# Detection of squamous cell carcinoma using deep neural network.

Shivam Panchbhai

201051001

April-June Report

The oral cavity dataset that we used for this project is available on this link->

https://github.com/Tabassum2019/A-histopathological-image-repository-of-normal-epithelium-of-Oral-Cavity-and-OSCC/blob/master/README.md

we have only used the 100x **first set** of the oral cavity Histopathological images. In the first set there are 89 Normal images and 438 OSCC images.

The first task that we did was split the **First set** into train folder and validation folder.

So now the **train set** has 89 normal images and 438 OSCC images.

To correct this imbalance, we used data augmentation on 3 parameters on the train folder of the normal images.

1) Random rotation (between 25% to the left and 25% to the right)
2) Random noise
3) Horizontal Flip

Now, After Augmenting 89 Normal Images to 349

Total 349+89 = 438 Normal Images in the train folder. Now we have 438 Normal images and 438 OSCC images in the train set. The problem of data imbalance is resolved.

Optimiser used: RMSProp.

# Difficulties faced while working on the project.

1) Data imbalance.

2) Splitting the dataset into train, test, and validation.

3) Most of the time we exceeded the “use limit” of the TensorFlow GPU.

4) Specifying directory paths.

5) How to add flatten, dropout and dense layer on the top of the ResNet base layers.

6) Which learning rate, batch size etc is best for the model training.

7) How to use Grad Cam algorithm.

# Things learned while working on the project.

1) Data Augmentation algorithm.

2) Python split folder command.

3) How to use TensorFlow GPU.

4) How to resize the images so that it can be feed into the Resnet model.

5) How to use Image Data Generator command to make batches of the images.

6) How to load the Resnet model.

7) Experimented with splitting the dataset into
    a) Train and Validation sets,
    b) Train, Validation and Test sets,
    c) Train, Validation and Test sets with ReduceLRPlateau.

8) For each split of the dataset, we learned and tried to hyper tune our Resnet model by changing Learning Rates from the range of 1e-3 to 1e-8.

9) We also learnt how to plot and understand the
    a) Training and Validation Accuracy curve.
    b) Training and Validation Loss curve.

10) How to use early stopping code when loss is not decreasing even after certain epochs.

11) How to train the Resnet model, save its trained architecture for that specific dataset and how to save the model trained weights.

12) How to compile a Resnet model with optimizers, learning rate and loss function.

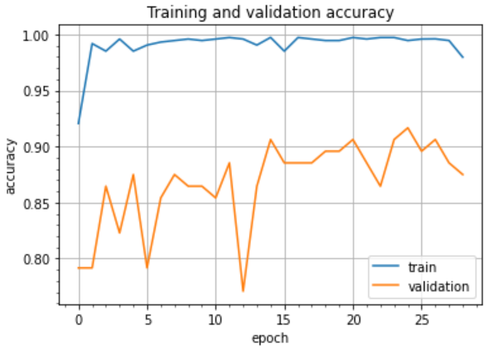
13) We also learnt to load the pretrained model instead of the training model once again for 4-5 hours.

14) We also learnt how to plot confusion matrix and what it means.

15) Grad Cam Algorithm.

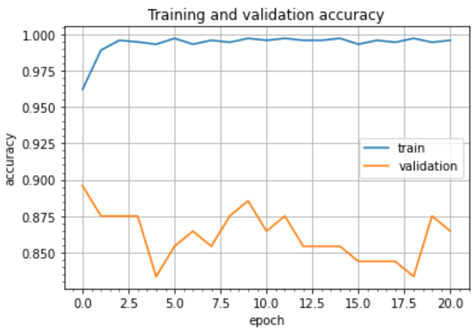
## 1.Tried Hyperparameter Tuning by changing **learning rates**.

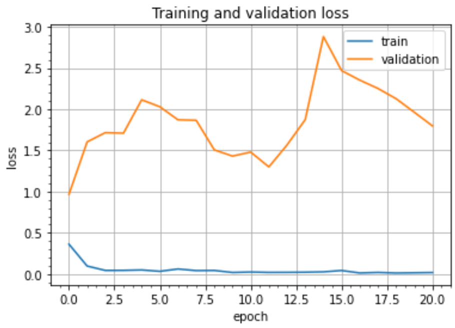
a) When The learning rate is **1e-03** we achieved accuracy prediction of **88.6%**




An **underfit model** as the training loss is flat. This means that model did not learn from the training dataset.

