Blindness Detection

Daniela Matinho | Hanna Kerr | Yuling Gu

Machine Learning
Fall 2019

AGENDA

- BLINDNESS DETECTION
- PROBLEM STATEMENT
- DATA PREPROCESSING
- MACHINE LEARNING TECHNIQUES:
  - PCA
  - LOGISTIC REGRESSION
  - RANDOM FOREST
  - ADA - BOOST
- ARTIFICIAL NEURAL NETWORKS:
  - CONVOLUTIONAL NEURAL NETWORKS
  - TRANSFER LEARNING
- FINDINGS & FUTURE WORK

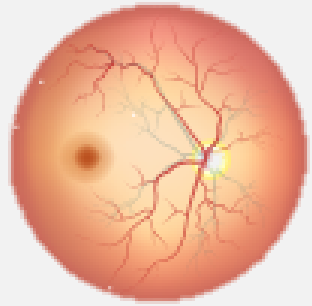
STAGES OF DIABETIC RETINOPATHY

Diabetic retinopathy is the most common cause of vision impairment and blindness. It is caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina).

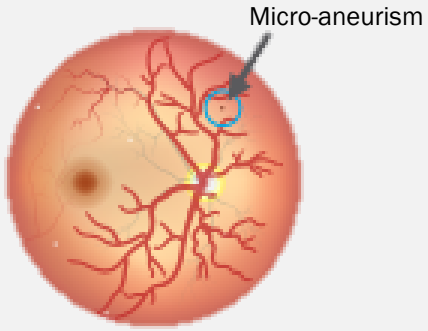
From 2010 to 2050, the number of diabetic retinopathy is expected to nearly double, from 7.7m to 14.6m in the US.

PROBLEM STATEMENT

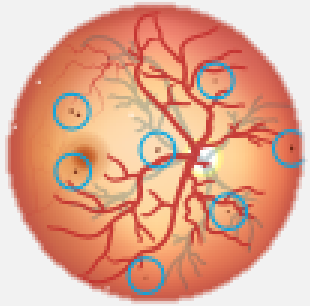
The purpose of this project is to correctly classify the five stages of the disease, given the images of different patients.



