Deep Learning Model for Eye Disease Detection

Problem statement

Health Promotion Board is searching for innovative, cutting-edge technological **solutions to facilitate mass eye screening** of common eye diseases in the general adult population.

They hope to **automate interpretation of retina screening images**, shorten the time taken to flag high-risk persons for further health assessment.

This would **allow early intervention,** lower risk of disease progression and **lower healthcare cost burden**. By minimising number of people in the population with severe eye disease, we minimise the use of more costly therapies.

Possible stakeholders: public health authorities, eye clinics, optical shops

My proposed solution

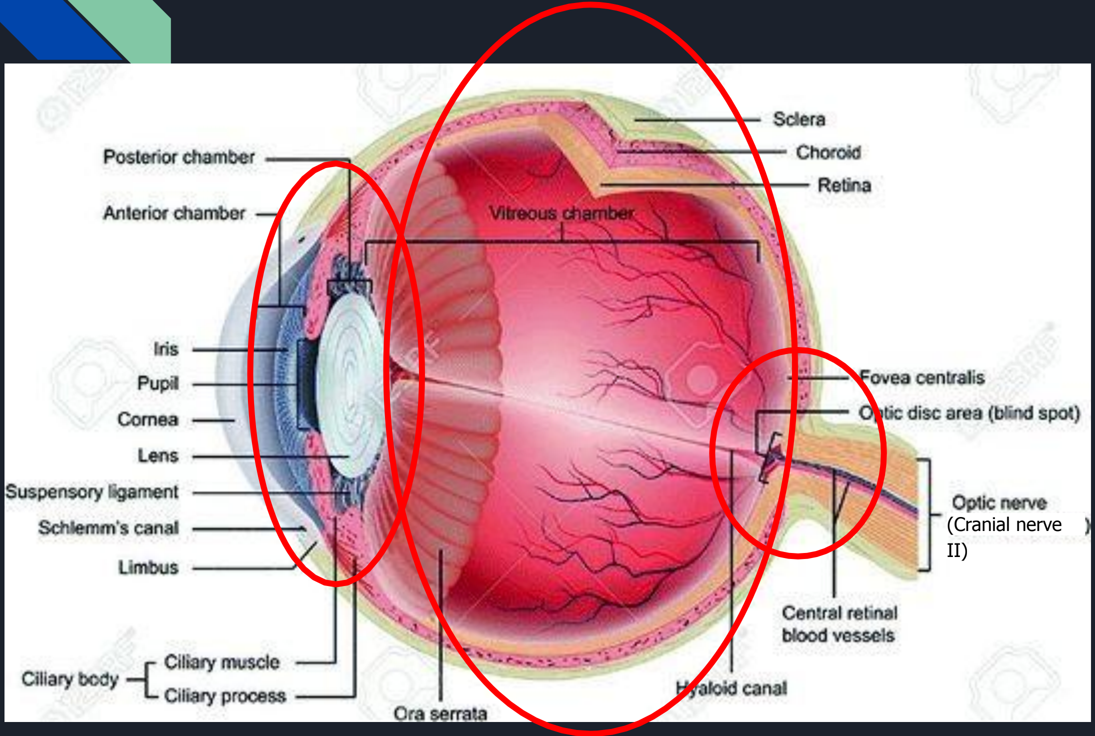
Using 4,217 retinal images collected from various sources, I constructed a machine learning algorithm to help classify a RGB retina image to any of 3 classes of common adult eye diseases and a **normal** control group:

- Cataract
- Diabetic retinopathy
- Glaucoma
- normal

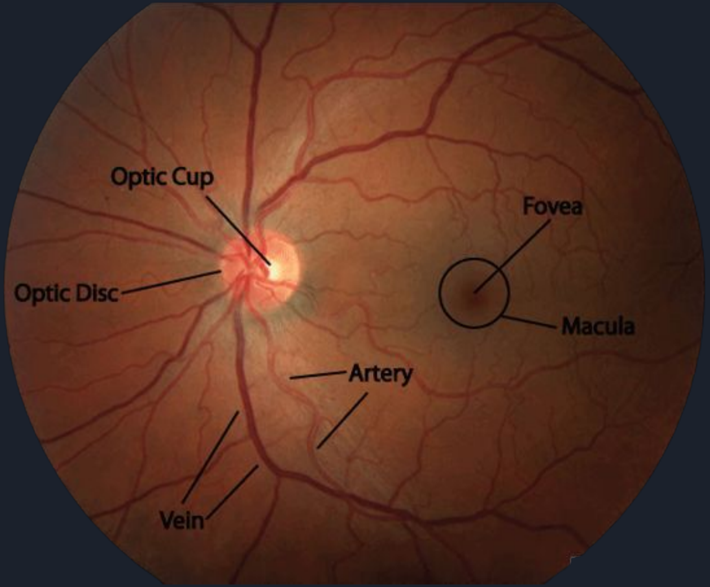
What do we want the model to look for in the images?

