Classifying Diabetic Retinopathy from fundus image Using Deep Learning

JOVITA V
20-PDS-012

- Description of the workflow
- Hardware and software requiremennts
- Method of Solution
- Expected Outcome

Workflow

Train Images
(85 % = 3112))

Validation Images
(15 % = 550)

Test Images

Pre -processing
&
Data Augumentation

Model
(Transfer Learning)

Evaluation Metrics
(kappa Score)

Model Interpretability
(Grad-cams)

Prediction

HARDWARE

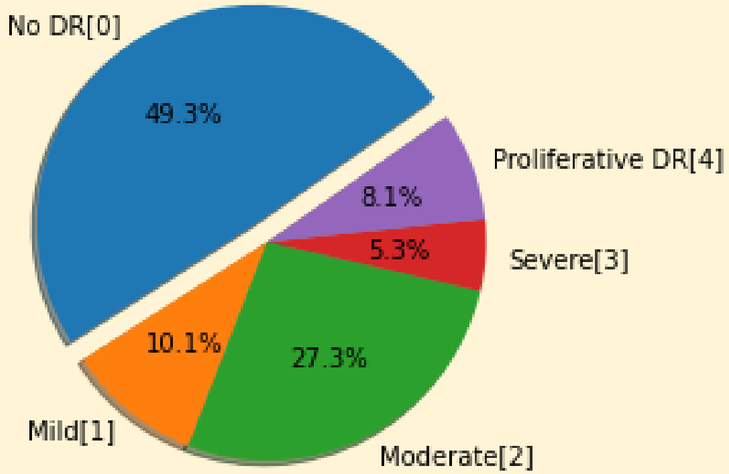
- RAM : 4GB
- System Type : 64-bit operating system, x64-based processor
- Processor : AMD PRO A4-3350B APU with Radeon R4 Graphics 2.00 GHz
- Device Name : Lenova

SOFTWARE

- Google Colab
- Programming Language : Python

EDA

Pie Chart Analysis of Number of Images on each target label:

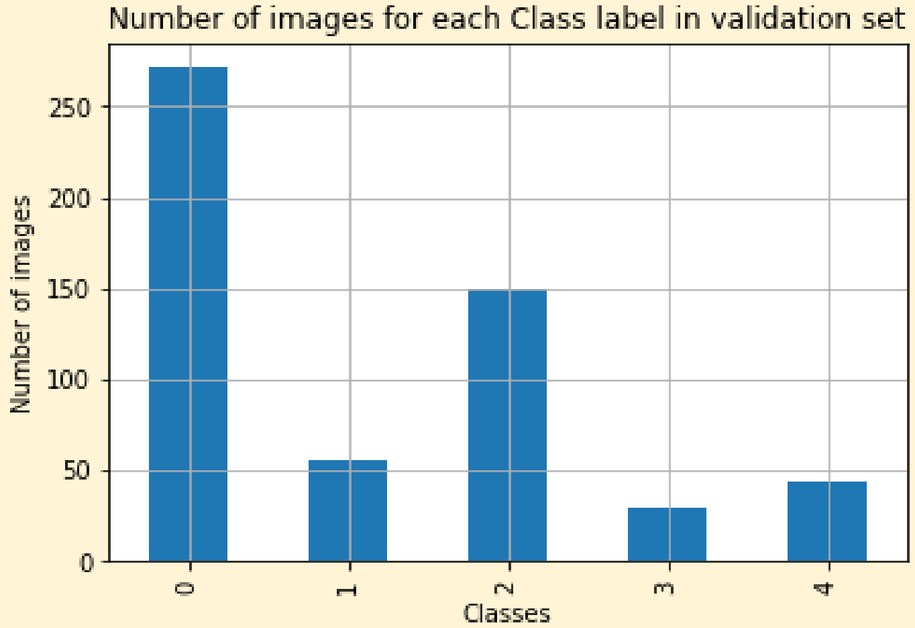


Train Images

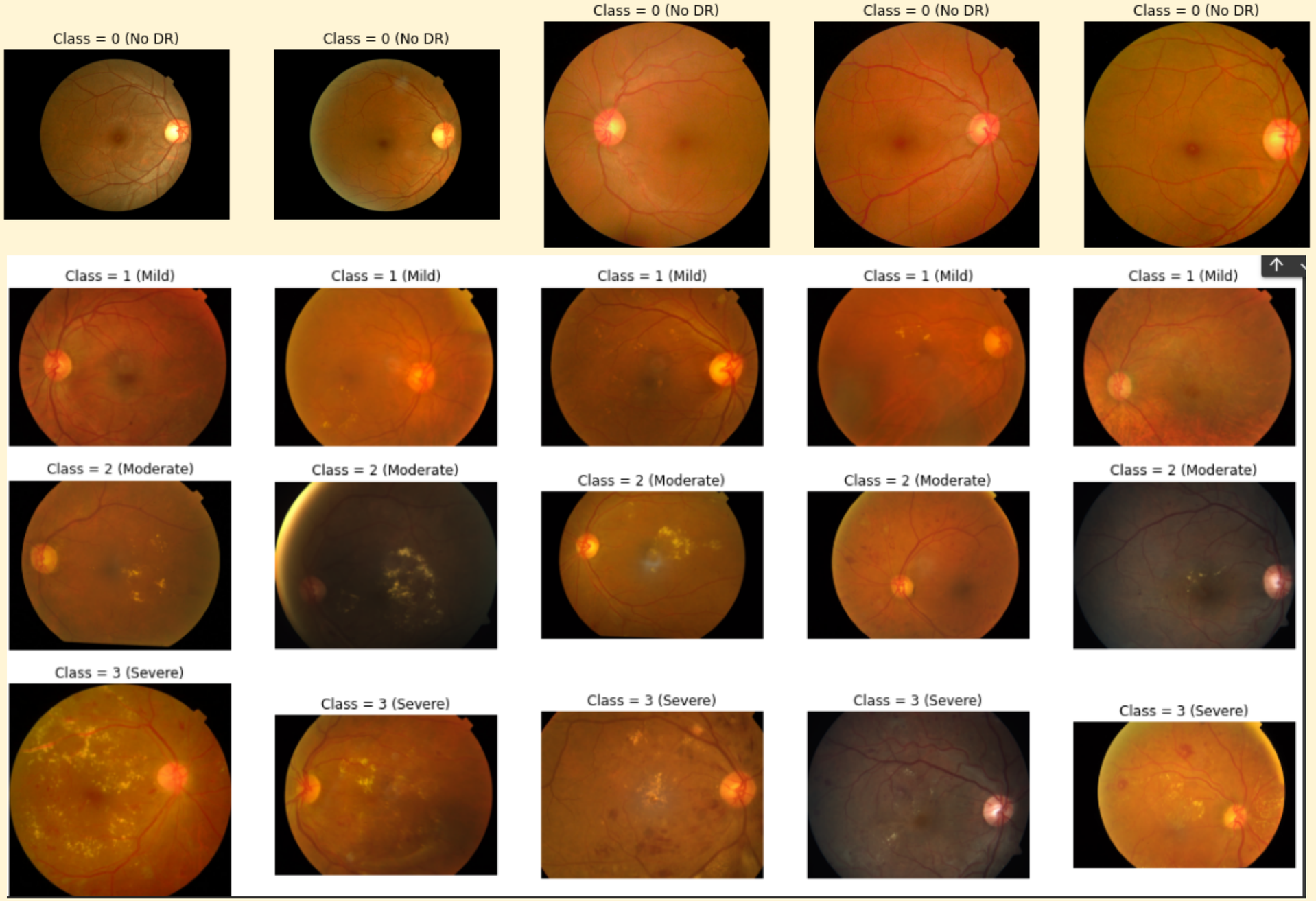
(85 % = 3112))

Validation Images

(15 % = 550)