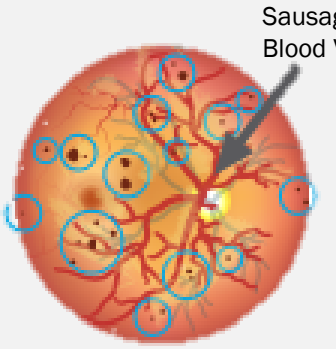
Normal



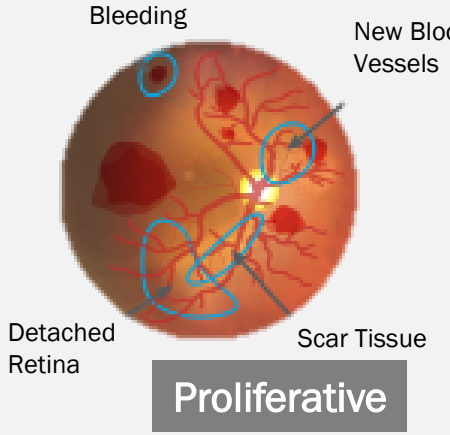
Mild



Moderate



Severe

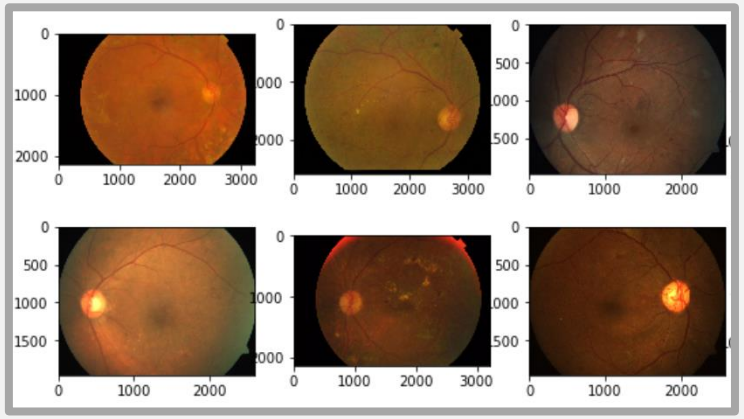


Proliferative

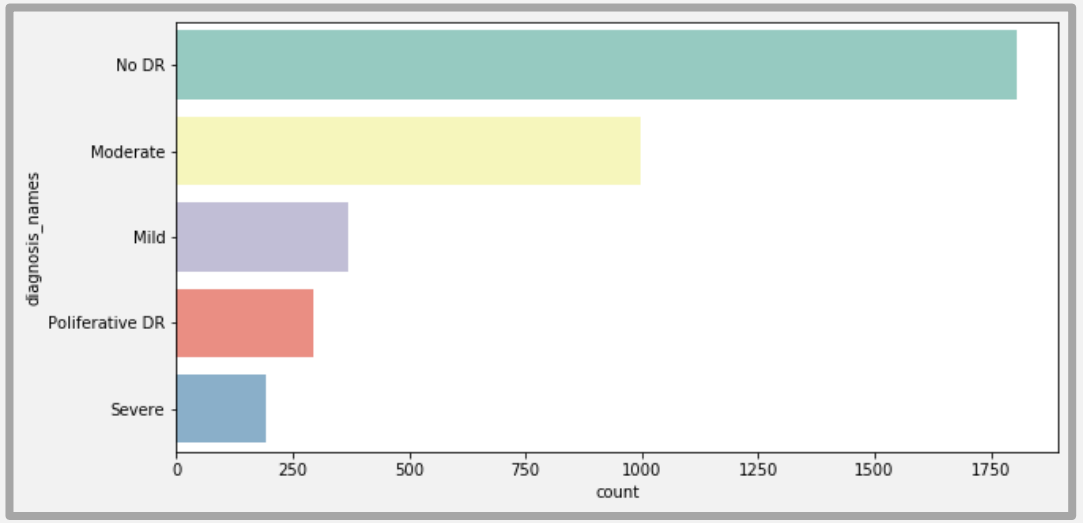
STAGES OF PROJECT

1. DATA COLLECTION

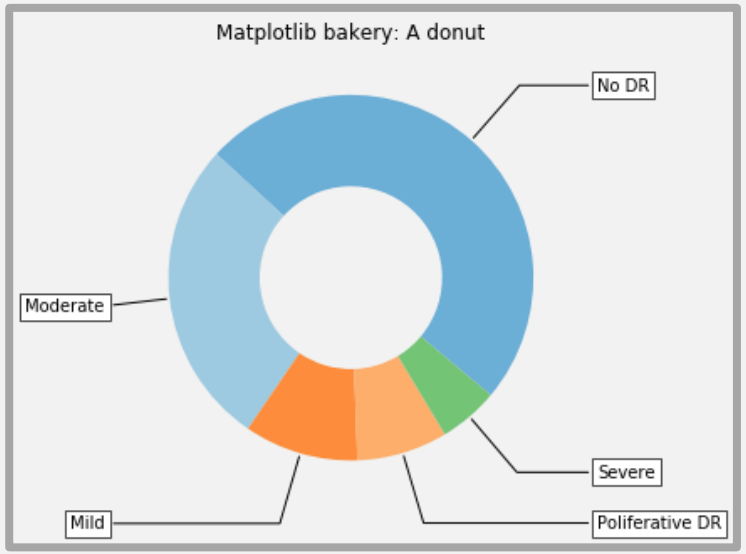
- Total of 3662 images



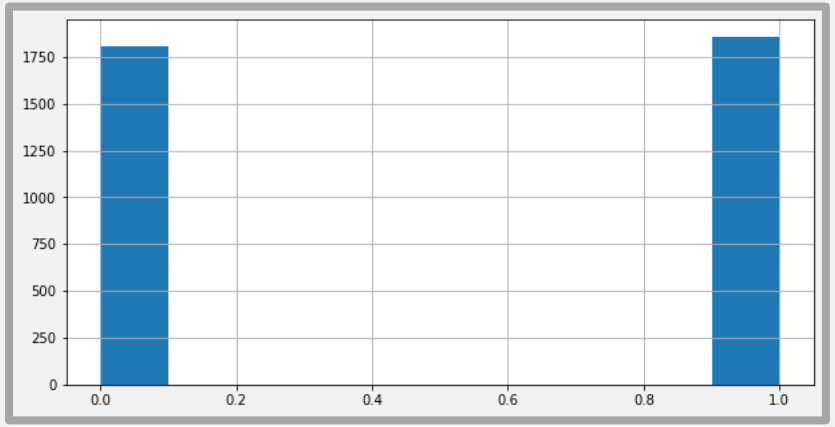
2. EXPLORATORY ANALYSIS



- 1805 images are not classified in the dataset (No DR)

