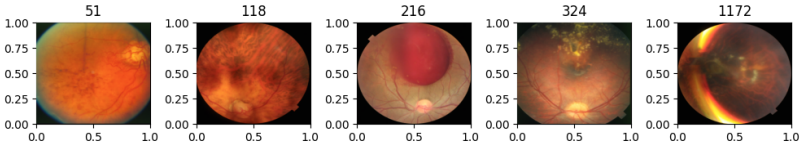
[17] En Zhou Ye, Joseph Ye, and En Hui Ye. Applications of vision transformers in retinal imaging: A systematic review. 2023.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

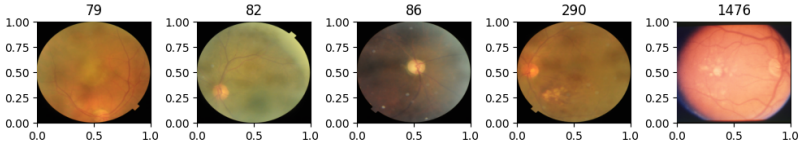
[19] Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*, 2021.

[20] Dennis. T. Confusion matrix visualization, 2019. https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea.

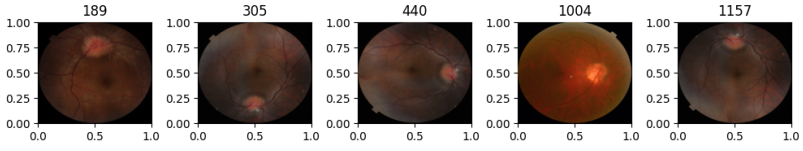
7 Appendix



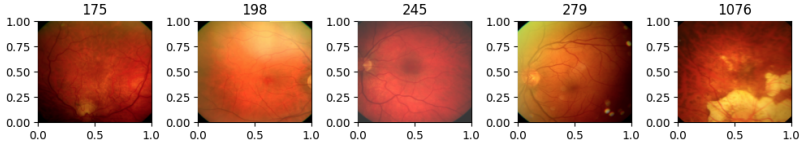
(a) Aberrations



(b) Blur



(c) Low Light



(d) Overly Cropped

Figure .1: Possible contributions to poor test results on certain classes.