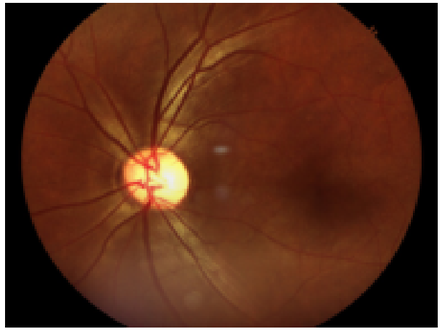


Dataset

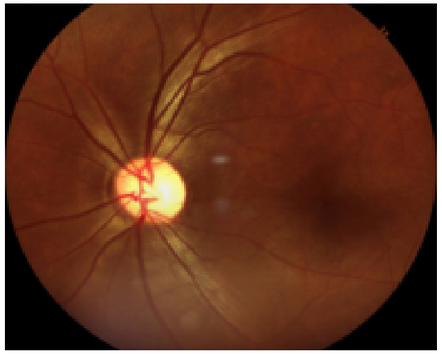

Class = 0 (No DR)
Class = 0 (No DR)
Class = 0 (No DR)
Class = 0 (No DR)
Class = 0 (No DR)
Class = 1 (Mild)
Class = 1 (Mild)
Class = 1 (Mild)
Class = 1 (Mild)
Class = 1 (Mild)
Class = 2 (Moderate)
Class = 2 (Moderate)
Class = 2 (Moderate)
Class = 2 (Moderate)
Class = 2 (Moderate)
Class = 3 (Severe)
Class = 3 (Severe)
Class = 3 (Severe)
Class = 3 (Severe)
Class = 3 (Severe)

Data Pre-processing

Before Image Processing:



Step-1: (Cropping extra dark pixels)

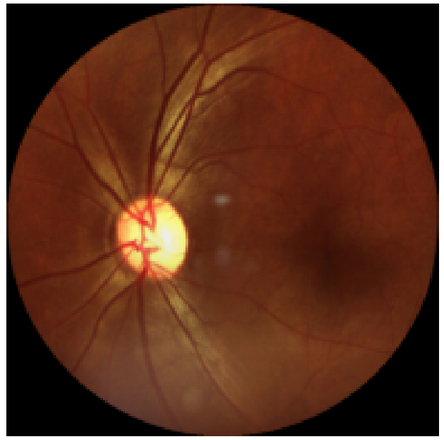


Cropping the Extra Dark Images

Step-2: (Resizing the image)

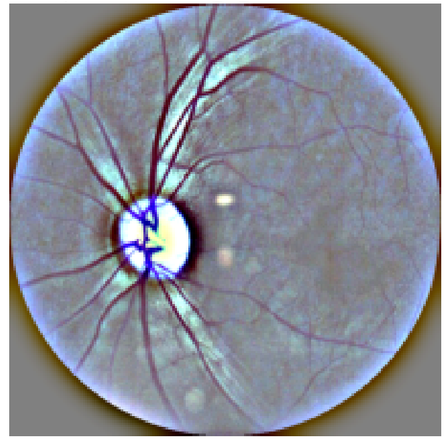
Resizing = (512,512)

Step-3: (Drawing circle)