Terminology

Basic eye anatomy

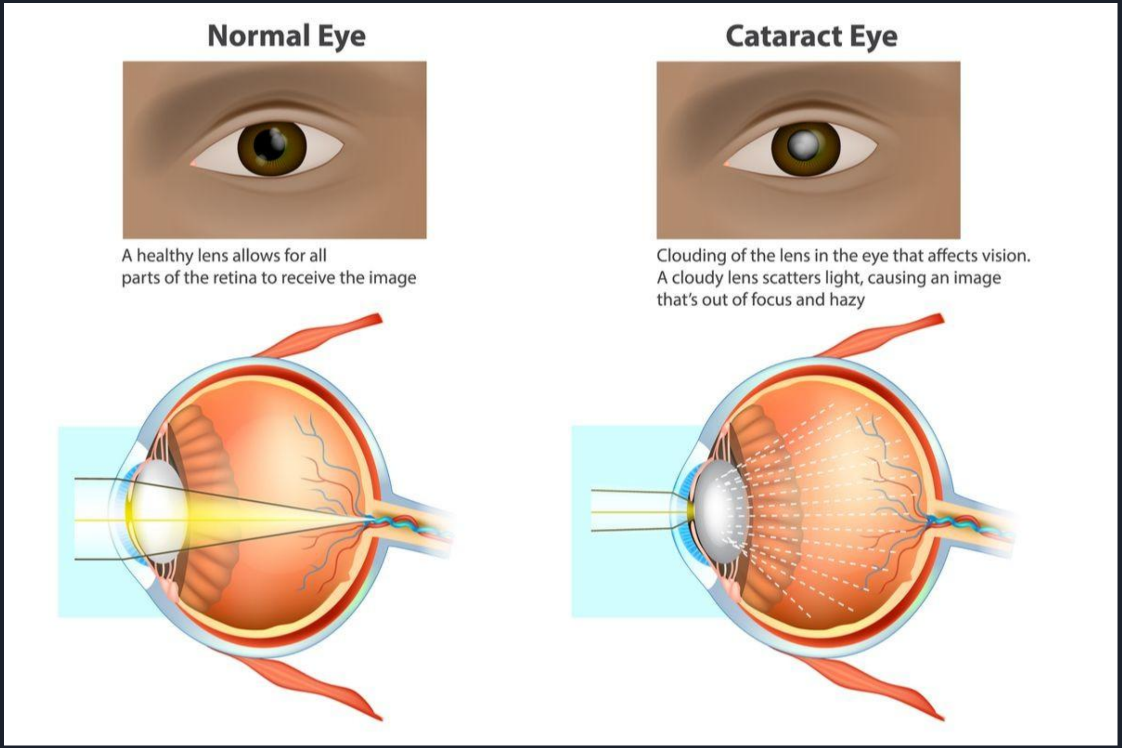


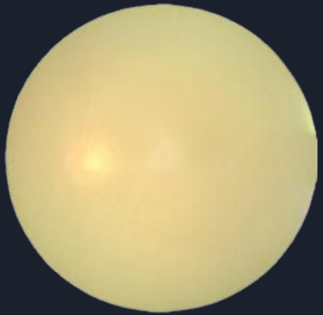
Credit: Springer



Credit: stanfordmedicine25.stanford.edu

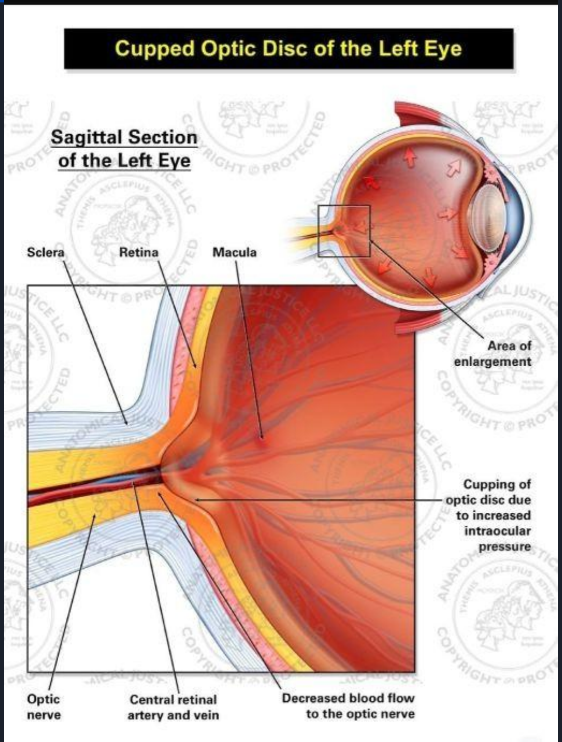
Cataract





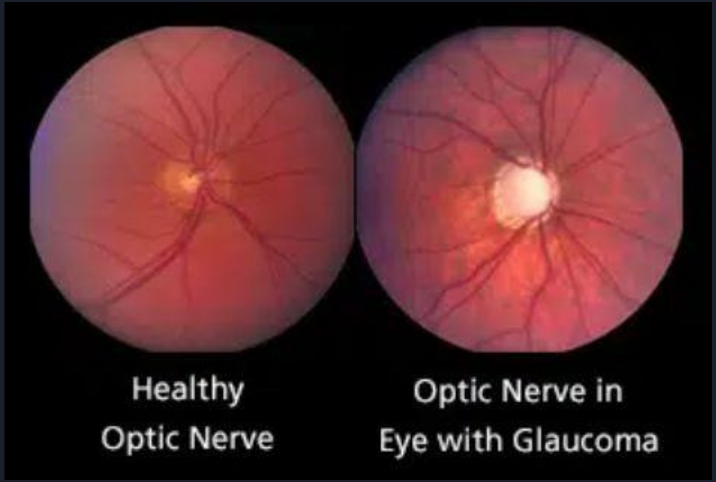
Credit: nvisioncenters.com

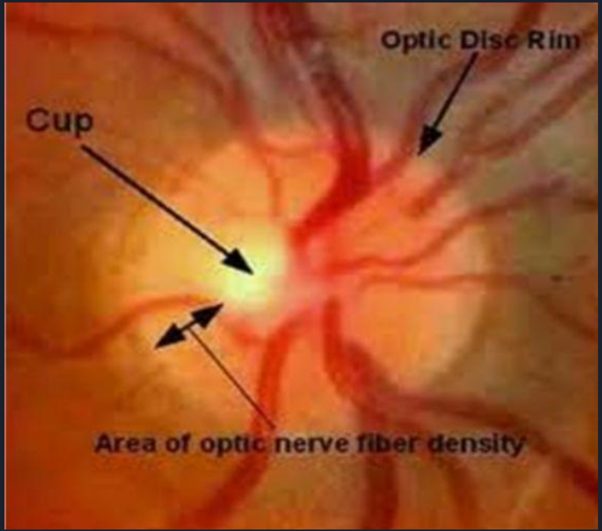
Glaucoma



Credit: anatomicaljustice.com

Credit: glaucoma.org

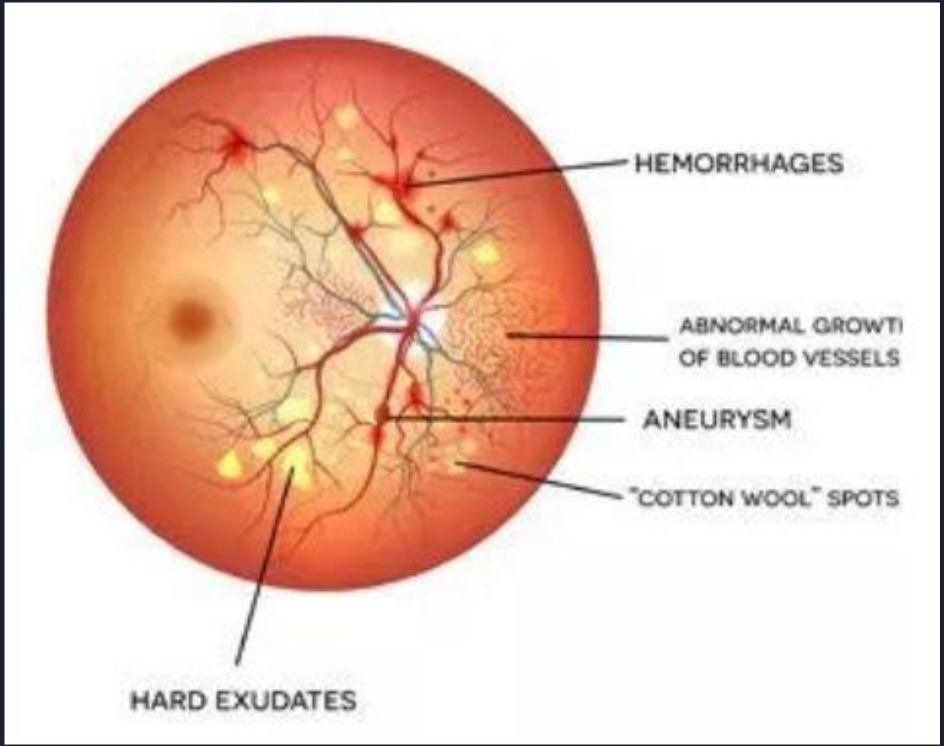




- cup-to-disc ratio > 0.5 = abnormal
- physiologic limit of 0.5

Credit: intechopen.com

Diabetic retinopathy



Back to Project

Image dataset

- 4,217 images
  - Cataract: 1038
  - Diabetic retinopathy: 1098
  - Glaucoma: 1007
  - Normal: 1074
- Attributes:
  - Variety of sizes
  - RGB
  - *.png, *.jpg, *.bmp
- Various sources

Image hashing

- 4215 unique images

Train / Validation / Hold-out:

- 0.5 / 0.25 / 0.25

Run base models

Metrics

- Categorical accuracy
- Precision
- Recall
- AUC
- F1 score (custom)

Optimizer = Adam
Loss = Categorical

crossentropy

Models:

- Sequential Model – CNN

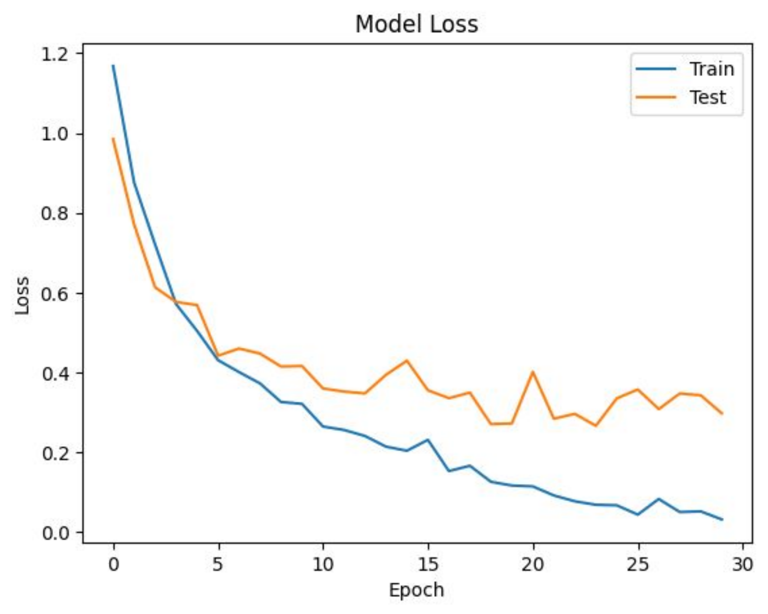
Weights

- ImageNet as reference
- Trainable = True

Added top layers:

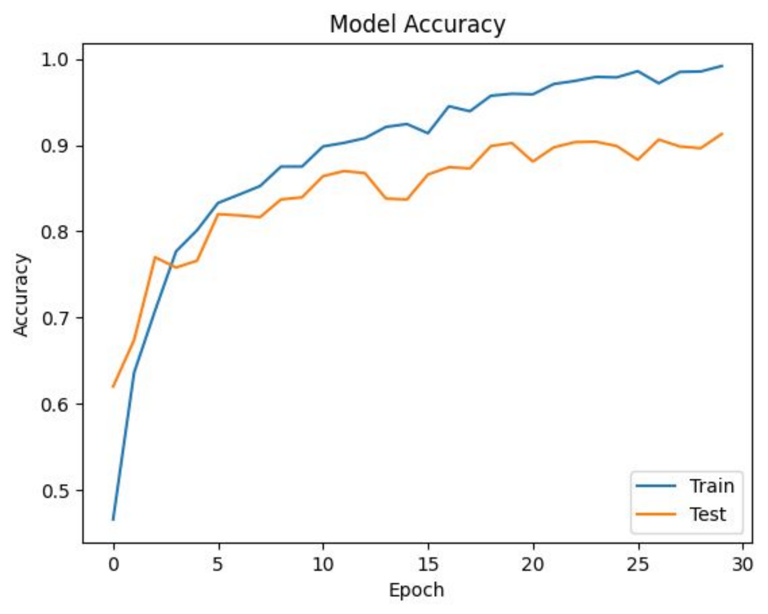
1. BatchNormalization
2. Dense (regularization)
3. Dropout (0.4)
4. Dense (4, softmax)

Model Loss vs Epoch



Model Loss
Train
Test
Loss
Epoch
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20
25
30

Model Accuracy vs Epoch



Model Accuracy
Train
Test
Accuracy
Epoch
0.5
0.6
0.7
0.8
0.9
1.0
0
5
10
15
20
25
30

Run models with augmentation layers

Added

- RandomFlip('horizontal')
- RandomRotation(0.1)
- RandomContrast(0.1)