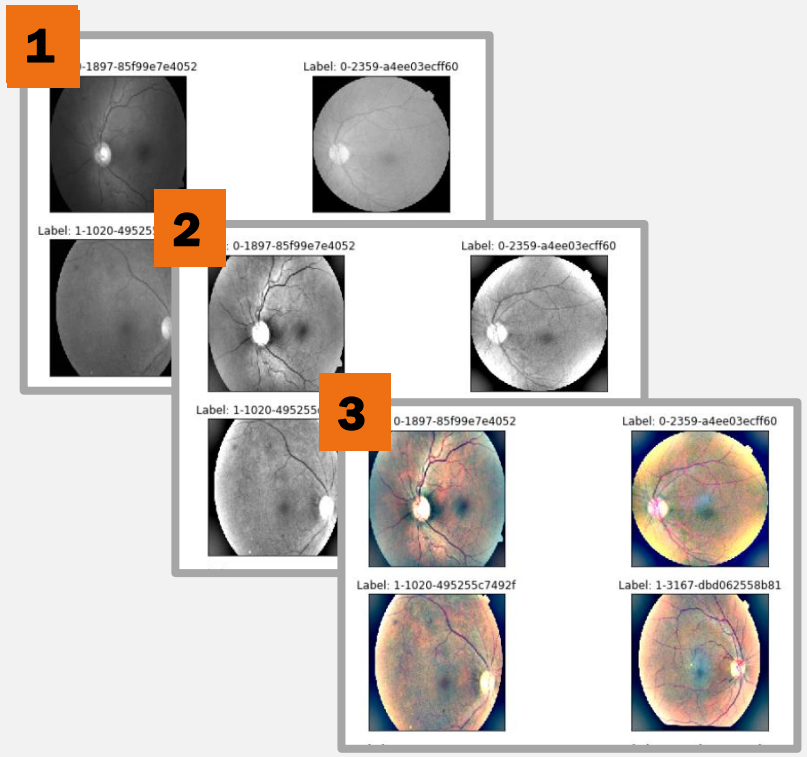


- Data is pretty balance when we look at DR and No DR



STAGES OF PROJECT

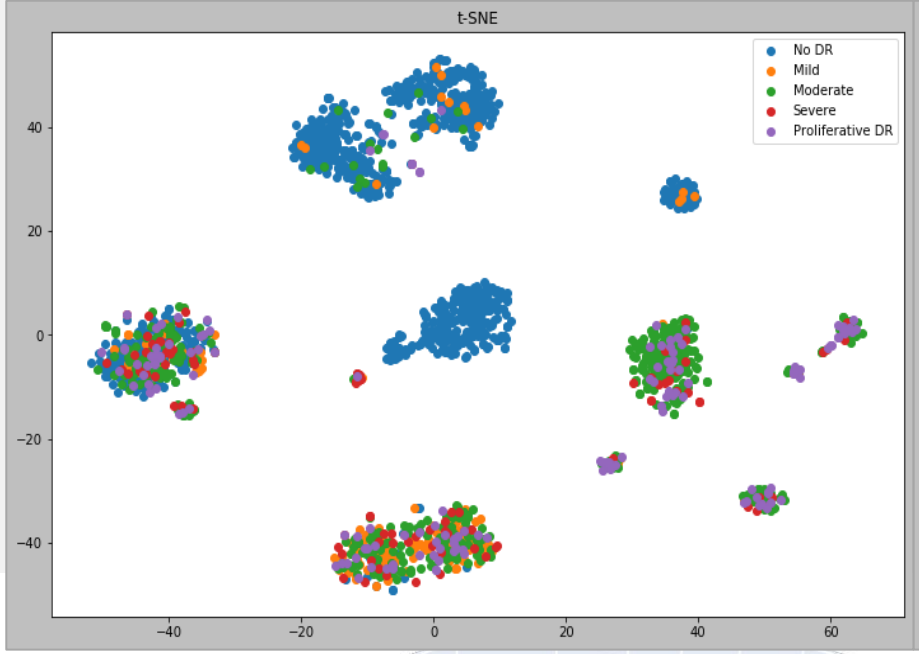
3. FEATURE ENGINEERING

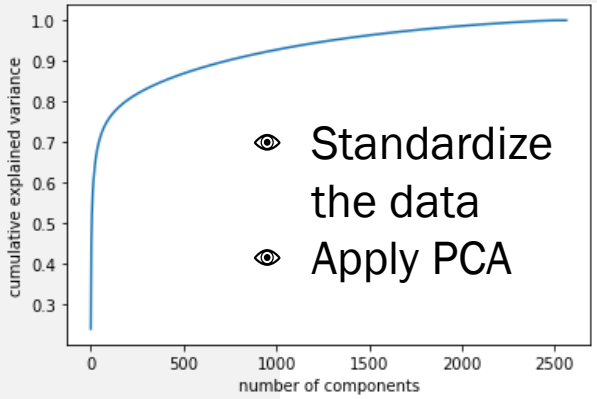


- Reduce components from 750 out 2563 in 3D

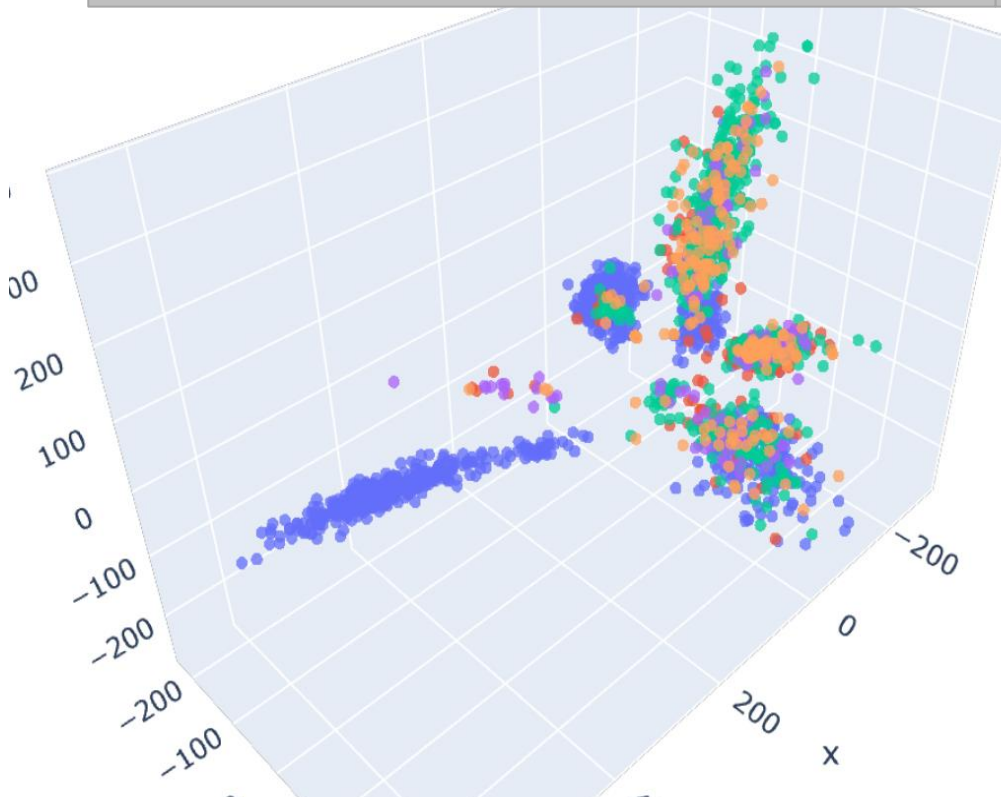
4. T-SNE & PCA

- T-SNE to see the distribution of the data





- Standardize the data
- Apply PCA



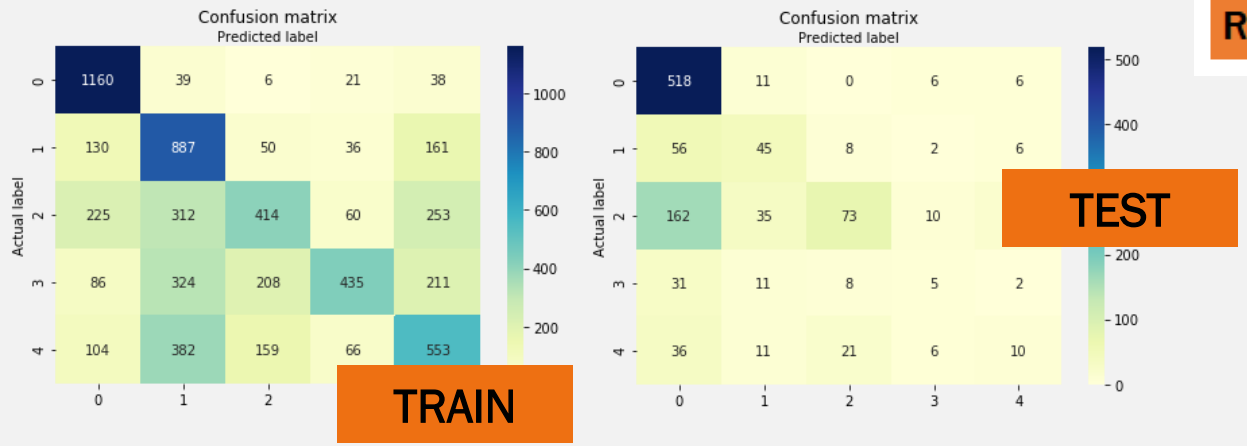
MACHINE LEARNING

1. LOGISTIC REGRESSION CLASSIFICATION
2. RANDOM FOREST CLASSIFICATION
3. ADA BOOST CLASSIFIER – RANDOM SEARCH

- ASSUMPTIONS:
  - Data split: 70/30
  - Flatten data
  - Standardize - Reduce components to750
  - Smote the data

```
Counter({0: 1264, 4: 1264,
         1: 1264, 2: 1264, 3: 1264})
```

  - Fit three machine learning models
  - Tune models



	TRAIN	TEST
Logistic Regression Classification	0.94	0.50
Random Forest Classification	0.55	0.59
Ada Boost Classifier - Random Search	0.66	0.52