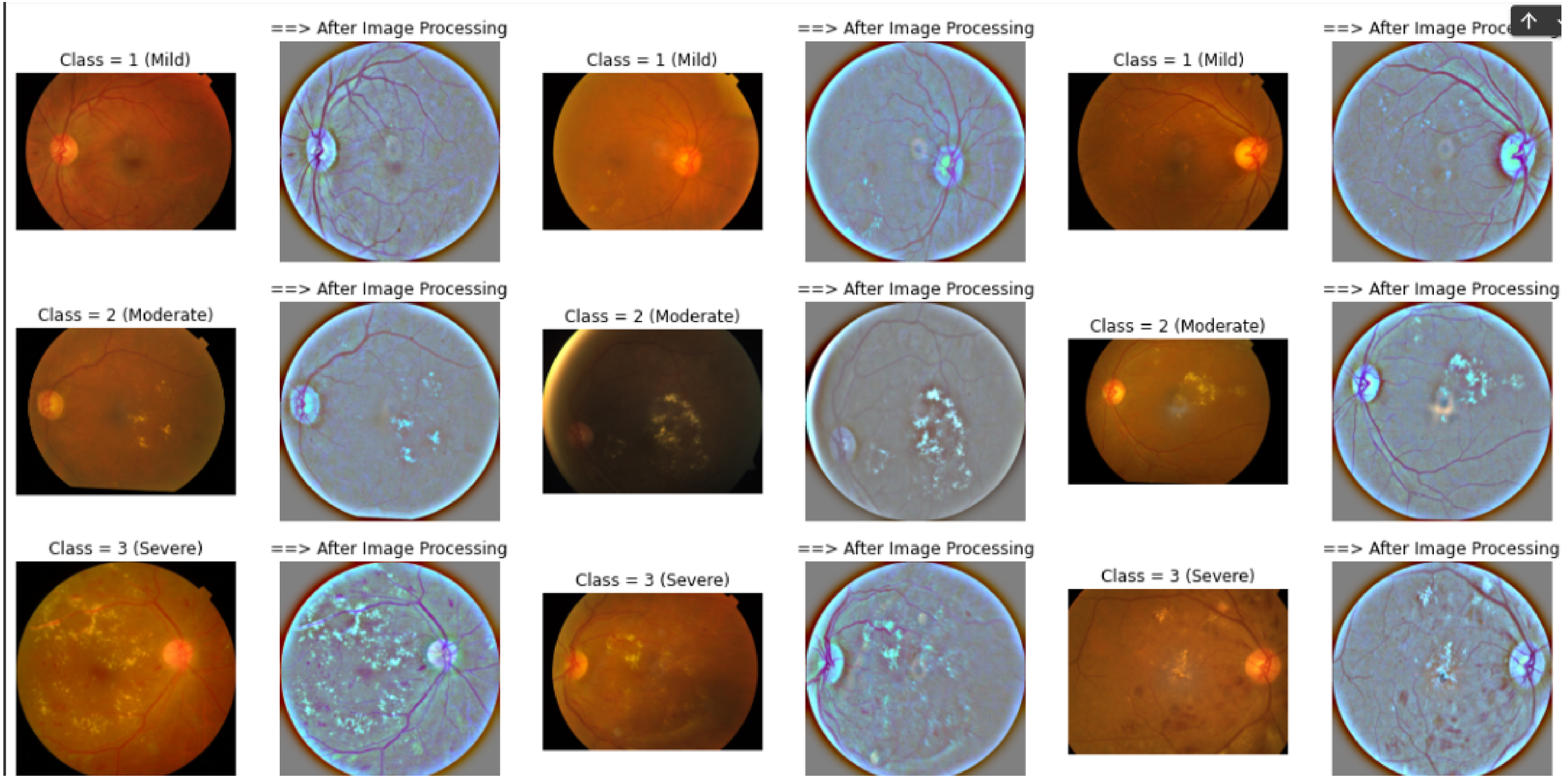


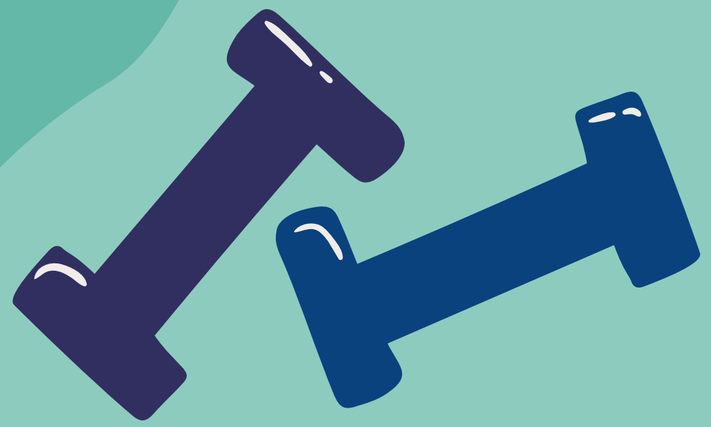
Drawing a circle from the center to give a similar shape to all the images.

