Background and Significance of Project

Optical Coherence Tomography (OCT) is a commonly performed diagnostic test designed to assist doctors in identifying retinal diseases, such as choroidal neovascularization (CNV), Diabetic macular edema (DME), and Drusen, that are the most common diseases resulting in the loss of sight. Approximately 30 million OCT scans are performed each year with an average price of $99 per scan. There is a significant effort world-wide in automating retinal diagnosis with the help of supervised deep learning models to bring down the cost and increase the accessibility of diagnosis.

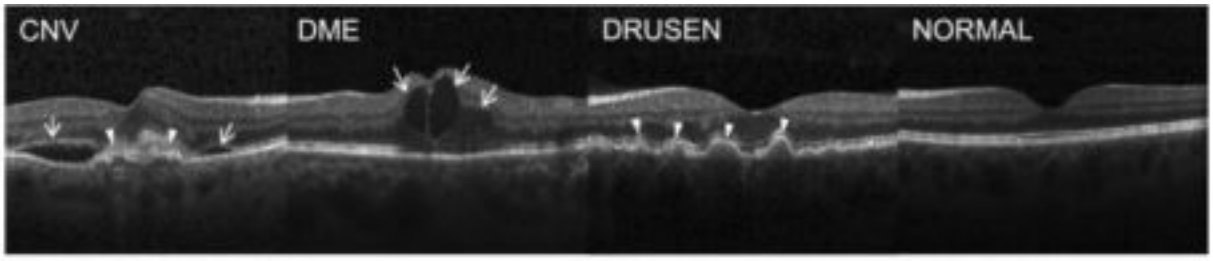
The biggest bottleneck in this endeavor, of automating retinal diagnosis, is the exorbitant price and unavailability of doctors for labeling images. This seriously limits the extent to which traditional supervised learning models can tackle this problem. With the help of self-supervised contrastive learning with downstream supervision, our framework can help develop models that can learn better, more robust, representations with just a fraction of the data required by traditional approaches. With improved representations, our approach can develop models that are less prone to annotation errors that are prevalent in the field of medical imaging. Additionally, by analyzing the representations of the self-supervised model in a latent-vector space, we enable the possibility of active learning by subsampling. In other words, we will be able to select the images that add the most value for the classification task and subsequently send them for labeling, thereby increasing the efficiency of the labeling process.

Related Work

Description	Paper	Github	Others
Published SOTA	https://bmcophthalmol.biomedcentral.com/articles/10.1186/s12886-020-01382-4	N/A	
Kaggle kernels	N/A	https://www.kaggle.com/c/diabetic-retinopathy-detection/notebooks	the accuracy scores are incorrect as the validation set is sampled from train set
Self Supervised	https://arxiv.org/abs/2006.10029	https://github.com/google-research/simclr	Based on Imagenet dataset

Explanation of Data sets

Data set includes the four classifications (CNV, DME, DRUSEN and NORMAL). More details on the types can be found here. Below is a picture differentiating between the classes



Data set contains 83,489 train and 968 test samples with varying image sizes in grayscale. It is imbalanced with the below distributions and we are using class weights in the loss function to mitigate the imbalance. From an implementation perspective, we are hosting the dataset in GCP and our wrappers abstract that storage

Classification Type	Count (Train set)
CNV	26318
DME	8616
DRUSEN	11350
NORMAL	37205

Explanation of Processes

Below is the workflow/process we have used

- Dataset
    - Using the augmentation strategies listed in simclr paper
- Supervised
    - Added conv + dense layer to Resnet50V2 base (imagenet) and retrained the last layers
    - Fine tuning the above architecture by retraining last few layers
- Self-Supervised and distributed training
    - Retrained entire simclr network with our dataset to learn representations
    - Fine tuning - Pending, approach is to retrain an MLP head
- Hyper parameter optimizations
    - All networks are trained using hyper param sweeps (from weights and biases)

- For Simclr due to batch size increase using distributed training

Experiments & findings from the above process is summarized in https://github.com/anoopsanka/retinal_oct/blob/main/README.md

Explanation of Outcomes

Data Augmentation

Data augmentation increases the diversity data available for training the model, without actually collecting new data. We follow the augmentation strategy by SimCLR closely by sequentially applying random cropping, random color distortion (contrast, brightness, saturation, hue), and random gaussian blur. On top of the augmentation suggested by the original SimCLR paper, random rotation between (-45, 45) degree is applied to the image. In the supervised model, the data is cropped to (224,224,3) and the current unsupervised model is trained on (128, 128, 3) image data.

Supervised model

Training outline

Following closely the guidelines in Transfer learning & fine-tuning (keras.io), the pretrained resnet model is freezed so to avoid destroying any of the information the pretrained model contain during the future training rounds. A new, trainable layers are added on top of the frozen resnet model. In our case, the classification layer would learn the make predictions on the features extracted by the pretrained model. Before finetuning, the new layer is first trained to convergence. The resnet model is then unfreezed, and retrain on the dataset with a low learning rate.

The training dataset is inherently imbalanced. To tackle this, the loss function is weighted by the class weight of the imbalance dataset, where the minority class is weighted more heavily in the objective function.

Performance

The trained classification model has trouble differentiating Drusen and CNV.

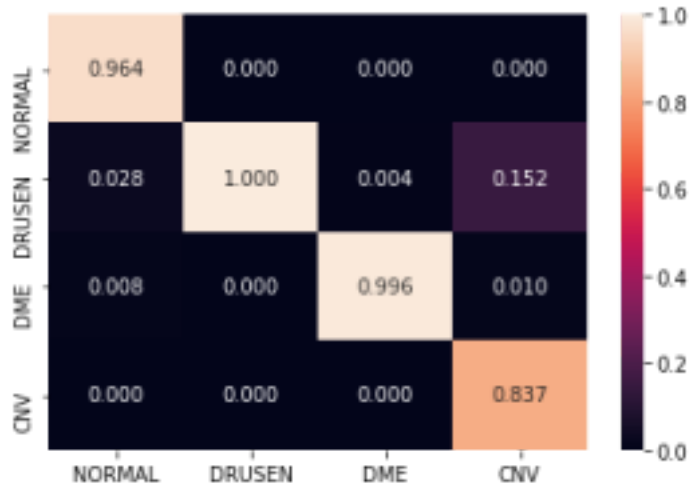


Fig: Confusion matrix for the supervised model on the test set data. The values are normalized by the number of predicted labels.

```
 precision recall f1-score support

 0 0.96 1.00 0.98 242
 1 1.00 0.79 0.88 242
 2 1.00 0.98 0.99 242
 3 0.84 1.00 0.91 242

 accuracy 0.94 968
 macro avg 0.95 0.94 0.94 968
weighted avg 0.95 0.94 0.94 968
```

Table: Classification report of the respective class predictions

Saliency Map

GradCAM [1610.02391] Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization (arxiv.org) is applied to the model to visualize the important regions in the image for predicting the concepts.

Unsupervised model

Saliency Map, classification report, confusion matrix

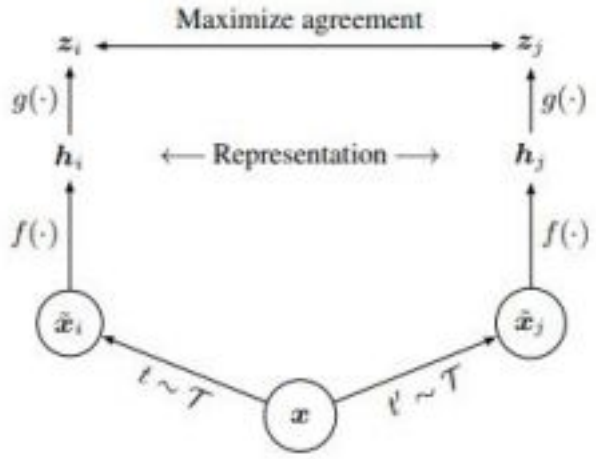


Fig: SimCLR model architecture (taken from Figure 2 of A Simple Framework for Contrastive Learning of Visual Representations (arxiv.org)). The input image x are separated augmented with different parameters to form x_i and x_j. The two copies are fed into the feature extractor to output the representation. The model is being trained on contrastive loss, where the training objective is to identify its own augmented copy from the augmented representation.

On top of the model shown above, a simple dense layer is attached to the output of the base resnet model for simple classification taske. A tf.stop_gradient is placed between the output from the resnet model to the classification layer. This ensures that the gradient information from the label data is not flowing back to the resnet model, while the classification results serve as an proxy to how well SimCLR can extract features relevant to a linear classification model.

The stop gradient operator can be removed once the model is trained to convergence for further fine-tuning, similar to the training steps outlined for finetuning classification model. Note that this hasn't been performed yet in the current report.

The SimCLR model is trained on batch size of 128 and a detailed study has shown that the classification performance scales with increasing batch size. Currently, in order to fit a batch size of 128 into a single GPU machine, the image is downsized to 128x128. LAMB optimizer [1904.00962] Large Batch Optimization for Deep Learning: Training BERT in 76 minutes (arxiv.org) is used to cope with the large batch size involved for optimization. And a linear warmup followed by a cosine annealing training schedule is adopted.

Performance

Fig: Confusion matrix of the SimCLR model on the test set data. Noted that the resnet model has not been finetuned for the classification task.

	precision	recall	f1-score	support
NORMAL	0.88	0.83	0.85	242
DRUSEN	0.97	0.46	0.63	242
DME	0.86	0.79	0.82	242
CNV	0.59	0.98	0.73	242
accuracy			0.76	968
macro avg	0.82	0.76	0.76	968
weighted avg	0.82	0.76	0.76	968

Table: Classification report for the SimCLR model.

Representation Learning

We extract the feature layers from the two models to visualize the clustering properties in the hidden dimension. For the classification model, the final hidden layer prior to the classification layer is taken, as for the SimCLR model, the projection head output trained on the contrastive loss objective is taken. They are then normalized by their l2 norm. And project to lower dimension with various manifold reduction strategies. The classification model shows distinct separations into four clusters. This explains the high accuracy achieved by the classification model. The SimCLR model has not been finetuned on the label images and is therefore agnostic to the ground truth labels. While the separation is not as apparent as in the supervised

model, similar classes are shown to cluster on their own. This signifies the useful representation learnt by SimCLR through a completely unsupervised contrastive learning approach.