- The model has more difficulties in classifying Moderate and No DR classes

CONVOLUTIONAL NEURAL NETWORKS

Input
225x225x3

RELU

RELU

FLATTEN

FULLY CONNECTED

RELU

Softmax

FULLY CONNECTED

0 – No DR

1 – Mild

2 – Moderate

3 – Severe

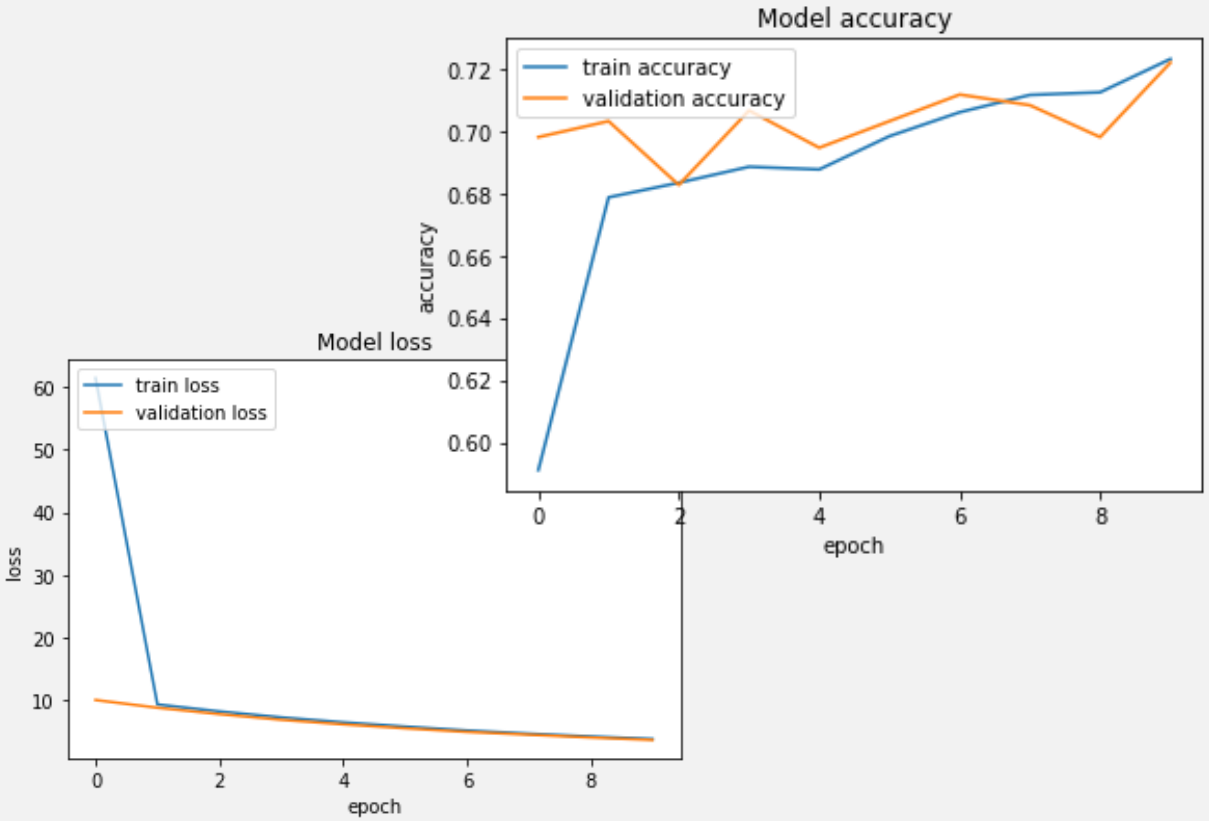
4 – Proliferate

CONVOLUTION | DENSE | MAXPOOLING | PADDING | DROPOUT LAYERS

Total number of layers: 15

CONVOLUTIONAL NEURAL NETWORKS

1. BUILD BASIC CNN



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 225, 225, 32)	2432
max_pooling2d_1 (MaxPooling2	(None, 112, 112, 32)	0
conv2d_2 (Conv2D)	(None, 107, 107, 64)	73792
dropout_1 (Dropout)	(None, 107, 107, 64)	0
conv2d_3 (Conv2D)	(None, 107, 107, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 53, 53, 64)	0
dense_1 (Dense)	(None, 53, 53, 128)	8320
conv2d_4 (Conv2D)	(None, 53, 53, 96)	196704
max_pooling2d_3 (MaxPooling2	(None, 26, 26, 96)	0
conv2d_5 (Conv2D)	(None, 26, 26, 96)	83040
max_pooling2d_4 (MaxPooling2	(None, 13, 13, 96)	0
dense_2 (Dense)	(None, 13, 13, 512)	49664
flatten_1 (Flatten)	(None, 86528)	0
dense_3 (Dense)	(None, 256)	22151424
dense_4 (Dense)	(None, 5)	1285

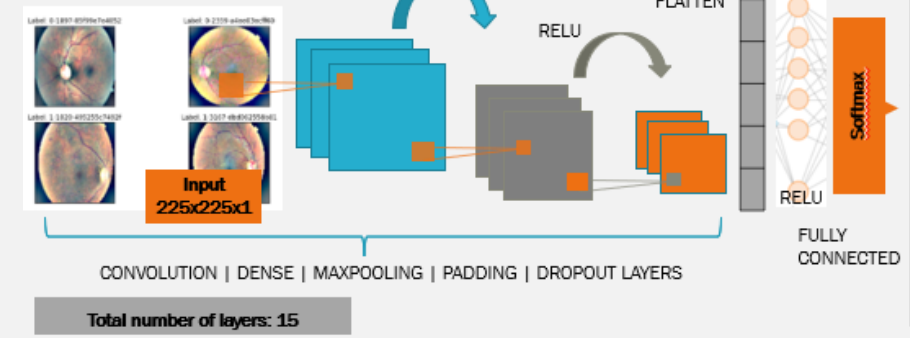
Total params: 22,603,589
Trainable params: 22,603,589
Non-trainable params: 0