Step-4: (Adding gaussian blur)



Using Smoothing technique in order to remove the noise in images


Class = 1 (Mild)
==> After Image Processing
Class = 1 (Mild)
==> After Image Processing
Class = 1 (Mild)
==> After Image Proce
Class = 2 (Moderate)
==> After Image Processing
Class = 2 (Moderate)
==> After Image Processing
Class = 2 (Moderate)
==> After Image Processing
Class = 3 (Severe)
==> After Image Processing
Class = 3 (Severe)
==> After Image Processing
Class = 3 (Severe)
==> After Image Processing



Computing Classweight:

Since the data set is highly imbalanced adding classweight can make it balanced by adding weights to each of the classes.
The class with more weight means they are less in samples. Similarly the class with less weight means they are more in samples.

```
class_weights = class_weight.compute_class_weight(class_weight='balanced'
                                                  classes = [0,1,2,3,4],
                                                  y = train_labels)
print(class_weights)
```

```
[0.40573664 1.98216561 0.73309776 3.79512195 2.47968127]
```

```python
def create_datagen():
    return ImageDataGenerator(
        zoom_range=0.2,  # set range for random zoom
        rotation_range = 180,
        horizontal_flip=True,  # randomly flip images
        vertical_flip=True,  # randomly flip images
    )

# Using generator
data_generator = create_datagen().flow(x_train,train_labels, batch_size=BATCH_SIZE, seed=2019)
```

Data Augumented

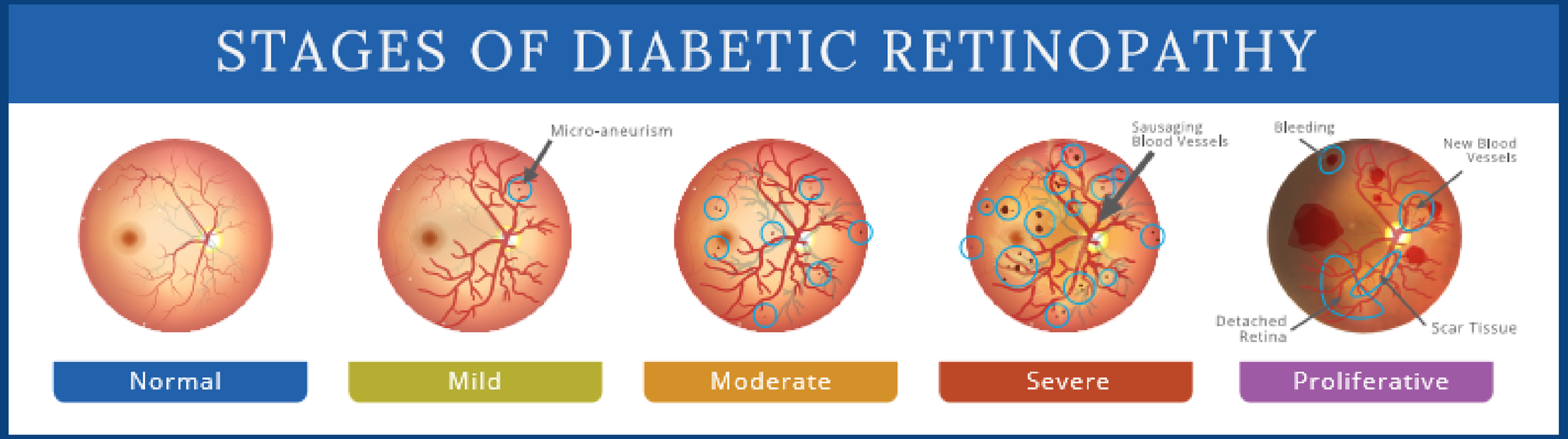
Cohen's Kappa:

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

where,

Po is observed agreement (same as accuracy).

Pe is chance of disagreement observed between raters.