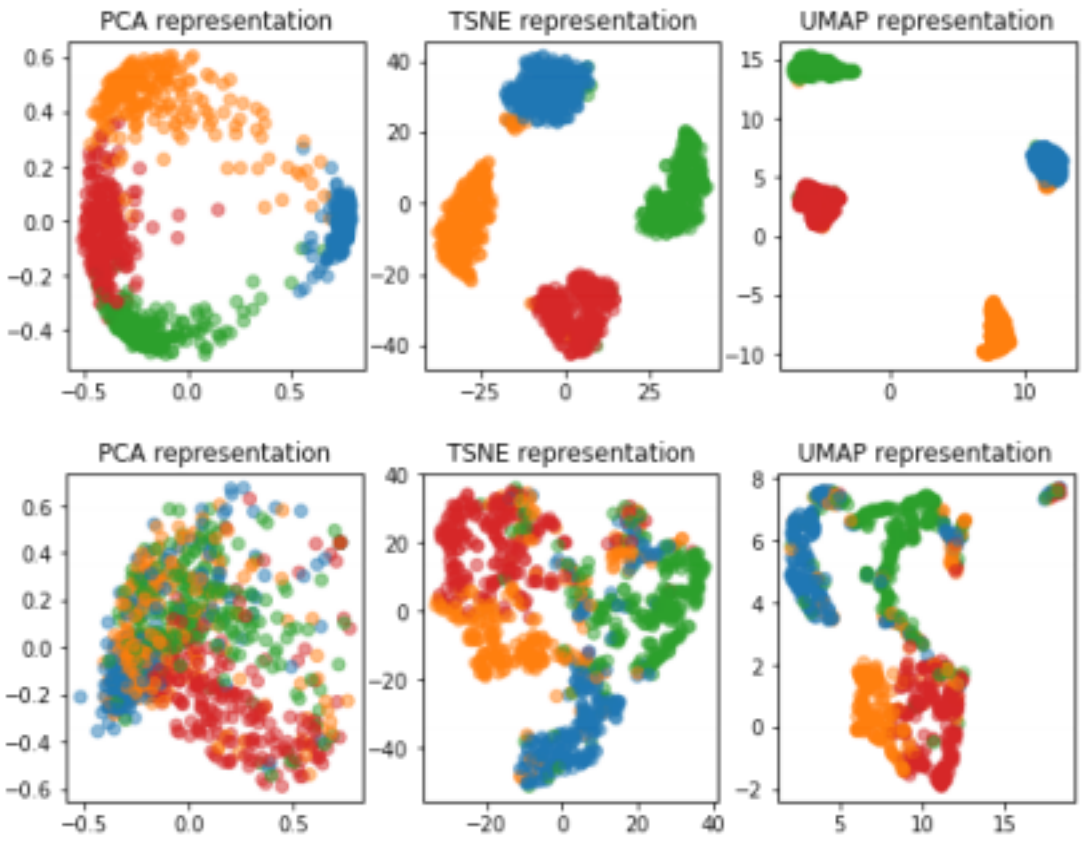


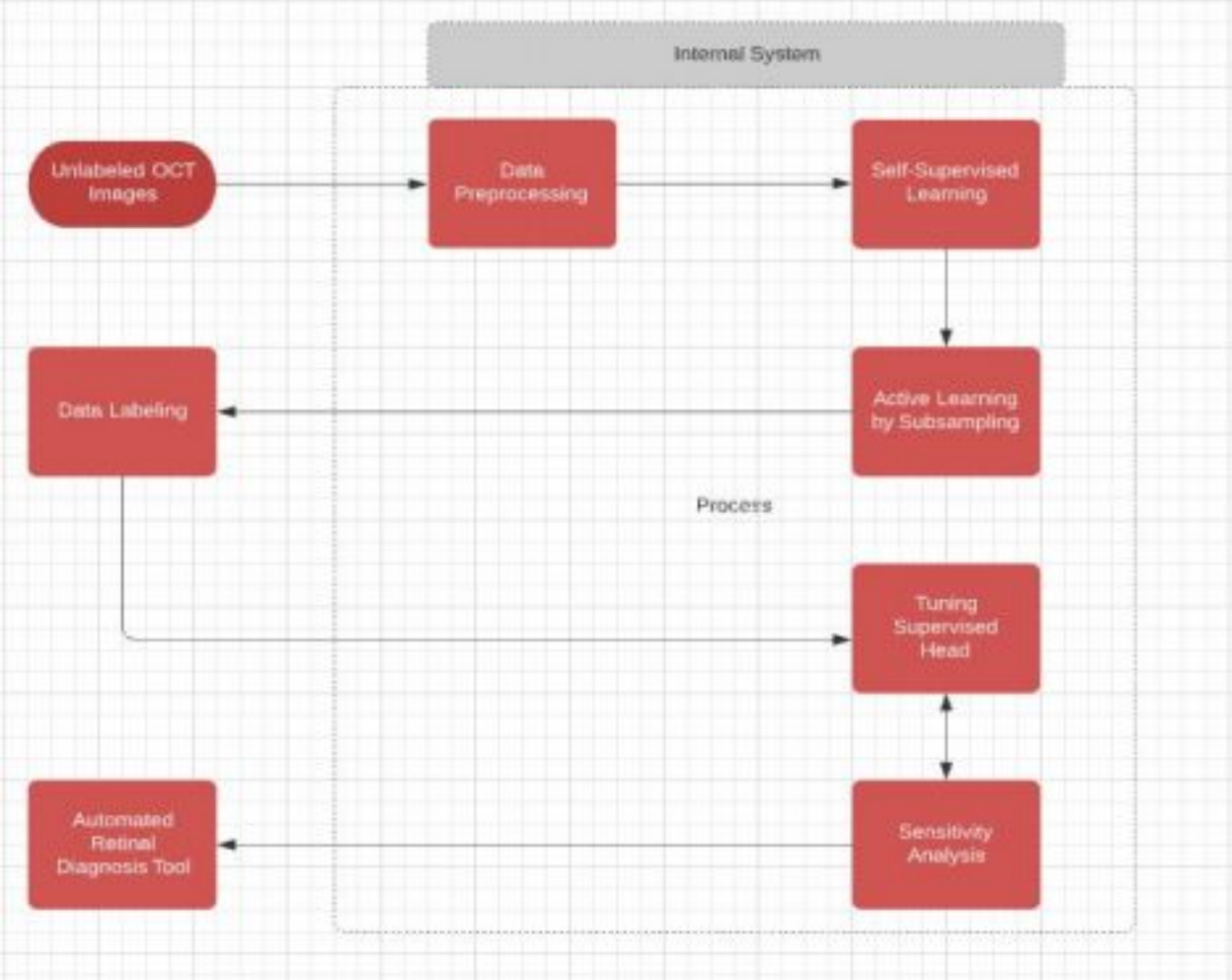
Figure: Representation learnt by the classification model and the SimCLR model. Note that the representation layer selected from SimCLR has not seen any labeled images at all.

System Design

The final system that we develop will aid not only in automating the diagnosis of Retinal diseases but also aid in improving the efficiency of data labeling. By identifying the most valuable images to label, using active learning, and learning better representations, using self-supervised learning, we will be able to significantly reduce the time, effort, and cost associated with data-labeling.

The unlabeled-images are pre-processed by our system and fed into the self-supervised learning model, which learns the representations of the training data in a latent-vector space. Based on the decision boundaries between the clusters formed in this space, we identify the most informative images to label. These images are then used to fine-tune a supervised model that is developed by adding an MLP layer on top of the self-supervised model that is

pre-trained
using the unlabeled images. The trained classifier is then passed through a sensitivity-analysis pipeline that will measure its quality in terms of responsiveness to augmentations, susceptibility to mislabeled data, etc.



Ethical Considerations

Studies have shown that eye color has significant correlations with gender (https://pubmed.ncbi.nlm.nih.gov/23601698/#:~:text=However%2C%20we%20have%20found%20an,%2C%20males%20lighter%20than%20females). Since we are using OCT images, which are devoid of color channels, our system is immune to gender based bias.

However, since the age of a person can affect the retinal vasculature (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5527847/), the predictions of our system can be impacted by biased data. Since the open-source dataset that we are using does not provide the demographic information related to the people associated with the OCT scans, we are unable to ensure that the data used is representative of every age group.

Future work and Timeplan

Having set up the base infrastructure for experimentations, we have started tuning the networks for the self-supervised and the subsequent supervised tasks. We have also started developing the Sensitivity analysis pipeline to further increase the robustness of our system. Our next steps will include the following major tasks:

- Experimentations with a fraction of the labeled data to identify the reals in which our framework can outperform the traditional supervised models
- Active Learning by Subsampling to improve the efficiency of labeling

Supervised Model Experiment Results

Configuration Used:

Learning Rate Initial Training: 0.002
Learning Rate Fine Tuning: 0.00005
Early Stopping Patience: 10
Sample Sizes used: 1% (of training data), 5%, 10%, 98%

Random Seed: 7
Image Size = 224
Batch Size = 32
Max Epochs = 100
Crop Proportion = 0.875

Approach:

At every step, a fraction of the data (1%, 5%, 10%, 50%, or 98%) was used for training a supervised ResNet model loaded with the ImageNet weights. This model was then used to evaluated using the following resources:

- Confusion Matrix
- Dimensionality-reduced-projections: The test images were passed through the model to extract the activations from the final GlobalAveragePooling2D layer. These representations, from a latent-vector space, were then reduced using two dimensionality reduction algorithms: PCA and UMAP. Using the 2-D and 3-D visualizations of the projections, the performance of the classifier was qualitatively studied.
- Loss/Accuracy curves from model training
- GradCAM activations: 3 images (one each from Drusen, CNV, and DME) were sampled from the test set and used for studying the saliency maps and the occlusion sensitivity maps. These 3 images were used across all the fractions of the data for comparability.

Results for Image Size: 224

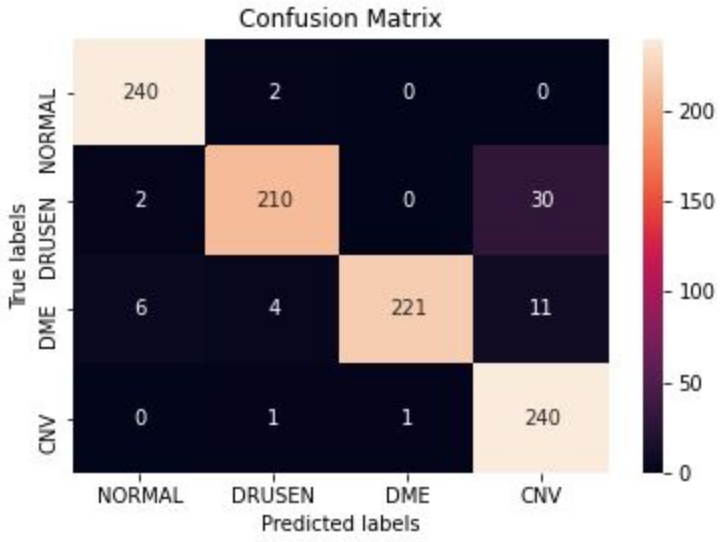
Combined Results (Averaged over 5 randomly sampled train splits):

The following results were collected by training the model on 5 randomly selected splits of the training data and testing on a separately maintained test set.

S. No.	Data Fraction	Accuracy	Precision	Recall	F1 Score
1	1%	0.910123	0.922199	0.910124	0.908315
2	5%	0.983471	0.984496	0.983471	0.983487
3	10%	0.988017	0.988345	0.988017	0.988017
4	100%	0.998967	0.998971	0.998967	0.998967

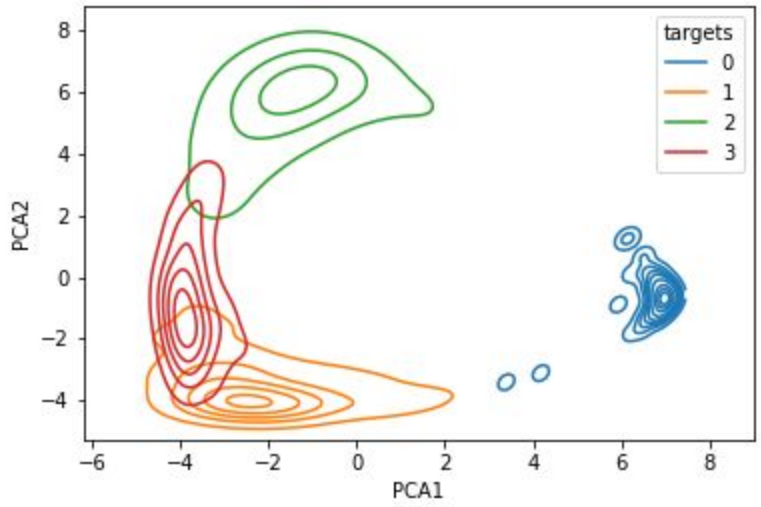
Trained On 1% of the training data and tested on the entire test data

Confusion Matrix:



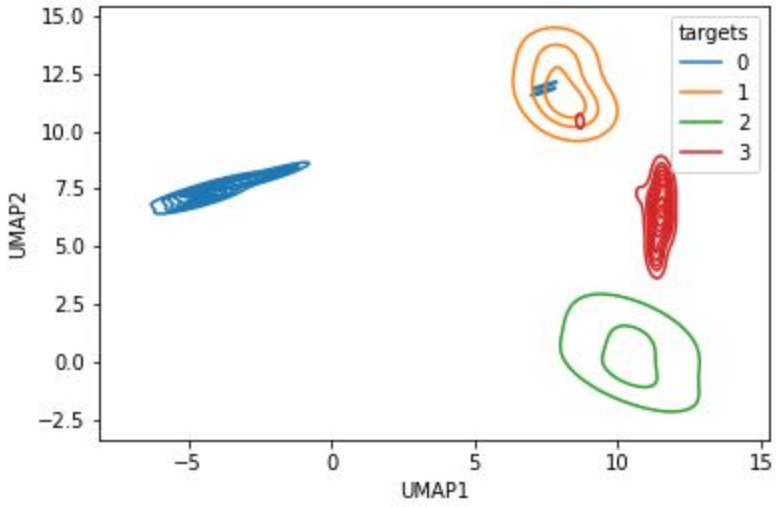
Projections after Dimensionality reduction:

PCA 2D:



PCA 3D:
https://drive.google.com/file/d/1xRryIzA3VkhMSEt9GJXYNh23xMxAIJ6a/view?usp=sharing

UMAP 2D:

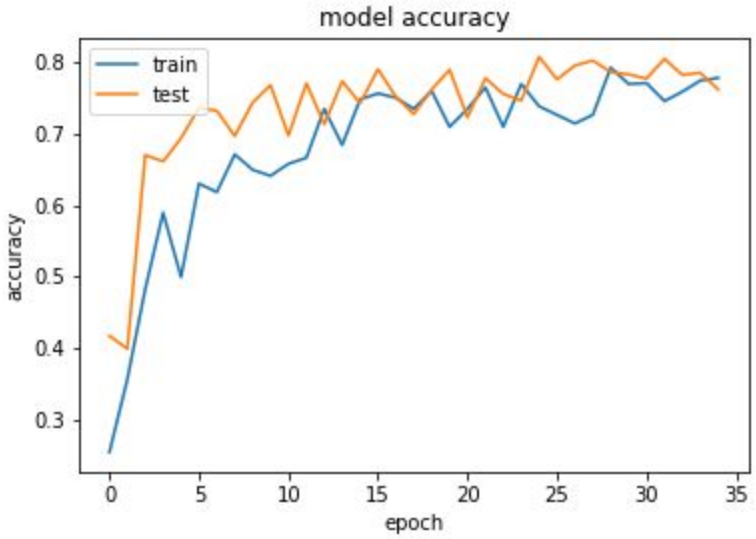


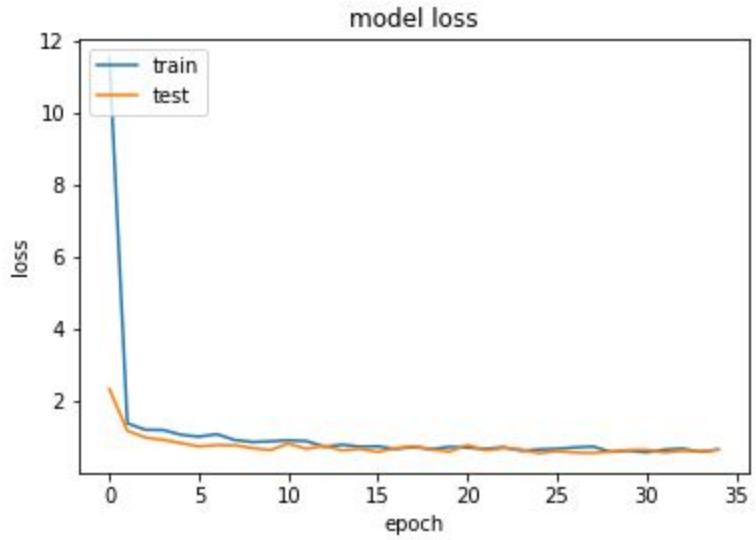
UMAP 3D:

https://drive.google.com/file/d/1jwqR_GZNTb5v8llqADgmkS_sQwi5DLSA/view?usp=sharing

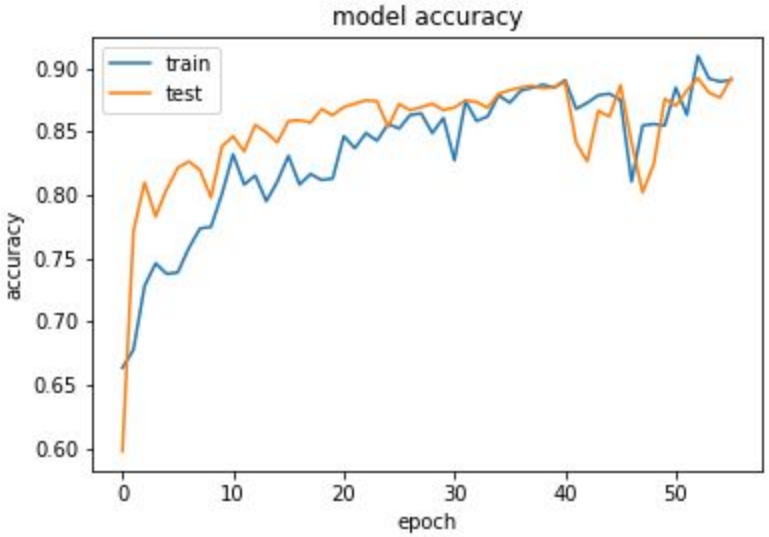
Training History:

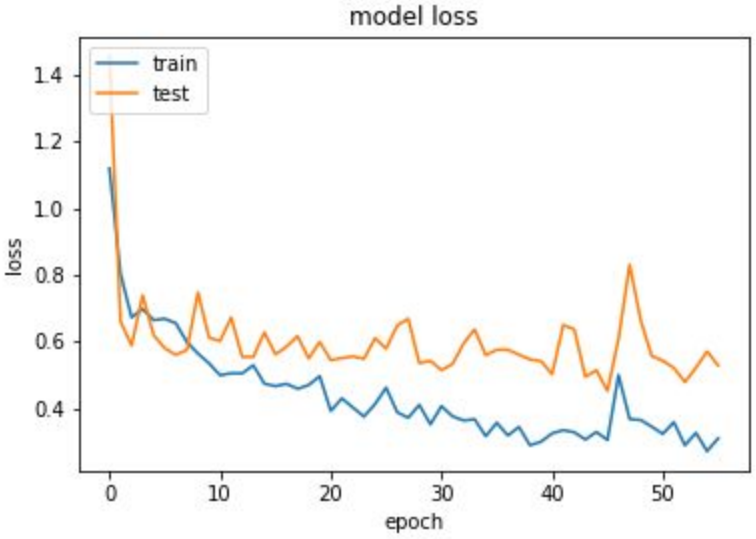
Initial Training:



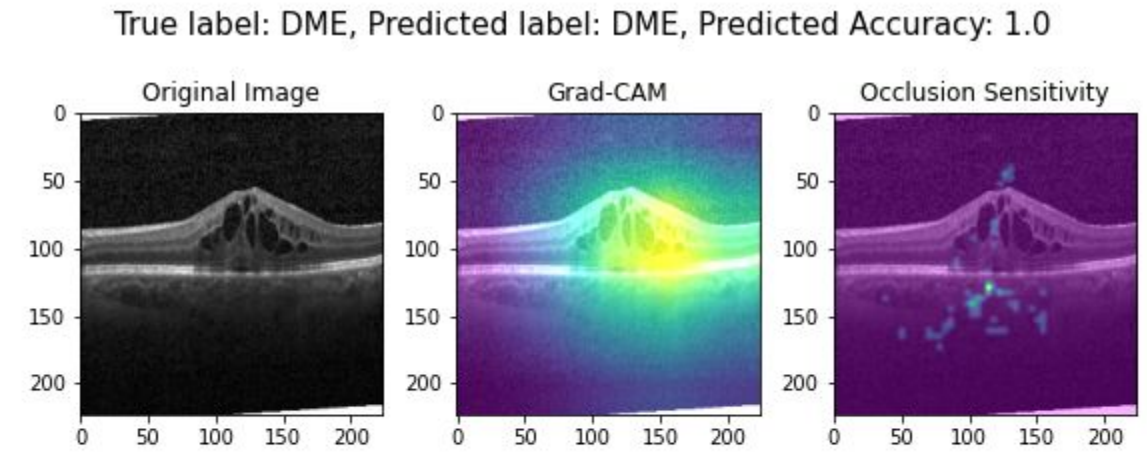


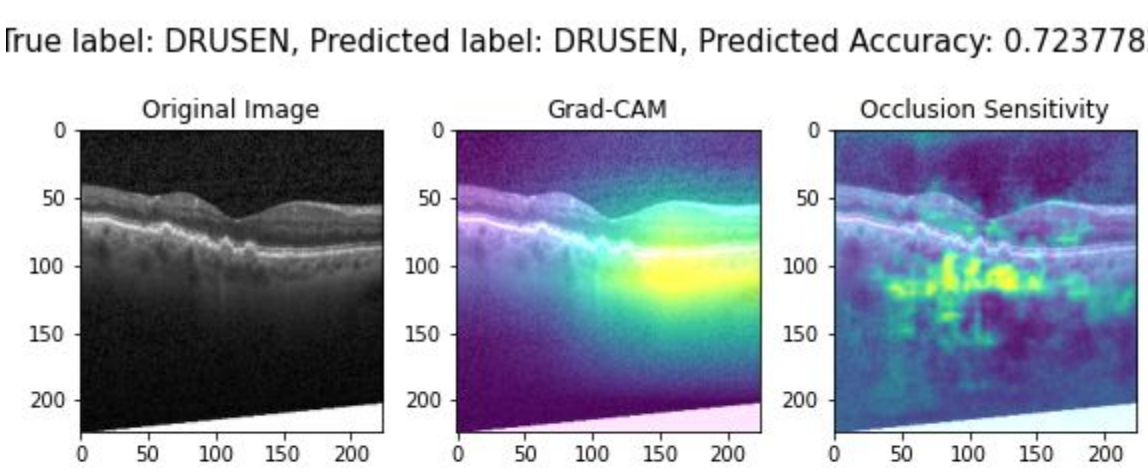
Finetuning:

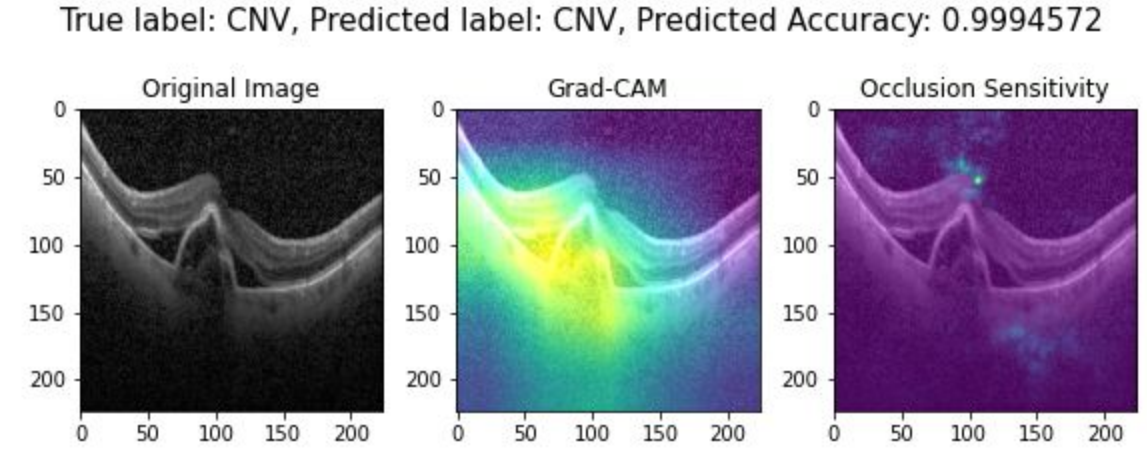

model accuracy
train
test
accuracy
epoch
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0
10
20
30
40
50


model loss
train
test
loss
epoch
0.4
0.6
0.8
1.0
1.2
1.4
0
10
20
30
40
50

GradCAM (Taken for 3 images from the test set):

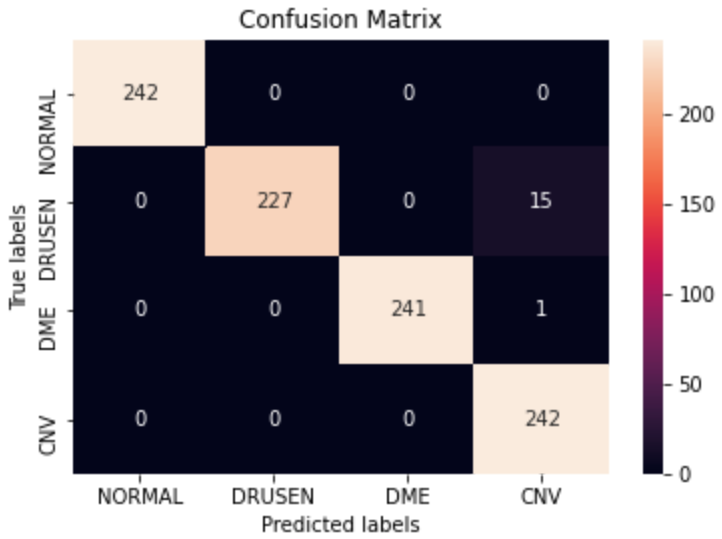






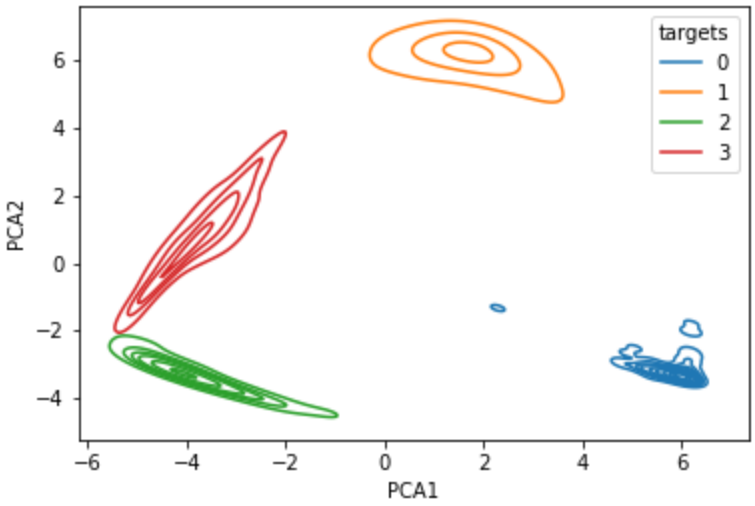
Trained On 5% of the training data and tested on the entire test data