Weights

- ImageNet as reference
- Trainable = True

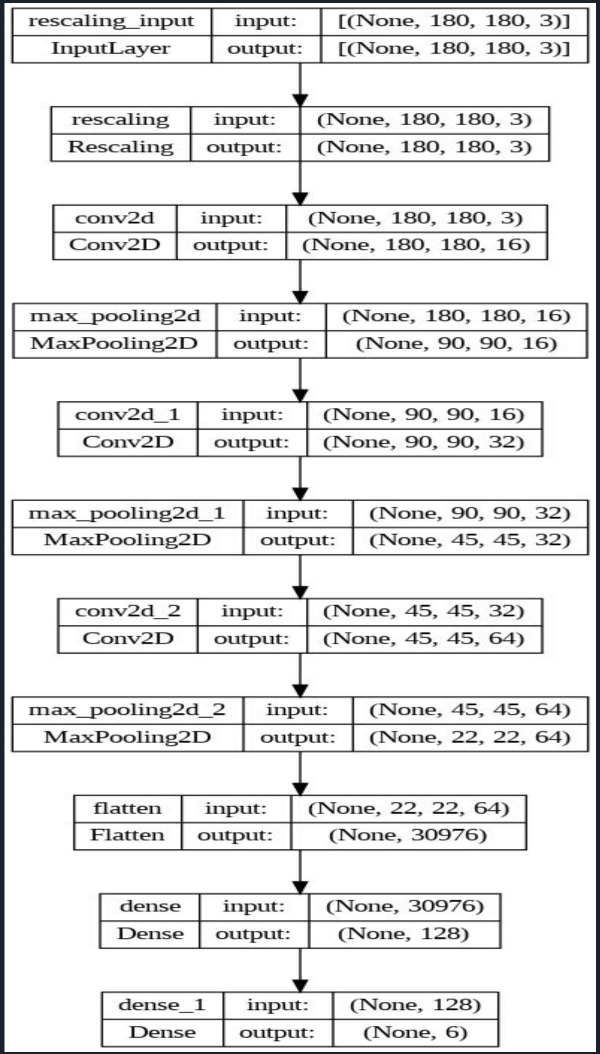
Same metrics

- Categorical accuracy
- Precision
- Recall
- AUC
- F1 score (custom)

Optimizer = Adamax

Loss = Categorical crossentropy

Model Plot


rescaling_input
InputLayer
input:
output:
[(None, 180, 180, 3)]
[(None, 180, 180, 3)]
rescaling
Rescaling
input:
output:
(None, 180, 180, 3)
(None, 180, 180, 3)
conv2d
Conv2D
input:
output:
(None, 180, 180, 3)
(None, 180, 180, 16)
max_pooling2d
MaxPooling2D
input:
output:
(None, 180, 180, 16)
(None, 90, 90, 16)
conv2d_1
Conv2D
input:
output:
(None, 90, 90, 16)
(None, 90, 90, 32)
max_pooling2d_1
MaxPooling2D
input:
output:
(None, 90, 90, 32)
(None, 45, 45, 32)
conv2d_2
Conv2D
input:
output:
(None, 45, 45, 32)
(None, 45, 45, 64)
max_pooling2d_2
MaxPooling2D
input:
output:
(None, 45, 45, 64)
(None, 22, 22, 64)
flatten
Flatten
input:
output:
(None, 22, 22, 64)
(None, 30976)
dense
Dense
input:
output:
(None, 30976)
(None, 128)
dense_1
Dense
input:
output:
(None, 128)
(None, 6)

Model Summary

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling (Rescaling)       (None, 180, 180, 3)       0

 conv2d (Conv2D)             (None, 180, 180, 16)      448

 max_pooling2d (MaxPooling2  (None, 90, 90, 16)        0
 D)

 conv2d_1 (Conv2D)           (None, 90, 90, 32)        4640

 max_pooling2d_1 (MaxPoolin  (None, 45, 45, 32)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 45, 45, 64)        18496

 max_pooling2d_2 (MaxPoolin  (None, 22, 22, 64)        0
 g2D)