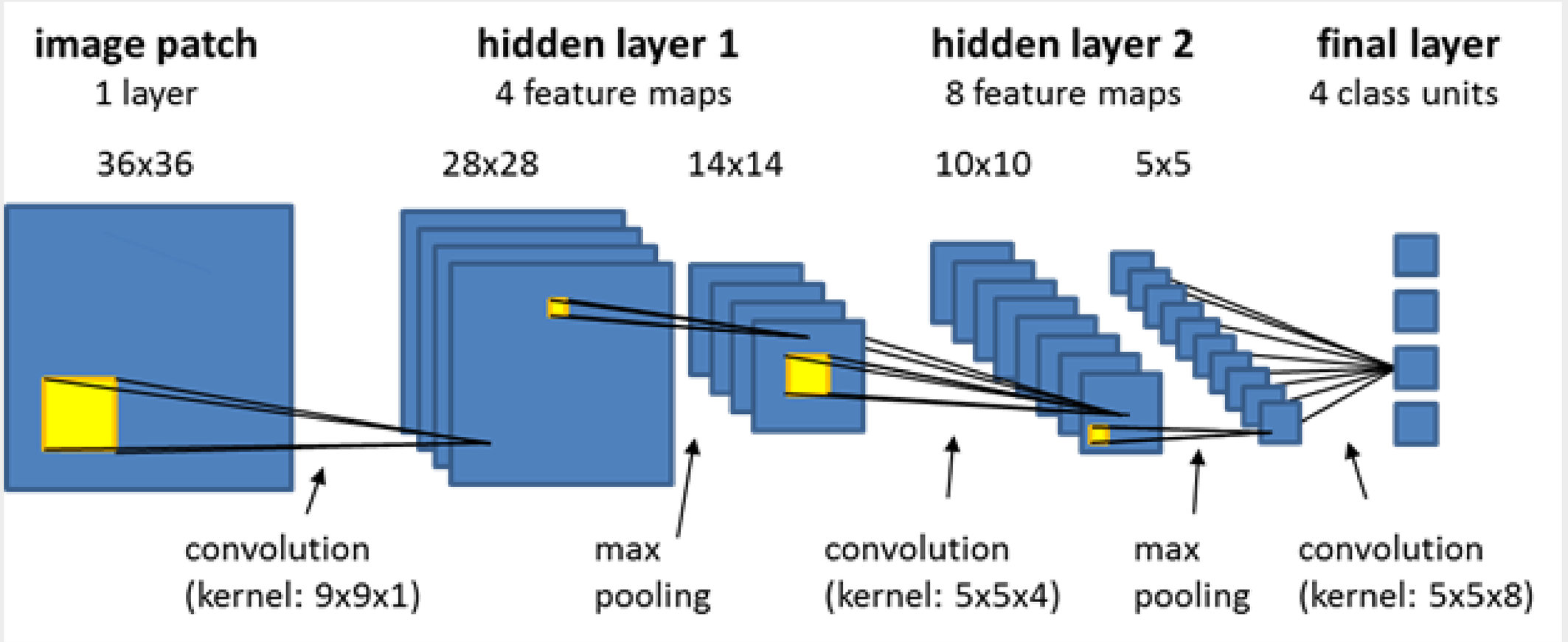


This property will be very helpful because in medical getting higher than expected will be okay because on later diagnosis it can be rectified.