Confusion Matrix:



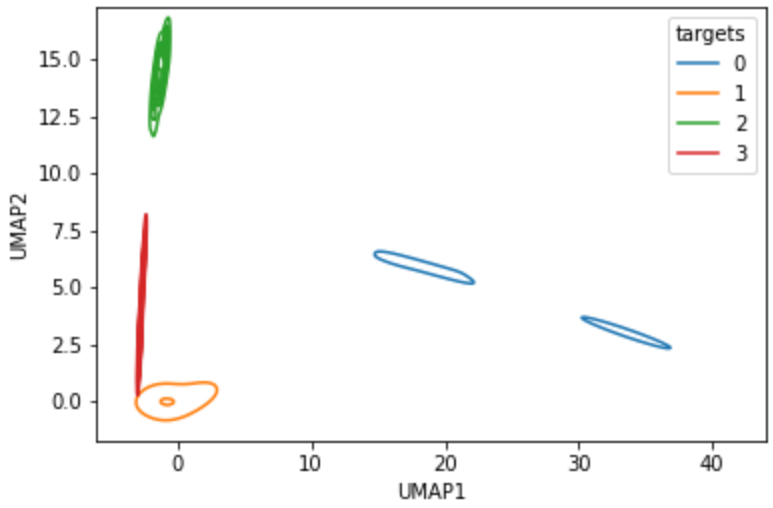
Projections after Dimensionality reduction:

PCA 2D:



PCA 3D:
https://drive.google.com/file/d/1-kejmvkw9qn6cABd4rjSvGhEG48SbVTL/view?usp=sharing

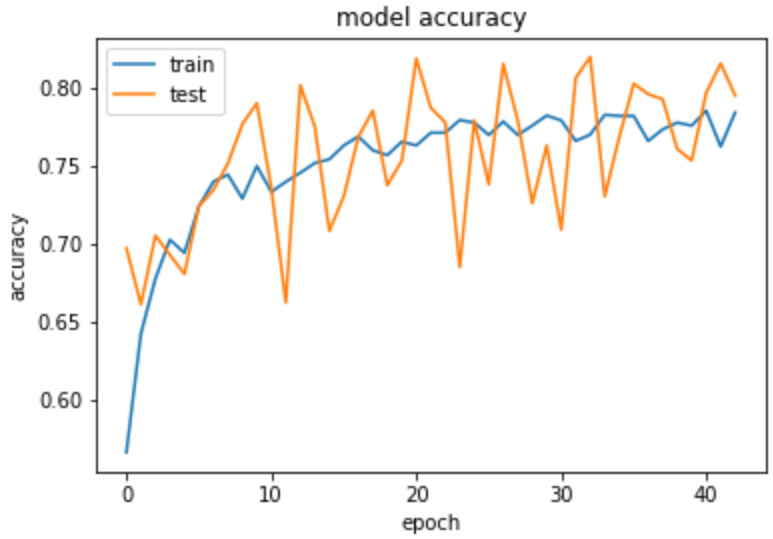
UMAP 2D:

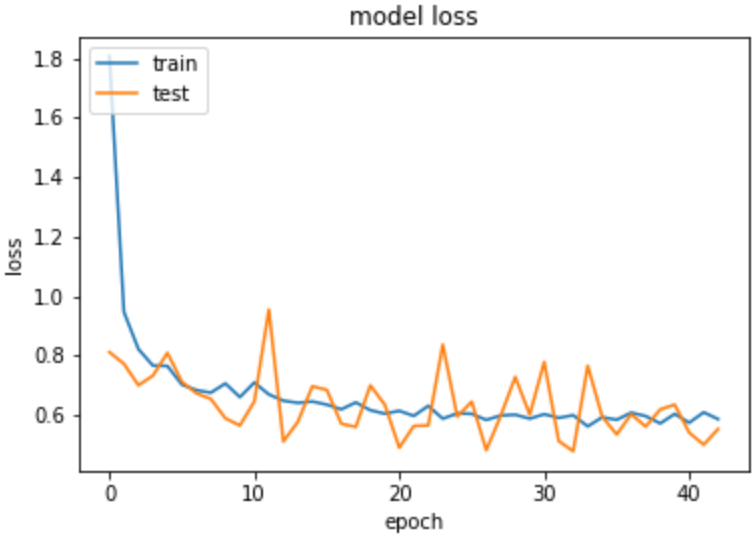


UMAP 3D:
https://drive.google.com/file/d/1D6kVFFagEOhCpf6h5P5noaD7EFq5pEse/view?usp=sharing

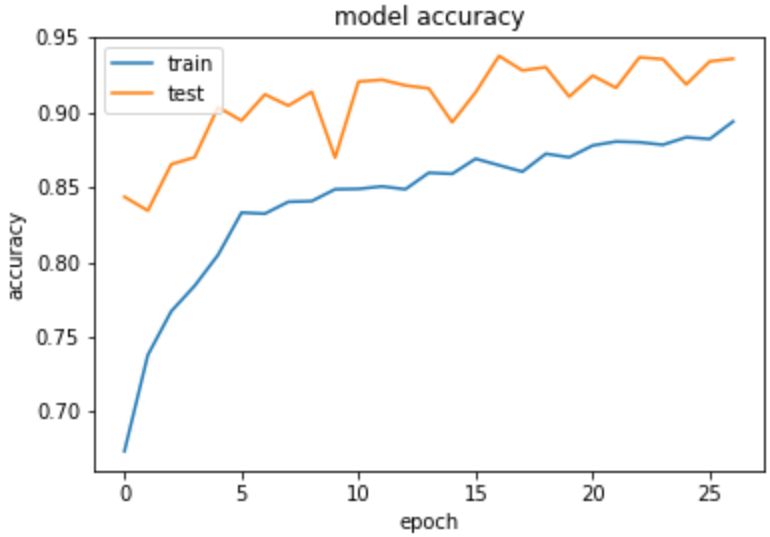
Training History:

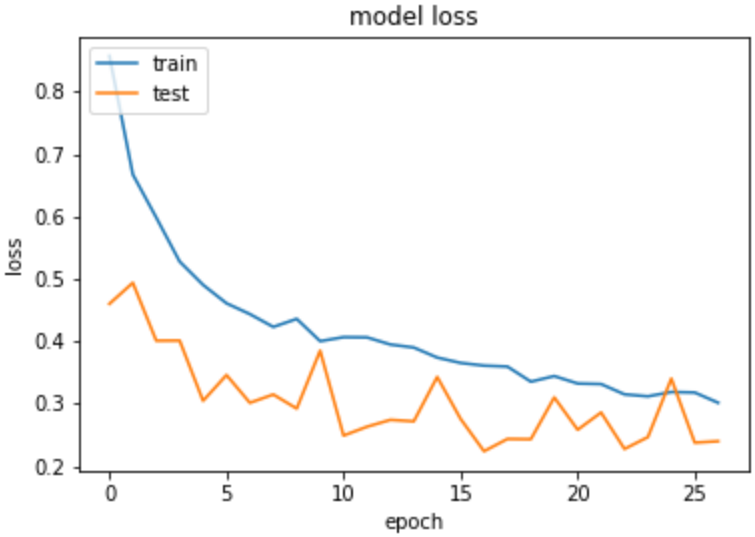
Initial Training:




model loss
train
test
loss
epoch
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0
10
20
30
40

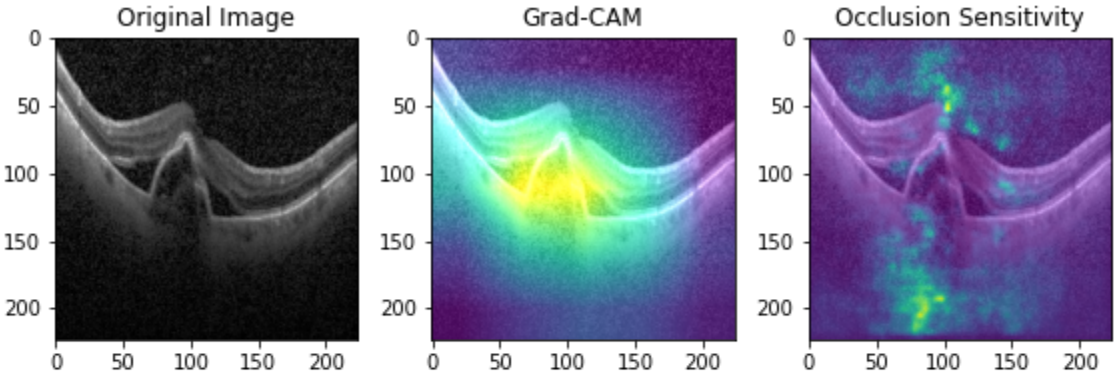
Finetuning:


model accuracy
train
test
accuracy
epoch
0.95
0.90
0.85
0.80
0.75
0.70
0
5
10
15
20
25

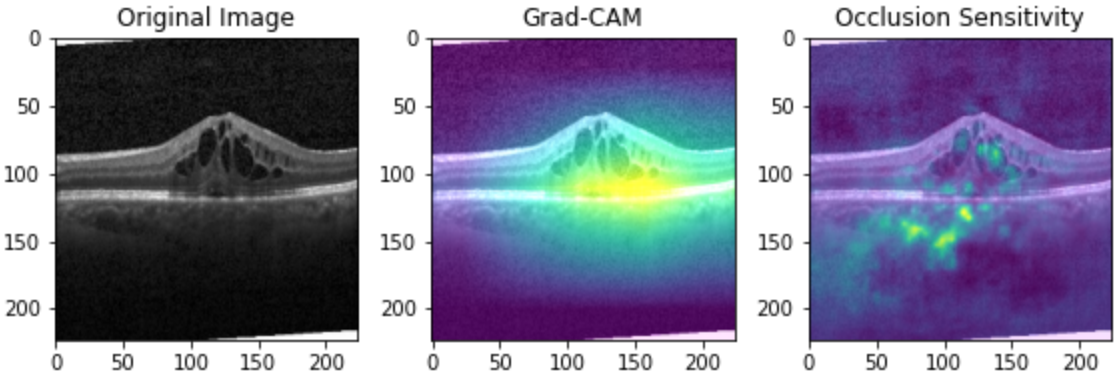


GradCAM (Taken for the same 3 images from the test set):

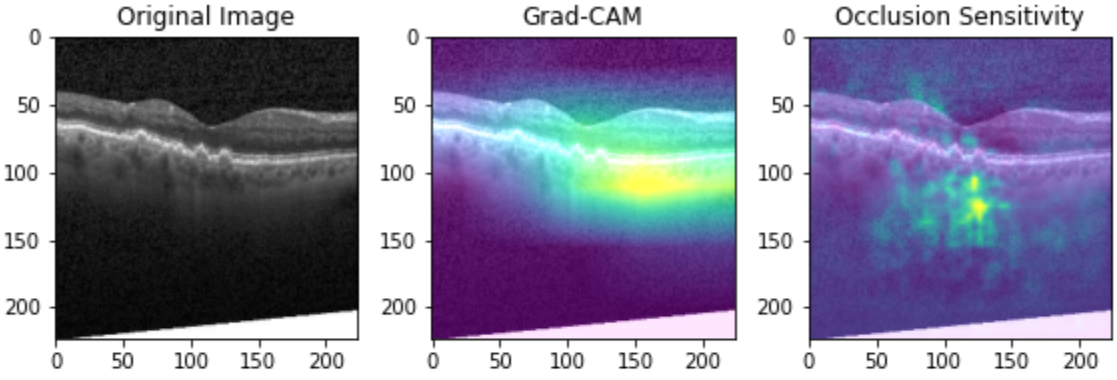
True label: CNV, Predicted label: CNV, Predicted Accuracy: 0.9996903