 flatten (Flatten)           (None, 30976)             0

 dense (Dense)               (None, 128)               3965056

 dense_1 (Dense)             (None, 6)                 774

=================================================================
Total params: 3989414 (15.22 MB)
Trainable params: 3989414 (15.22 MB)
Non-trainable params: 0 (0.00 Byte)
```

Class-level Accuracy

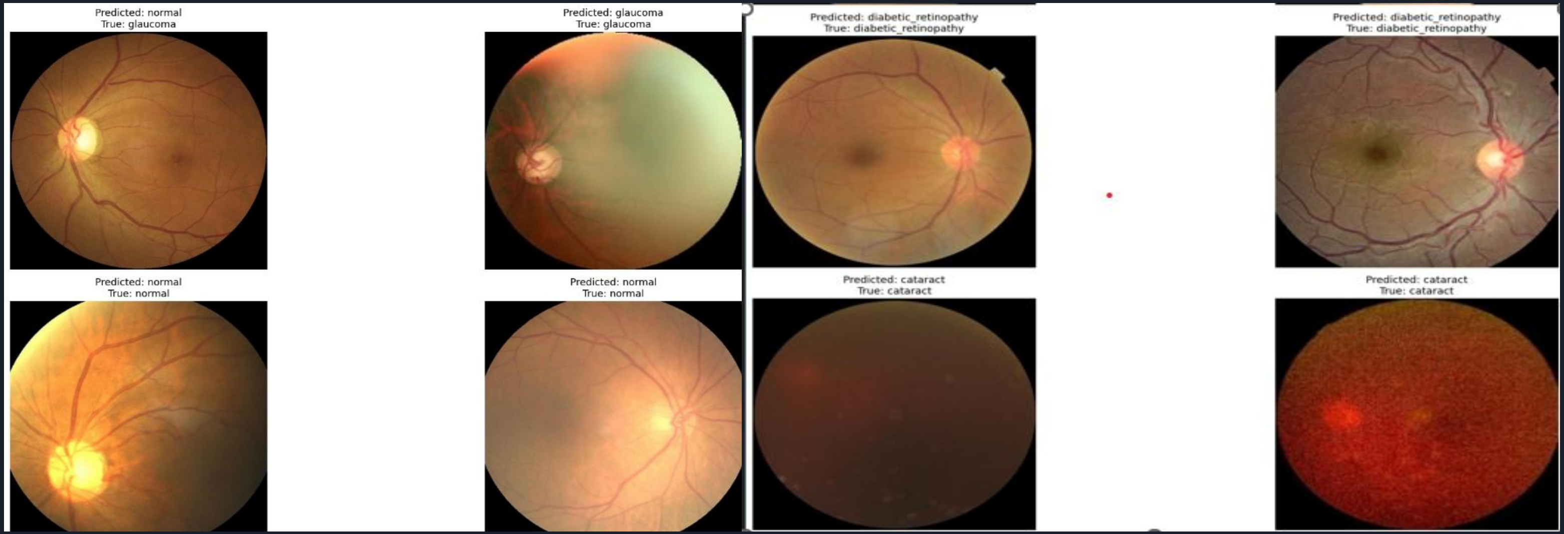
```
Class-level accuracy:
cataract: 0.9188
diabetic_retinopathy: 0.9724
glaucoma: 0.8887
normal: 0.8702
```

Accuracy & Loss

1. Training and Validation Loss
2. Training and Validation Accuracy

loss	accuracy	val_loss	val_accuracy
0.043694	0.985833	0.357252	0.8830
0.082823	0.971833	0.308056	0.9065
0.050213	0.985000	0.347222	0.8985
0.051818	0.985500	0.342703	0.8965
0.031641	0.991833	0.297583	0.9130

Predicted Labels



Limitations

- Model is restricted to identifying 1 disease class per image. Model is unable to identify multiple diseases if multiple abnormalities are present in retinal image
- Images with Low quality, Low differentiating features from normal may be wrongly classified
- Diabetic retinopathy abnormalities are not giving the strongest signals to the model, meaning images outside of dataset with diabetic retinopathy may be wrongly classified

Conclusions

- CNN base model with augmentation layers is chosen for deployment with 91% accuracy.
- This model is best used with other screening modalities to increase precision and sensitivity of diagnosis e.g.
    - tonometer (for anterior chamber pressure) for glaucoma
    - snellen chart (visual acuity) for overall vision ability
    - auto perimeter visual field analyser (glaucoma)
    - Optical coherence tomography (OCT) - (study retina layers, and subretinal deposits)