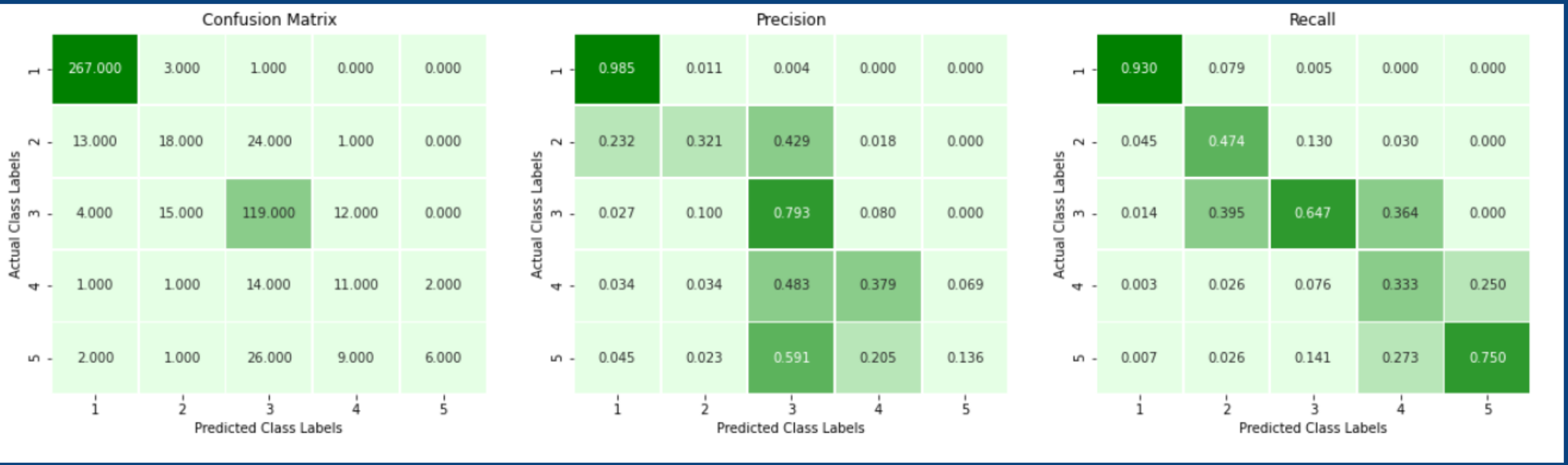
Callbacks

In order to print the kappa metrics on each epoch on validation data

Training Model

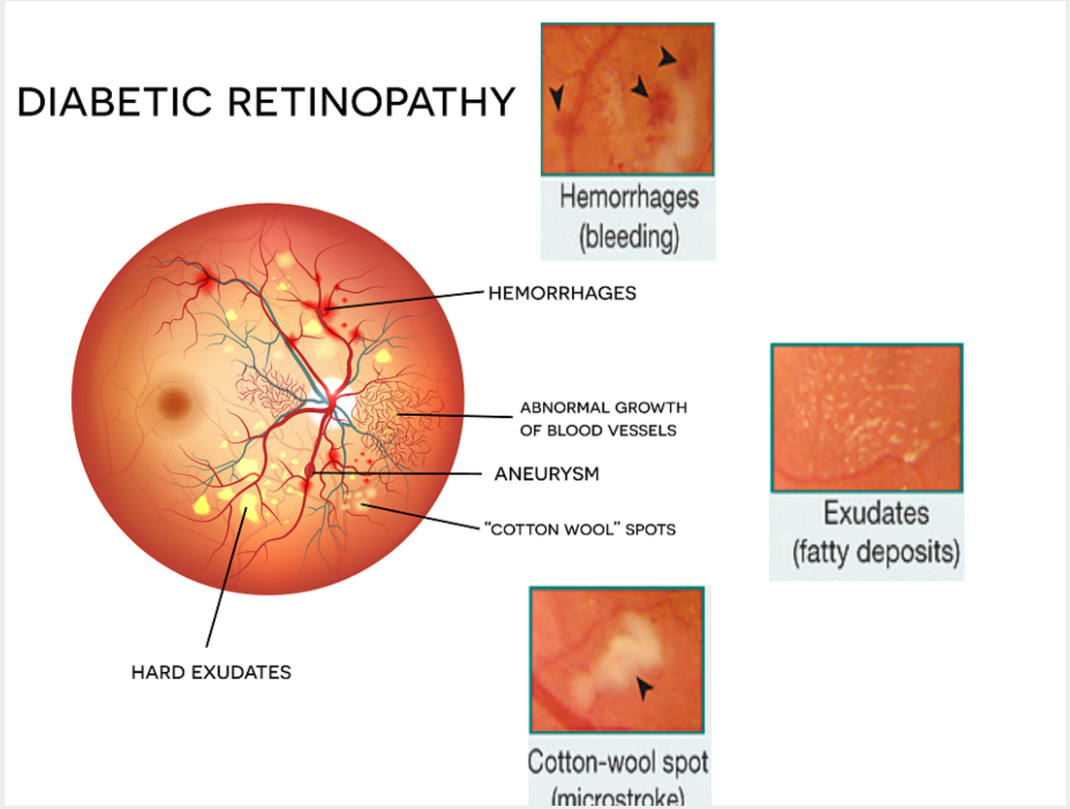