True label: DME, Predicted label: DME, Predicted Accuracy: 0.9964204

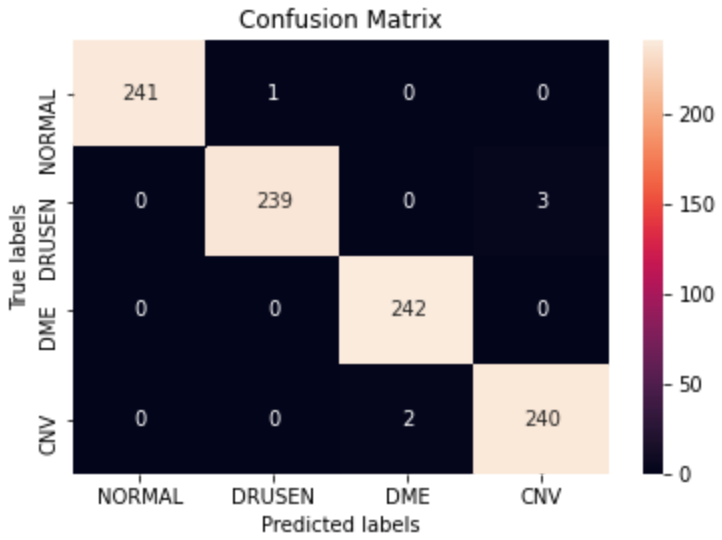


True label: DRUSEN, Predicted label: DRUSEN, Predicted Accuracy: 0.8488055



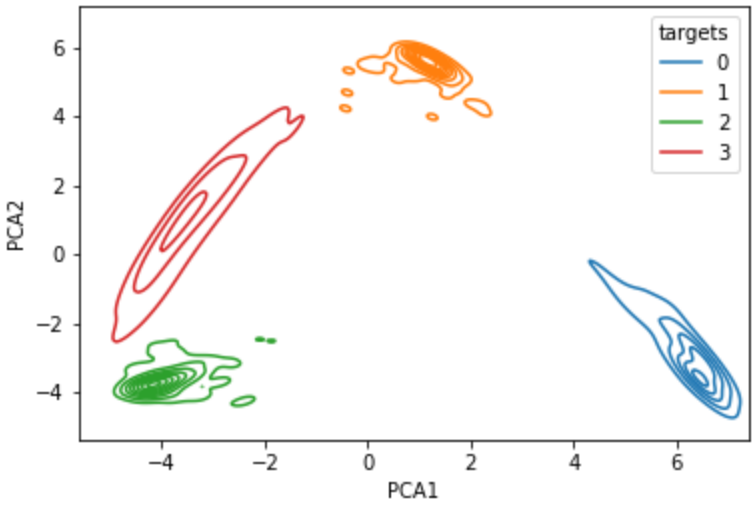
Trained On 10% of the training data and tested on the entire test data

Confusion Matrix:



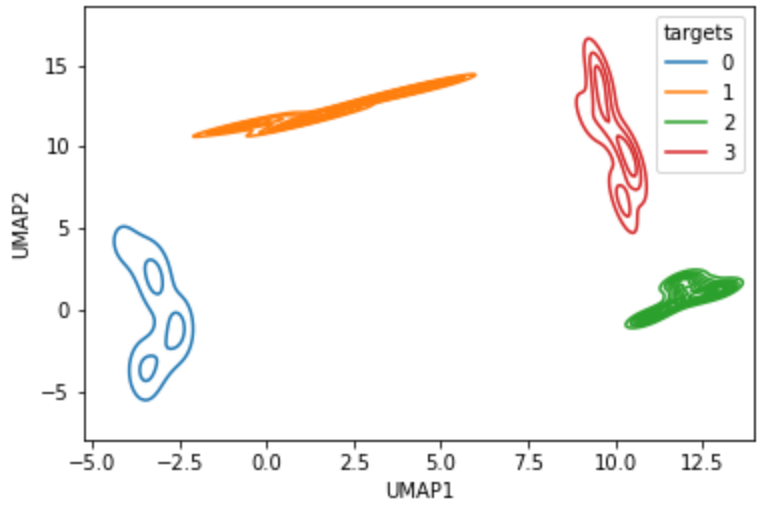
Projections after Dimensionality reduction:

PCA 2D:



PCA 3D:
https://drive.google.com/file/d/1V4NLjoosa9poPZ0UAiFSI9zjlhTaHqTn/view?usp=sharing

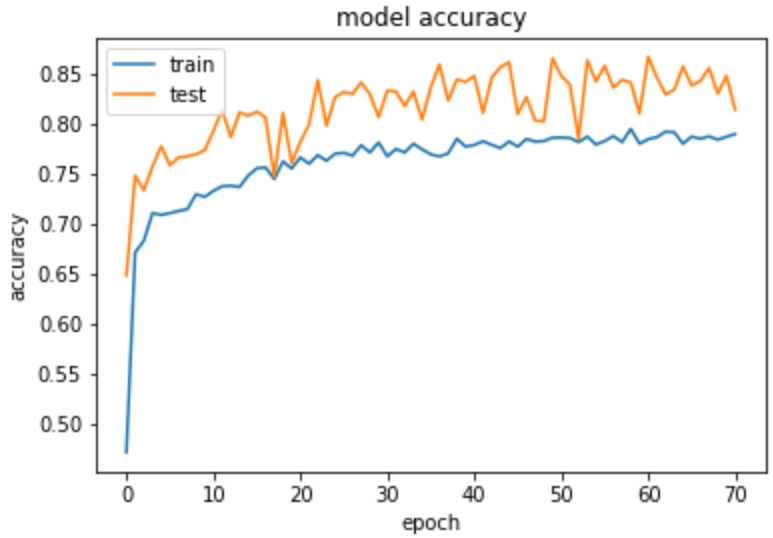
UMAP 2D:

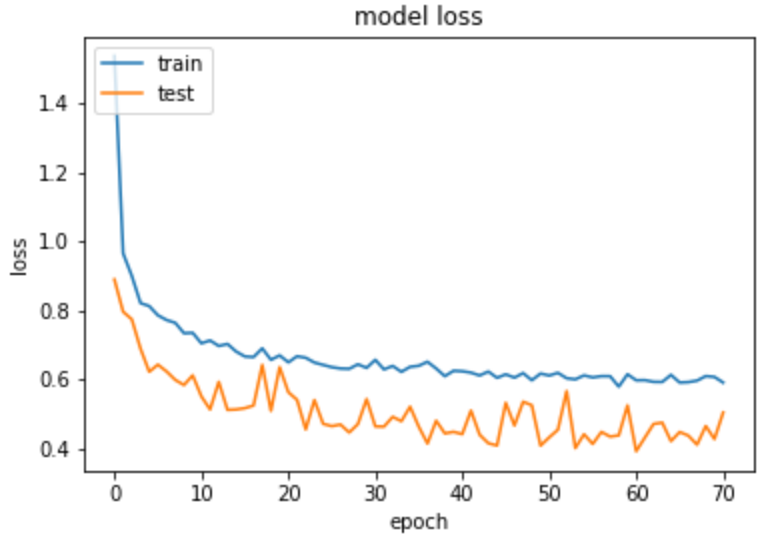


UMAP 3D:
https://drive.google.com/file/d/1439ZTlcGEAizeK4VFFGSLNF1Zci8WcBH/view?usp=sharing

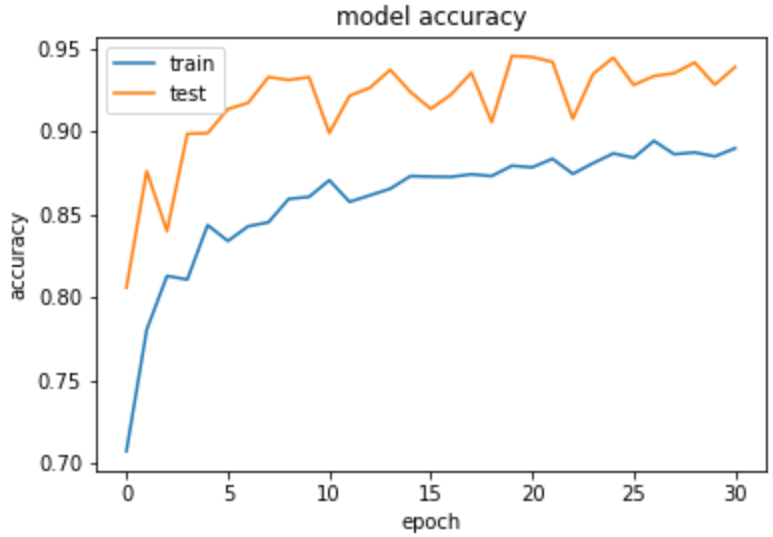
Training History:

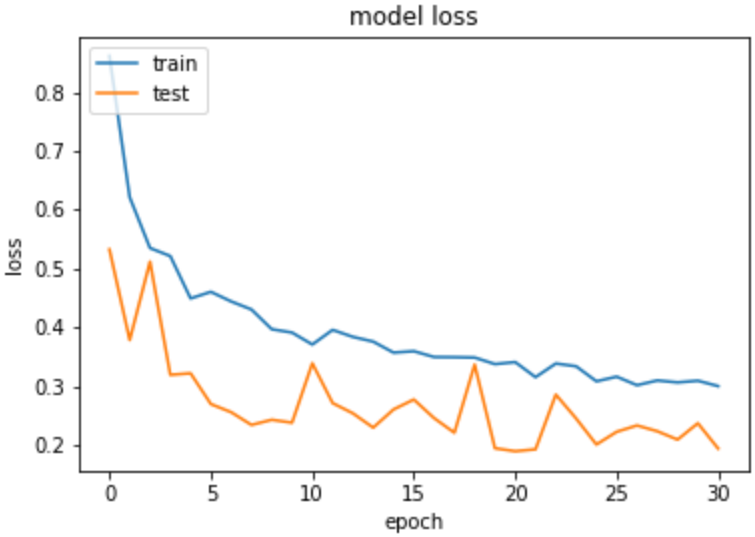
Initial Training:




model loss
train
test
loss
epoch
1.4
1.2
1.0
0.8
0.6
0.4
0
10
20
30
40
50
60
70

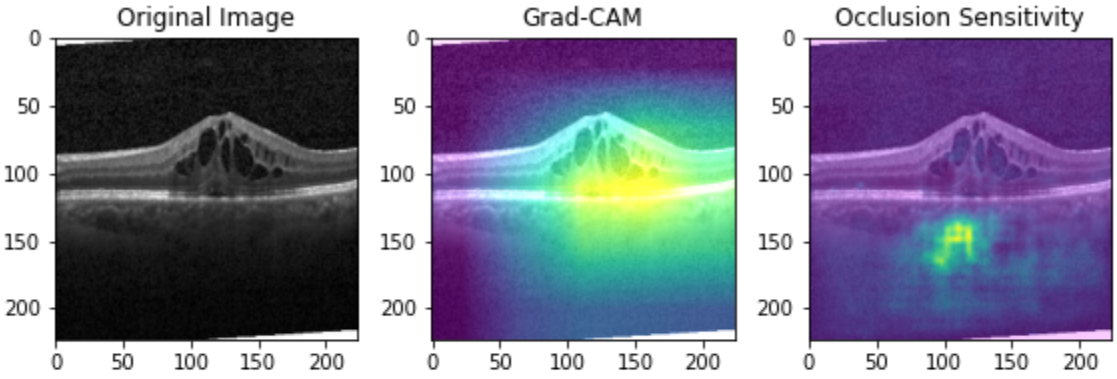
Finetuning:


model accuracy
train
test
accuracy
epoch
0.95
0.90
0.85
0.80
0.75
0.70
0
5
10
15
20
25
30