Accuracy Train: 72.3% | Accuracy Test: 73.7%

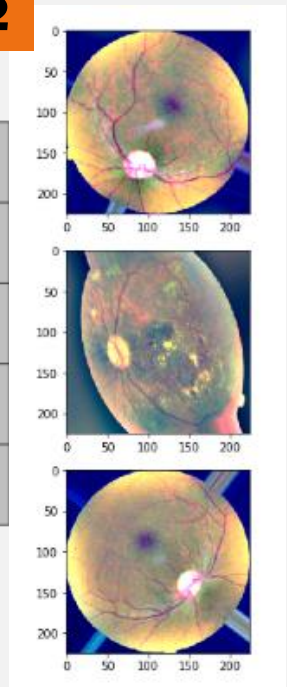
CONVOLUTIONAL NEURAL NETWORKS

2. BUILD CNN WITH DATA AUGMENTATION

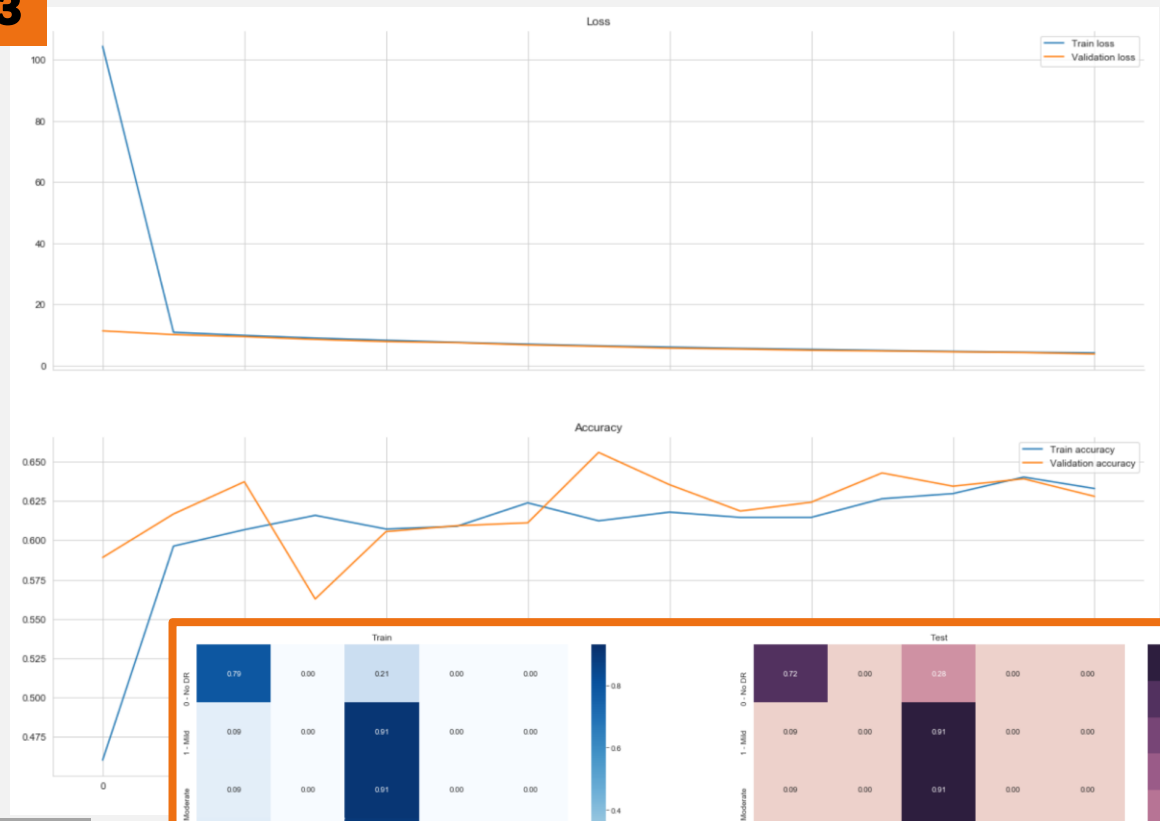
1



2



3



- ASSUMPTIONS:
  - Data Augmentation: horizontal flip, width shift (0.2), zoom (0.1)
  - Data validation split: 0.2
  - Callbacks: early stopping
  - Batch size: 50 | Epochs: 15