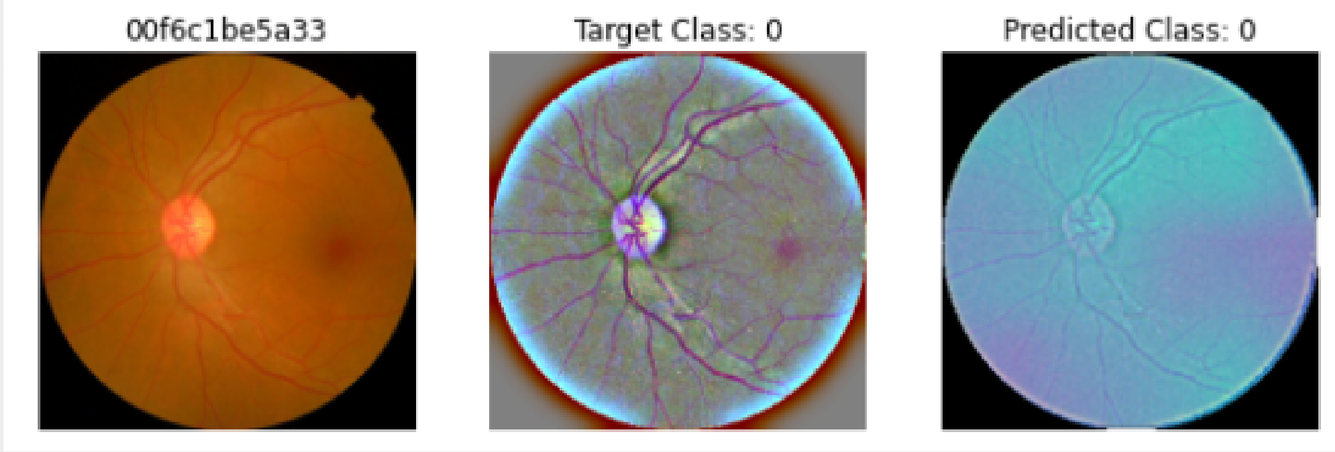
Expected Outcome:

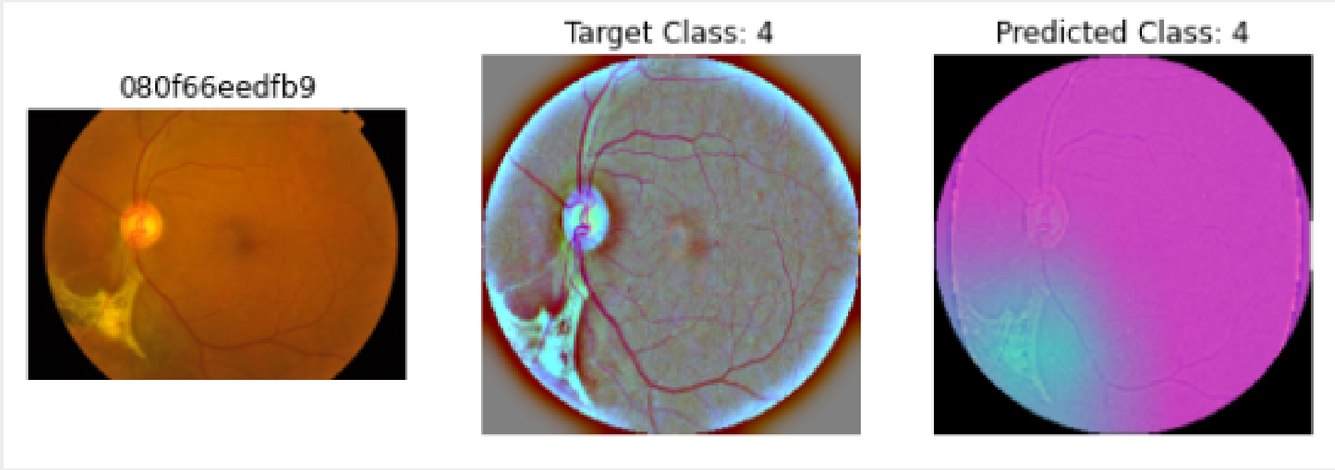


Model Interpretability
(Grad-cams)

Expected Outcome:







Thank You