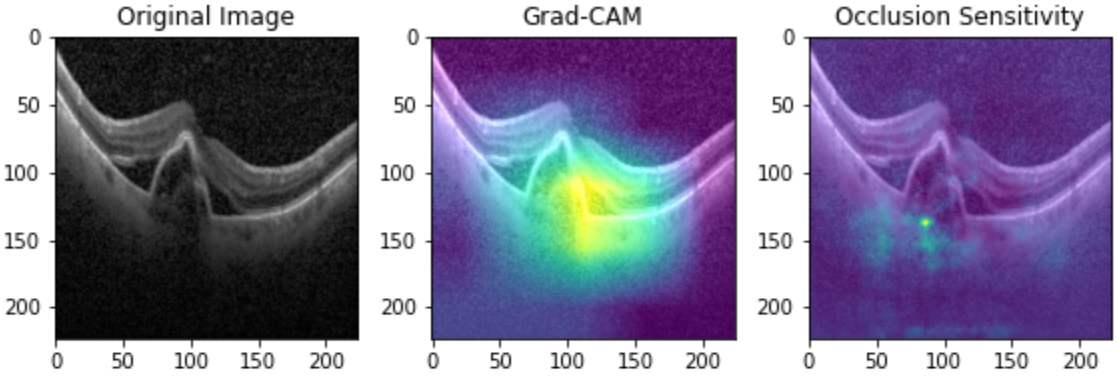


GradCAM (Taken for the same 3 images from the test set):

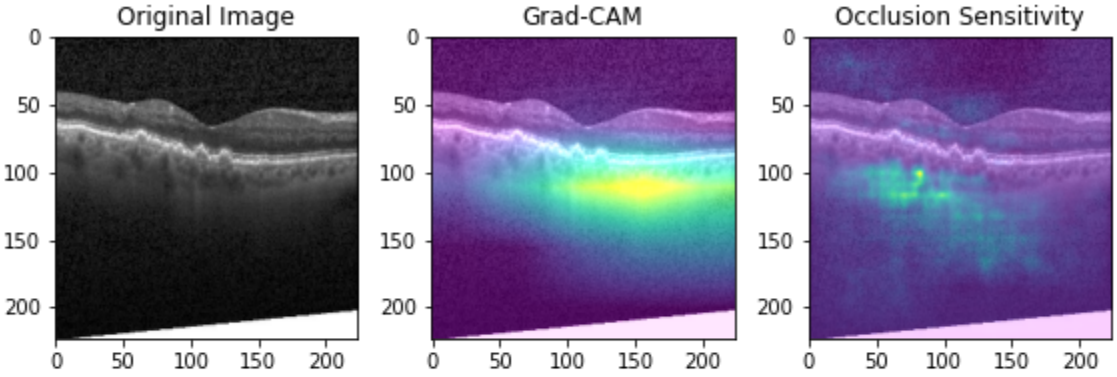
True label: DME, Predicted label: DME, Predicted Accuracy: 0.9998814



True label: CNV, Predicted label: CNV, Predicted Accuracy: 0.9882785

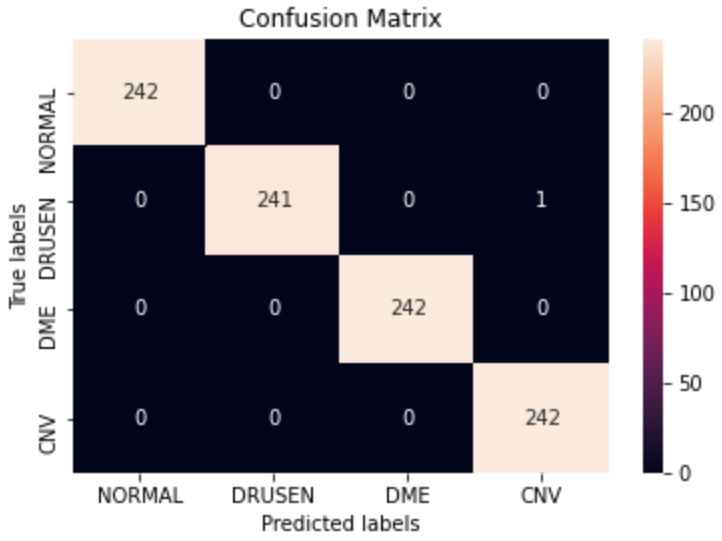


True label: DRUSEN, Predicted label: DRUSEN, Predicted Accuracy: 0.979095



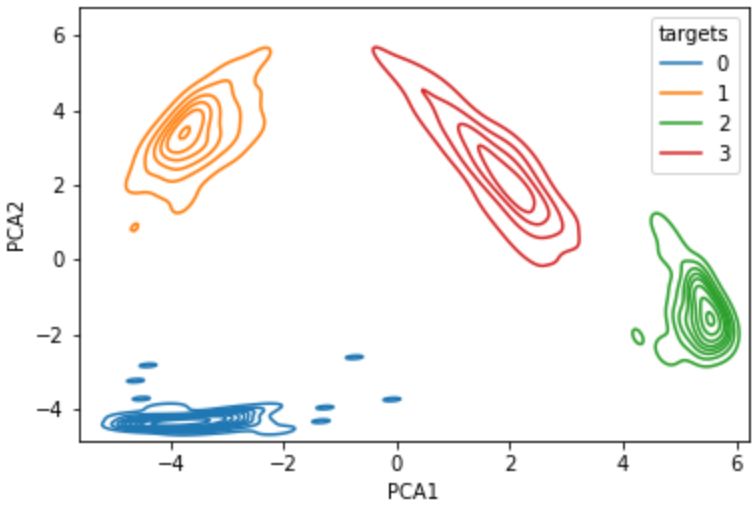
Trained On 98% of the training data and tested on the entire test data

Confusion Matrix:



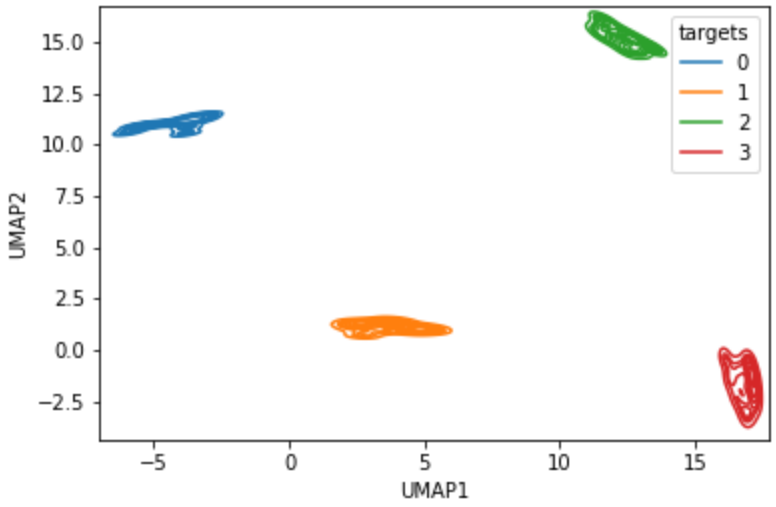
Projections after Dimensionality reduction:

PCA 2D:



PCA 3D:
https://drive.google.com/file/d/1gEIedM6o_WVNth-71871aBXoMR2nw-cZ/view?usp=sharing

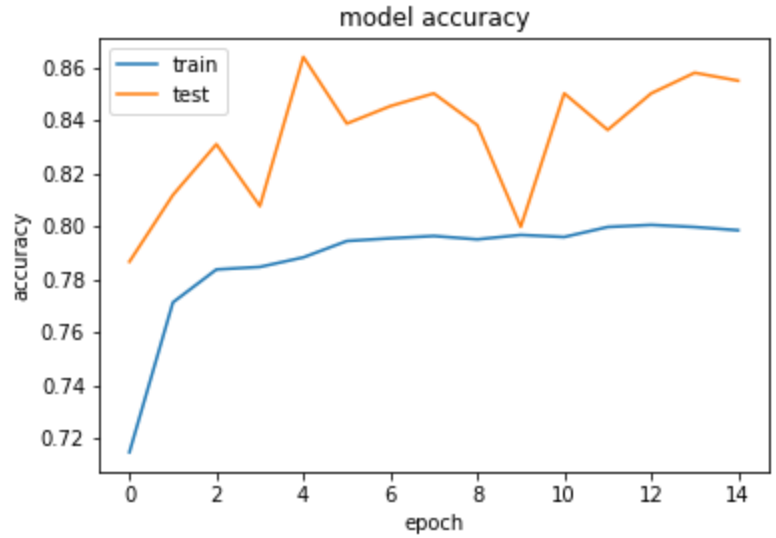
UMAP 2D:

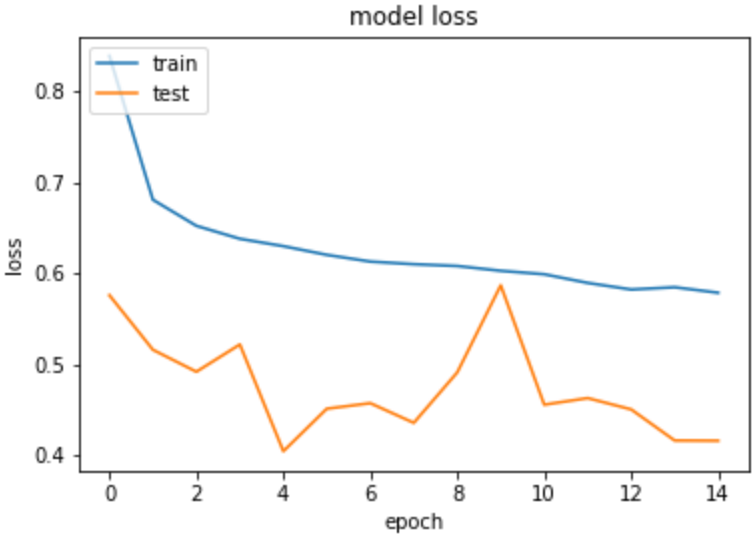


UMAP 3D:
https://drive.google.com/file/d/1-ItF3JGk3iNf7PVhjhSy829pALPbvLZK/view?usp=sharing

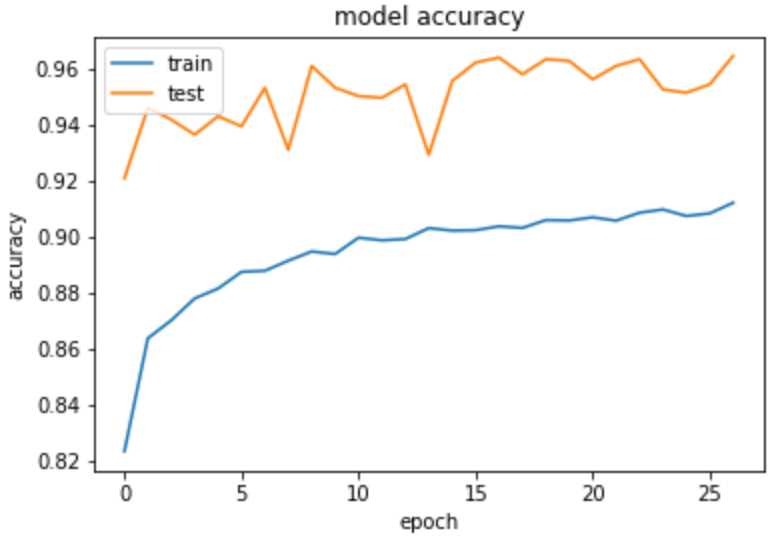
Training History:

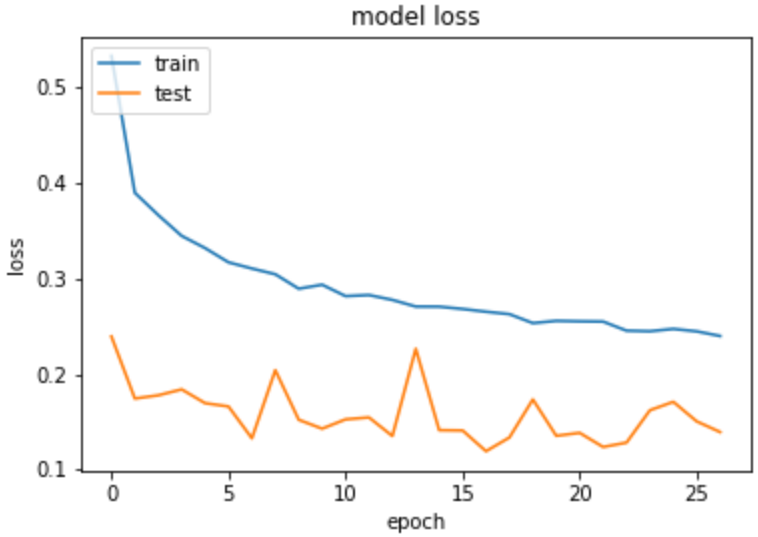
Initial Training:





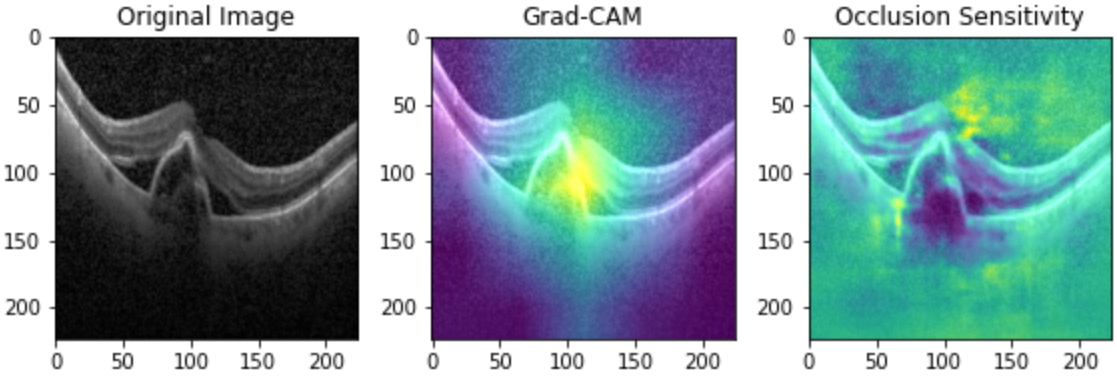
Finetuning:



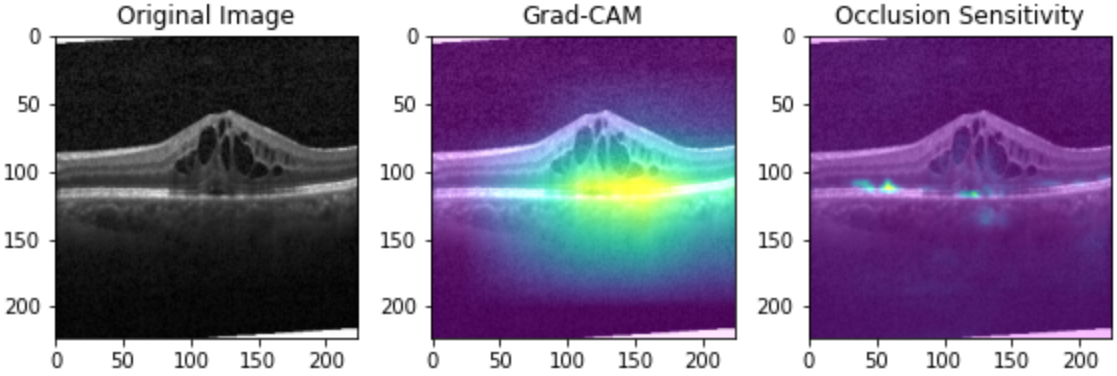


GradCAM (Taken for the same 3 images from the test set):

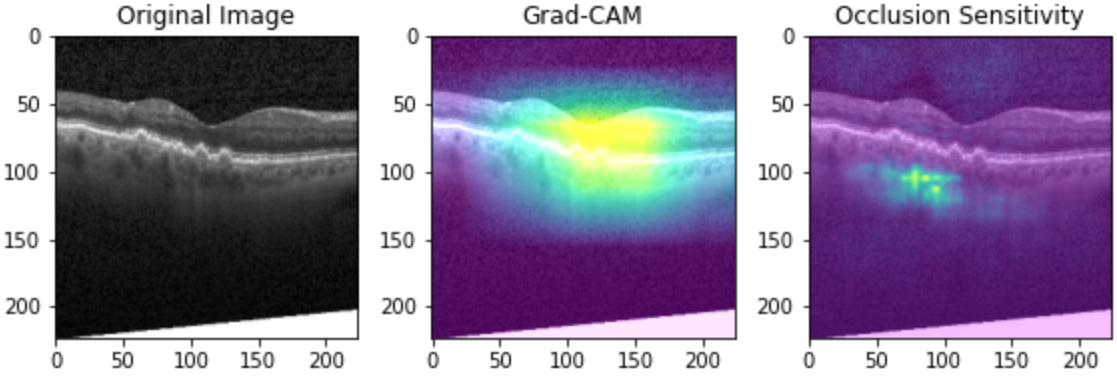
True label: CNV, Predicted label: CNV, Predicted Accuracy: 0.9846581



True label: DME, Predicted label: DME, Predicted Accuracy: 0.99999905

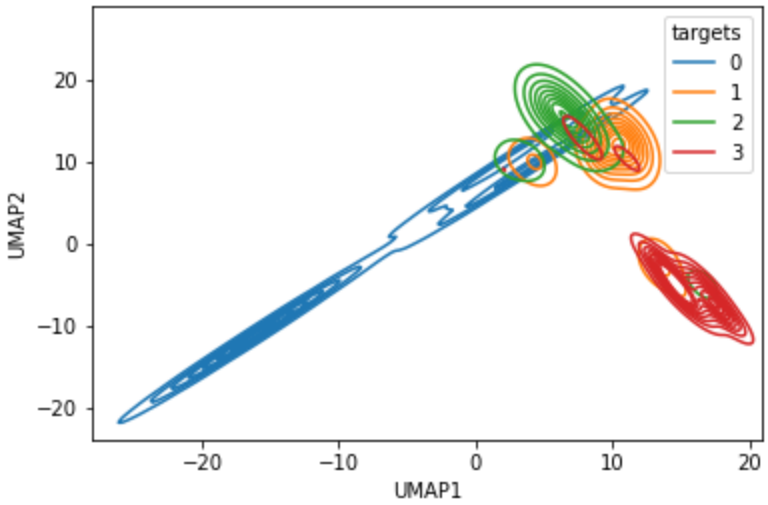


True label: DRUSEN, Predicted label: DRUSEN, Predicted Accuracy: 0.9893147

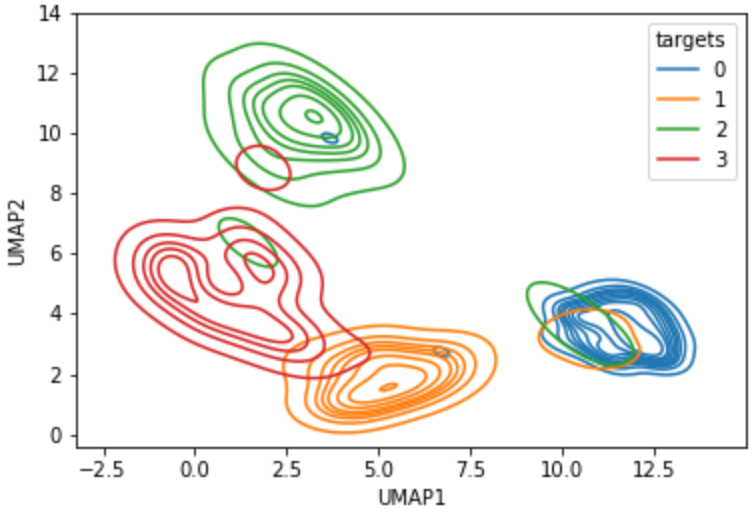


Observations:

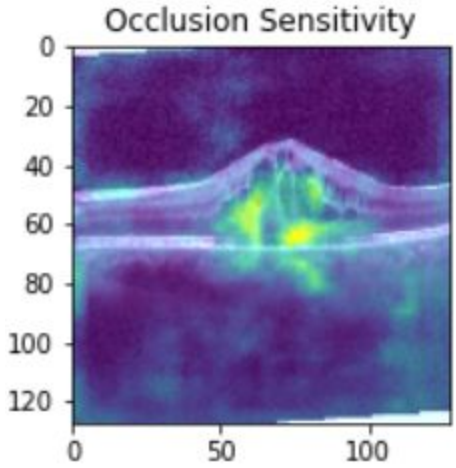
- An increase in the labeled data available for training is associated with an increase separability of projections and better resolution within the clusters:
  UMAP projections for 1% of the training data:



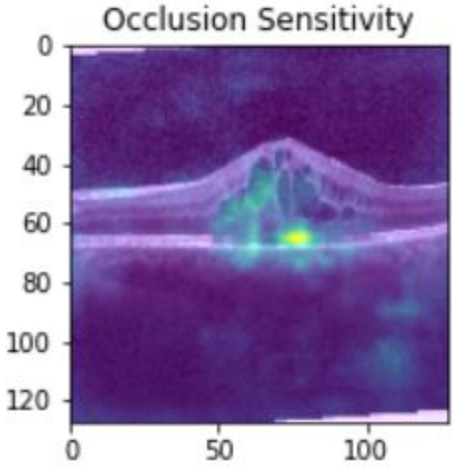
  UMAP projections for 98% of the training data:



- The robustness of the model is strongly correlated with an increase in the training images.
  Occlusion sensitivity from the model trained on 1% of the datasets:



GradCAM activations from the model trained on 98% of the datasets:



Clustering and Label Propagation:

Outline of the actual algorithm:

- Extract the embedding in the Resnet Output from the pretrained SimCLR model which has the shape of [batch_size, 4, 4, 512]
- A global 2D averaging is applied to obtain a [batch_size, 512] vector
- PCA and UMAP is used to reduce the dimension of the 512D vector to 8D vector (arbitrarily chosen 8)
- Apply clustering algorithm on the reduced emdebbing

- KMeans with 4 components and10 different initialization (n_init=10), the best one is chosen
    - Select the outlier based on the silhouette scores (sklearn.metrics.silhouette_samples — scikit-learn 0.24.1 documentation)
    - The Silhouette Coefficient is calculated using the mean intra-cluster distance (`a`) and the mean nearest-cluster distance (`b`) for each sample. The Silhouette Coefficient for a sample is `(b - a) / max(a, b)`. Note that Silhouette Coefficient is only defined if number of labels is 2 `<= n_labels <= n_samples - 1`.
    - Hyperparameters silhouette scores threshold is chosen
- Gaussian Mixture modelling with 4 components and 10 different initialization
    - Initialize with KMeans clustering results
    - And further estimate the covariance matrix of each cluster
    - Z-score can be estimated by $diag\sqrt{(x-\mu)^T\Sigma^{-1}(x-\mu)}$ where μ (cluster center) has the shape of [n_clusters, n_dimension], Σ (covariance matrix) has the shape of [n_dimension, n_dimension]
    - For each given sample/ image, it corresponds to a respective z-score on each cluster, i.e. z_score has the shape of [batch_size, 4]
    - Two Approaches to single out outlier:
        - Based on the z-score threshold, we will select the members that are ill-represented by any of the GMM clusters, the union of dataset that has z score greater than some threshold for all 4 clusters
          i,e, z_score[:,0] > threshold U z_score[:,1] > threshold U z_score[:,2] > threshold U z_score[:,3] > threshold
        - Another approach is based on the prediction probability. GMM makes an estimate of the probability of each sample belonging to a particular cluster. For samples that do not explicitly belonging to one class, the probability would be more evenly spread out over the 4 clusters. Those samples are extracted for further finetuning.
- Label Propagation
    - Once the “outliers” are selected, label propagation is run through the embeddings of the selected outlier.
      Only one labeled image per class is fed to the label spreading algorithm per run, i.e. 4 labeled images every time. We run this for 20 times with different labeled images (altogether requiring a total of initially labeled image of 20*4 = 80), and single out images from the “outlier” that is agreed upon with less than 10 different runs out of the 20 runs performed.
    - These are then fed to finetune the Resnet pretrained on Contrastive learning objective with an attached classification head.

Selecting Data Based on KMeans:

- Filter data with silhouette score less than 0.1

Gaussian Mixture model:

- Selecting outlier with z-score > 3 for all 4 clusters
- And a maximum vote of 10 out of 20 runs during label smoothing

The finetuned model is composed of the resnet trained on contrastive loss objective function, a global 2D averaging operation, and a dense layer for classification. The last 2 layers of the last residual block in the resnet-18 model is unfreezed for further finetuning.