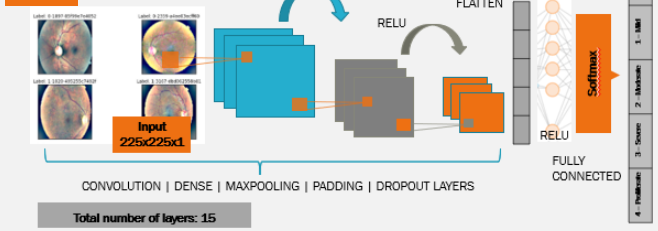
Accuracy Train: 64.8%

Accuracy Test: 69.8%

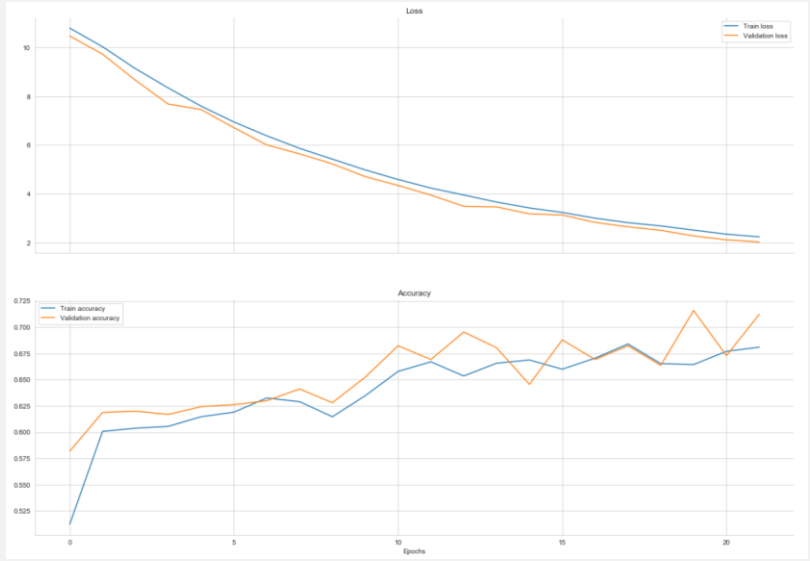
CONVOLUTIONAL NEURAL NETWORKS

3. BUILD CNN WITH DATA AUGMENTATION & TEST TIME AUGMENTATION

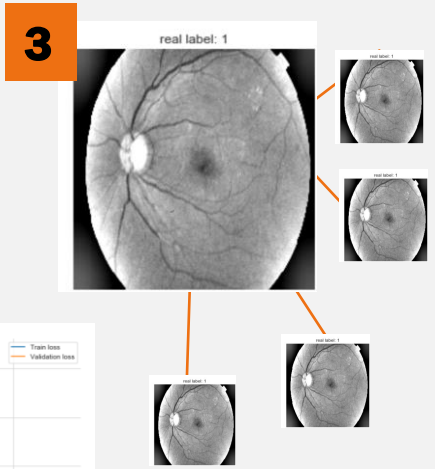
1

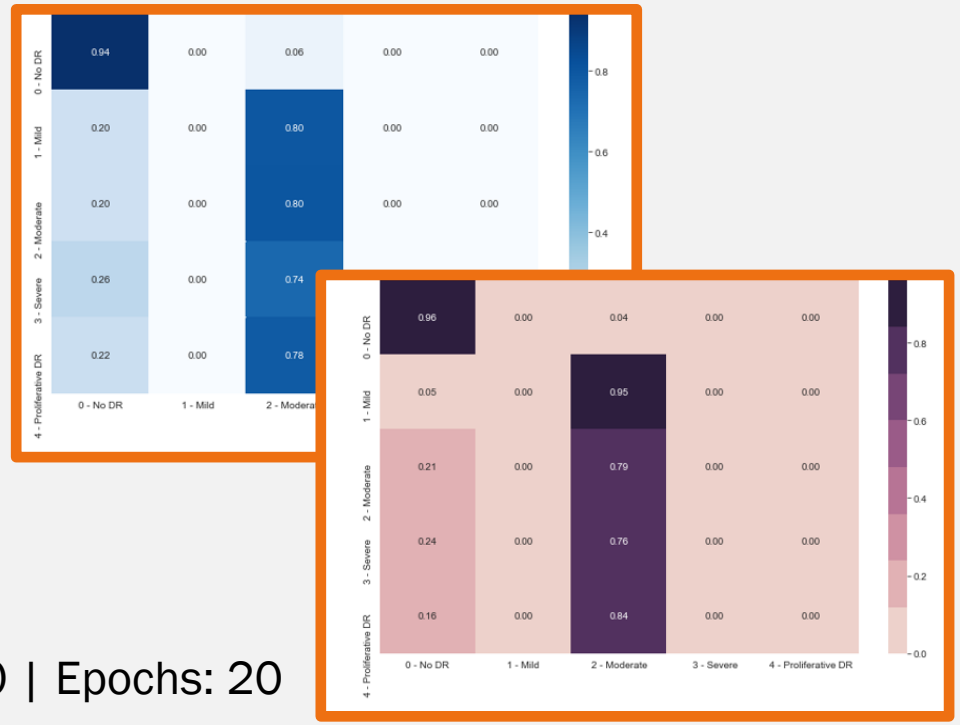


2



3

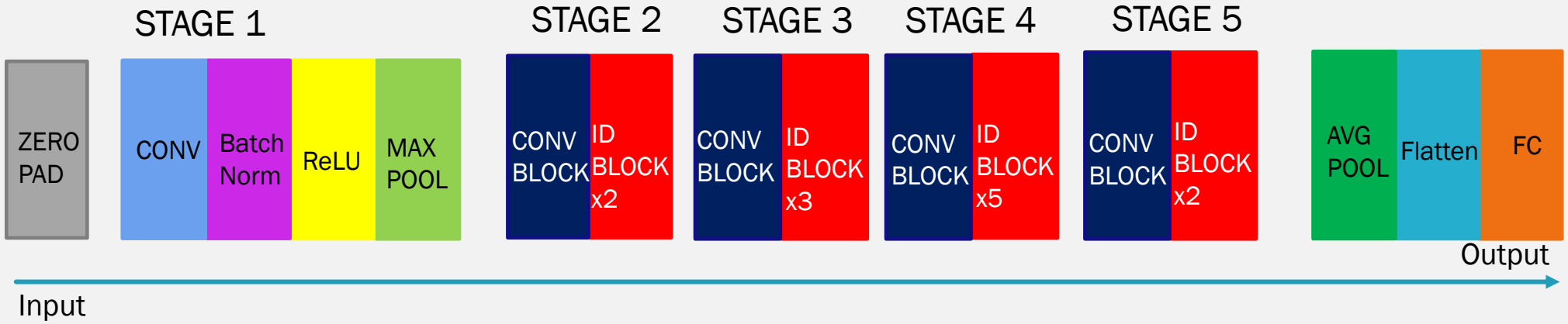




- Batch size: 60 | Epochs: 20

Accuracy Train: 68.1%| Accuracy Test: 70.7%

TRANSFER LEARNING

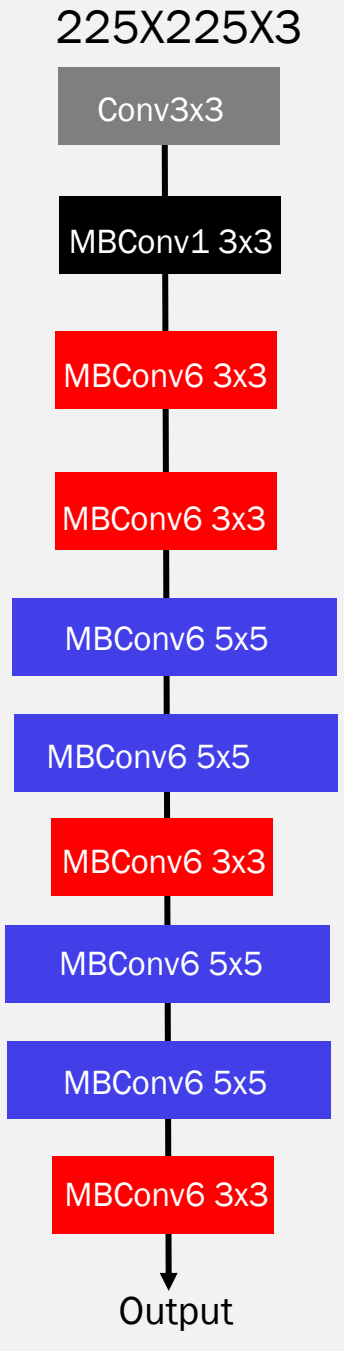


RESNET50

- CNN trained with > 1m images
- 50 layers
- Consist of 5 stages each with convolution and identity block

EFFICIENTNETB0

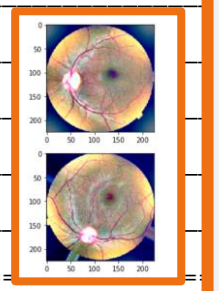
- Less parameters than others
- Inverted bottleneck MBConv
- Fewer channels & depth separable convolution



TRANSFER LEARNING

1. RESNET50

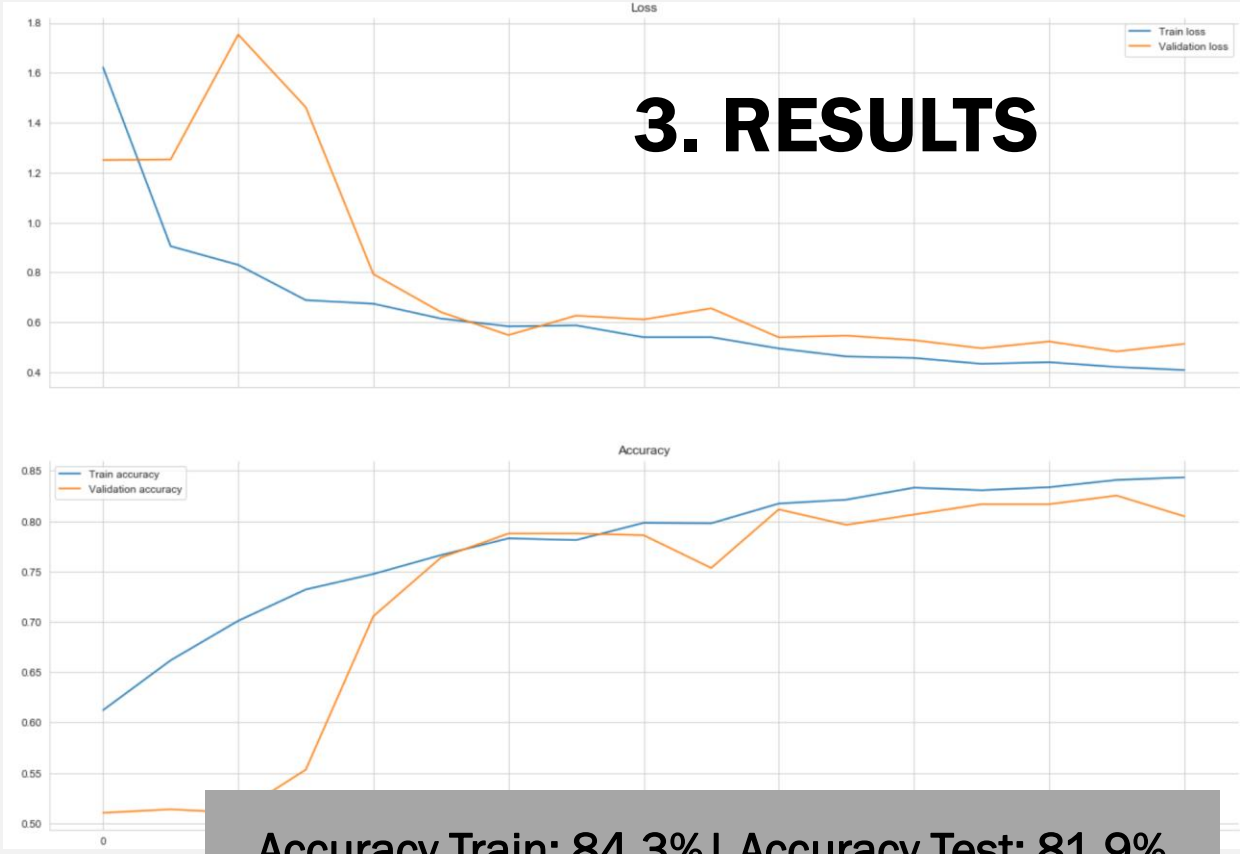
```
_________________________________________________________________
global_average_pooling2d_2 (Glo (None, 2048)
_________________________________________________________________
dropout_4 (Dropout)             (None, 2048)
_________________________________________________________________
dense_2 (Dense)                 (None, 1024)
_________________________________________________________________
dropout_5 (Dropout)             (None, 1024)
_________________________________________________________________
final_output (Dense)            (None, 5)
=================================================================
```



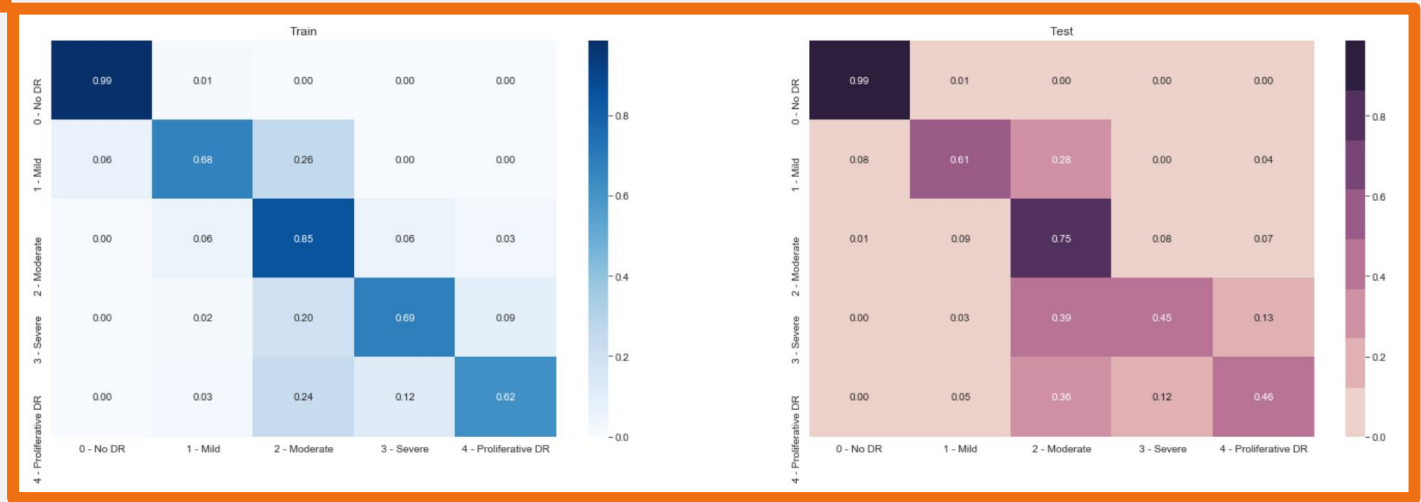
2. STEPS & TUNING

- Custom network
- Freeze the model
- Train the layers added
- Unfreeze layers in base network
- Joined train layers and part added

3. RESULTS



Accuracy Train: 84.3% | Accuracy Test: 81.9%



TRANSFER LEARNING

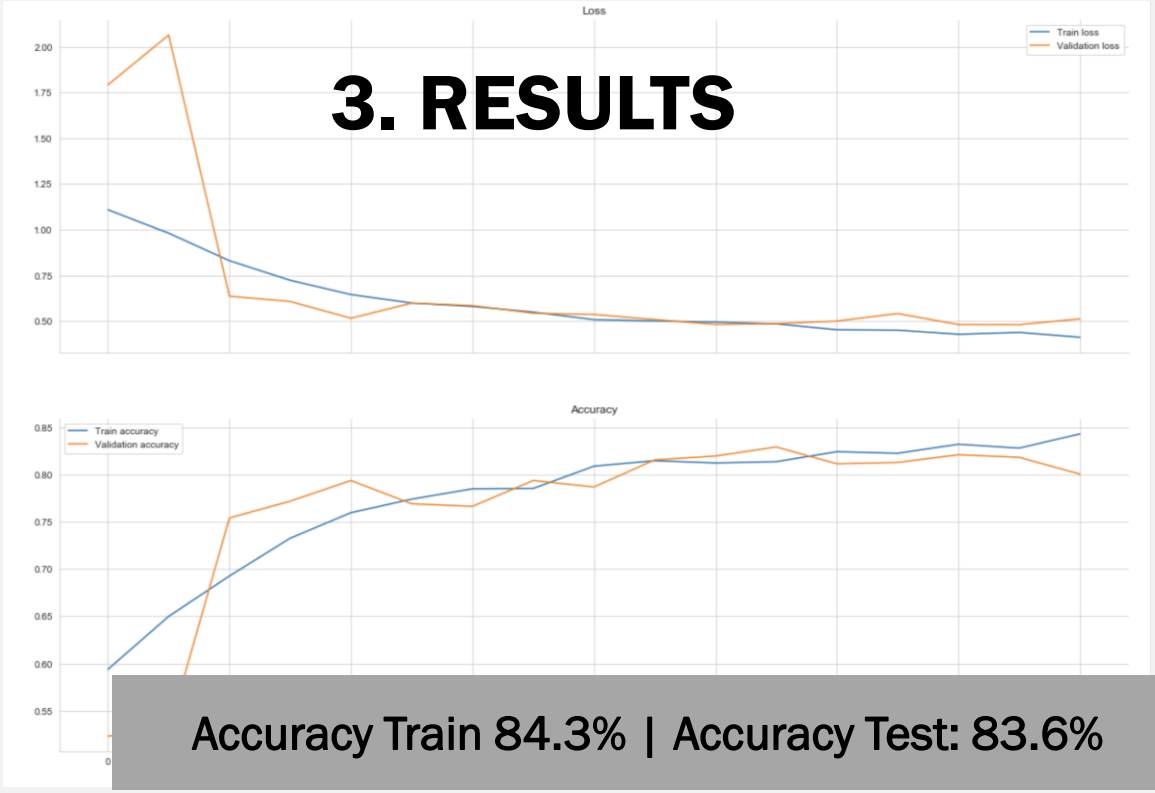
1. EFFICIENTNETB0

global_average_pooling2d_2 (Glo	(None, 2048)	0
dropout_4 (Dropout)	(None, 2048)	0
dense_2 (Dense)	(None, 1024)	2098176
dropout_5 (Dropout)	(None, 1024)	0
final_output (Dense)	(None, 5)	