Data selection	Normal	Drusen	DME	CNV	Total
KMeans Silhouette score < 0.1	1390	1970	1836	9071	14267
GMMOutlier (z > 3)	2896	682	1404	5068	10060
Gaussian Mixture with prediction probability between (0.4-0.6)	317	197	216	512	1242

	Accuracy	Precision	Recall	F1-score
KMeans	0.93	0.94	0.93	0.93
GMMOutlier (z > 3)	0.92	0.92	0.91	0.91
GMM uncertain (0.4<p<0.6)	0.91	0.91	0.91	0.91

Updates (2/19/2021)

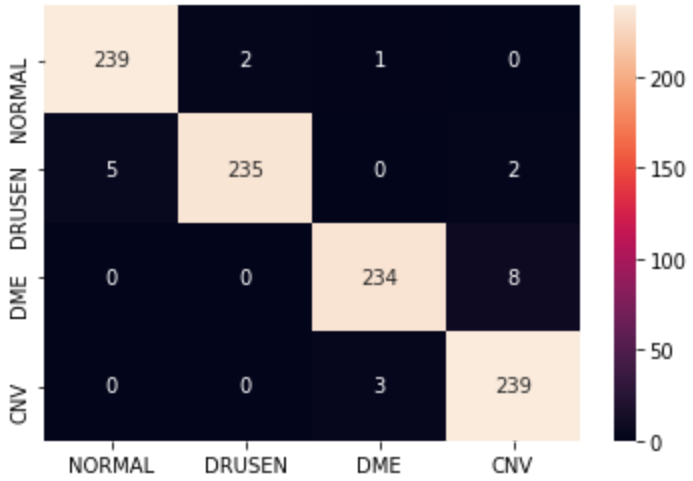
Gaussian Mixture Model (0.4 < p < 0.6)
5 runs: global average test accuracy:

The SimCLR model is trained on 224 image full training dataset
https://wandb.ai/hisunnytang/OCT-keras-SimCLR/runs/3ap2gynt?workspace=user-hisunnytang
It has a contrastive accuracy of ~ 99 percent. (99 percent of the time it is able to identify its augmented copy).

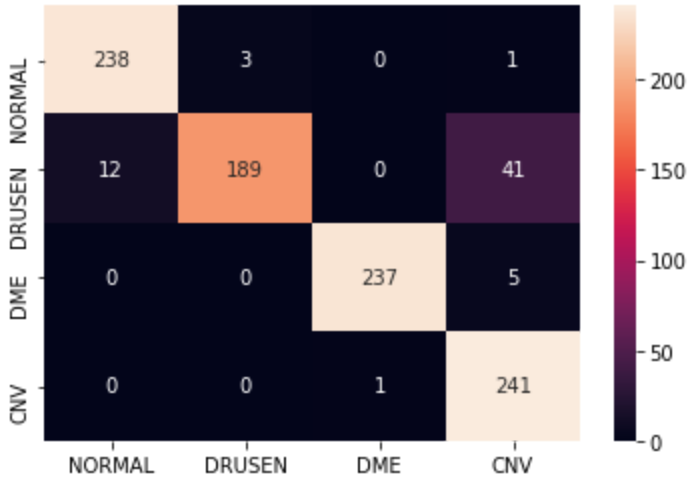
The pretrained resnet on contrastive loss is attached with a global average pooling 2D to give a 512 vector and then fed into a dense layer for softmax classification.

Uncertain Samples Selected by Label Propagation	Normal	Drusen	DME	CNV	Total samples selectively annotated	Fraction of Total Training Data	Average Test Accuracy
1 st run	770	331	273	483	1837	0.022	0.943
2 nd run	792	304	297	632	2025	0.024	0.979
3 rd run	571	226	225	473	1495	0.018	0.9345
4 th run	729	306	269	535	1839	0.022	0.9643
5 th run	692	202	235	209	1338	0.016	0.9718
Average	710.8+-78	273.8+-50.32	260 +-26	466+-141	1707+- 251	0.02 +-0.003	0.9518+-0.017

Weighted	Precision	Recall	F1-Score
1 st run	0.98	0.97	0.97
2 nd run	0.97	0.97	0.97
3 rd run	0.95	0.94	0.94
4 th run	0.98	0.98	0.98
5 th run	0.94	0.93	0.93



Best Performing Finetuned model on test set data



Worst Performing Model on test set data

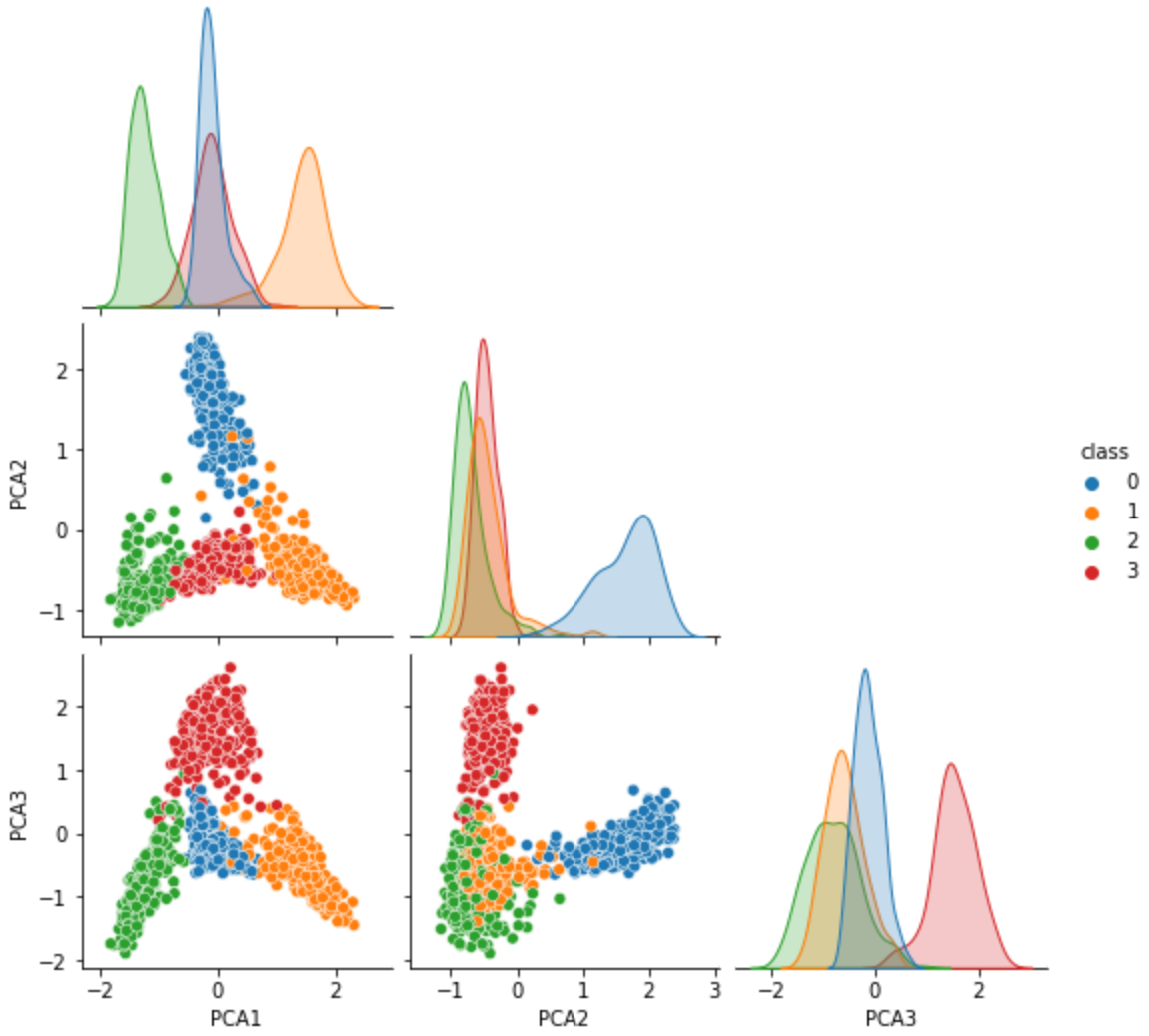
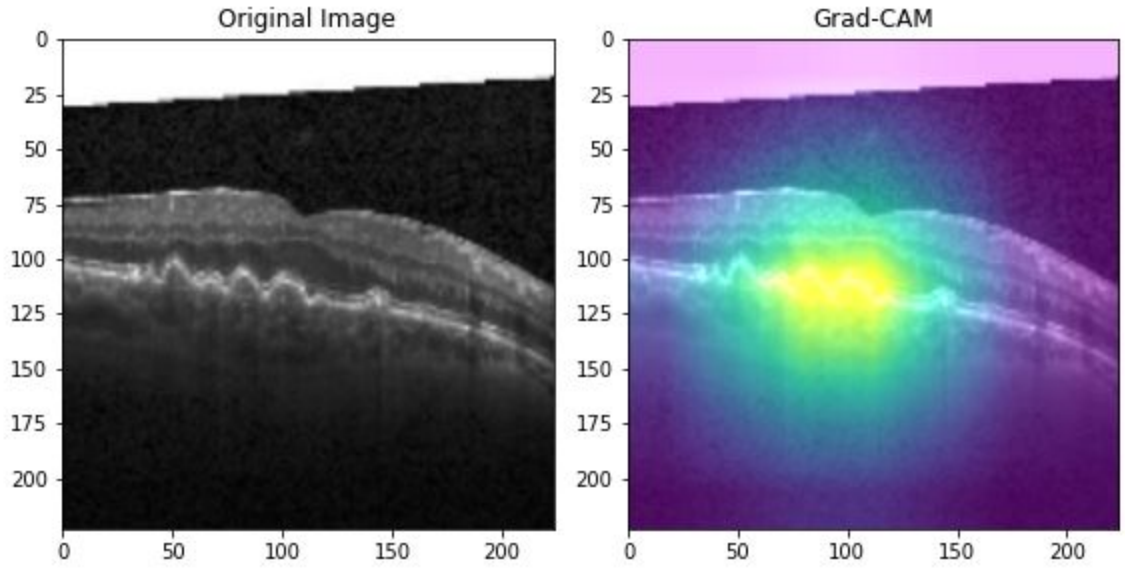
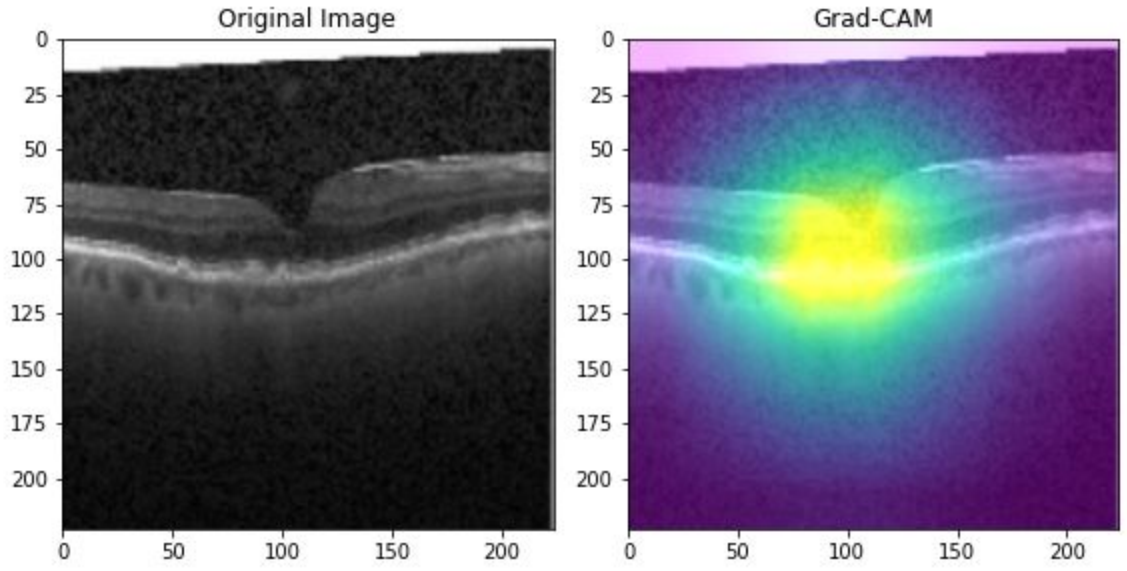


Figure: PCA reduction of the resnet output layer after fine tuning on the test set data.

True label: DRUSEN, Predicted label: CNV, Predicted Accuracy: 0.48011595



rue label: DRUSEN, Predicted label: NORMAL, Predicted Accuracy: 0.7212102



Saliency Maps of the FULL Test dataset can be found in the zip file `images_simclr.zip`, and the model weights too.

https://drive.google.com/drive/folders/14Xn5gqDN1TLSH_9i-yQoXe8VzZE8vvAG?usp=sharing