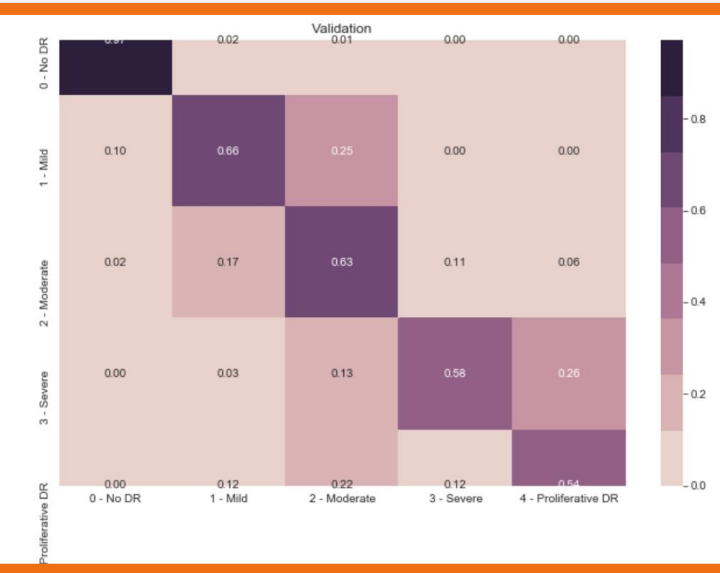
2. STEPS & TUNING

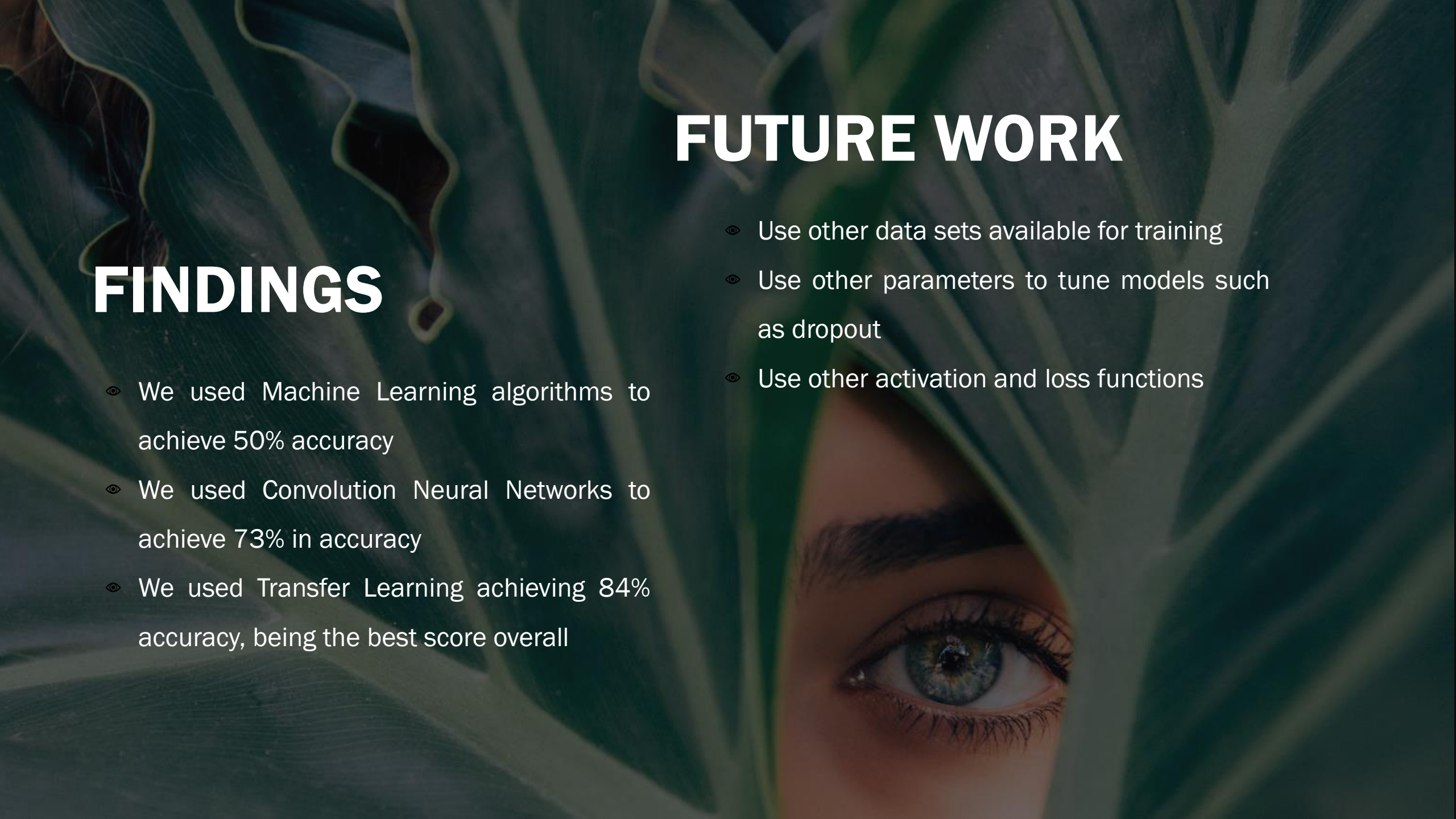
- Custom network
- Freeze the model
- Train the layers added
- Unfreeze layers in base network
- Joined train layers and part added

3. RESULTS



Accuracy Train 84.3% | Accuracy Test: 83.6%





FINDINGS

- We used Machine Learning algorithms to achieve 50% accuracy
- We used Convolution Neural Networks to achieve 73% in accuracy
- We used Transfer Learning achieving 84% accuracy, being the best score overall

FUTURE WORK

- Use other data sets available for training
- Use other parameters to tune models such as dropout
- Use other activation and loss functions

Thank You

Enjoy the break

Hanna Kerr

- Machine Learning models
- Transfer Learning model

Daniela Matinho

- CNN models
- Presentation

Yuling Gu

- Exploratory analysis
- Transfer Learning model