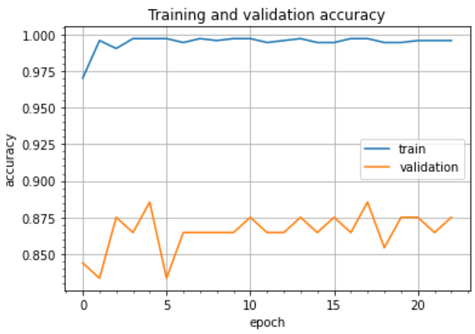
b) When The learning rate is **1e-04** we achieved accuracy prediction of **86.7 %**

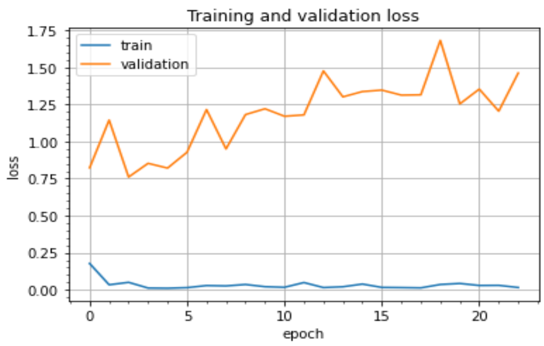





An **underfit model** as the training loss is flat. This means that model has did not learn from the training dataset.

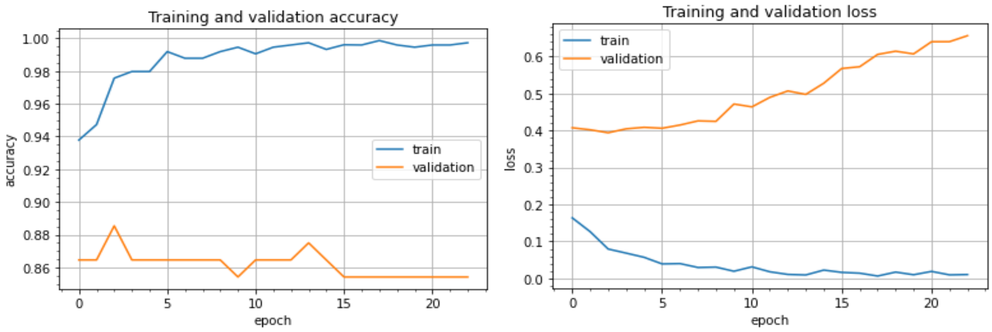
**c)** When The learning rate is **1e-05** we achieved accuracy prediction of **88.6%**





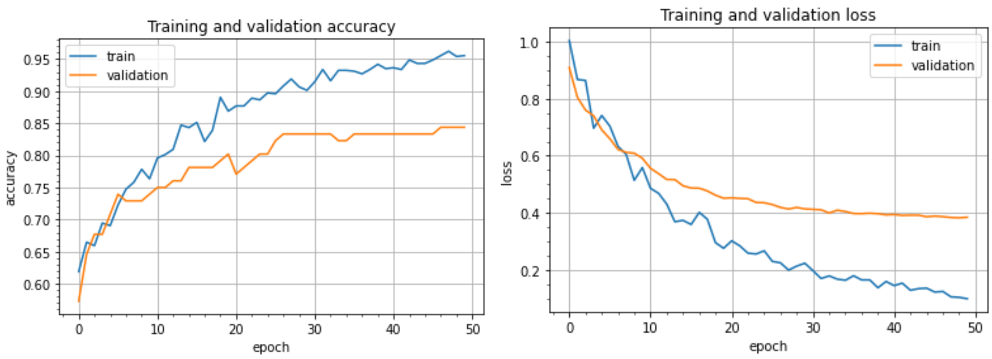

An **underfit model** as the training loss is flat. This means that model has did not learn from the training dataset.

**d)** When The learning rate is **1e-06** we achieved accuracy prediction of **88.6%**



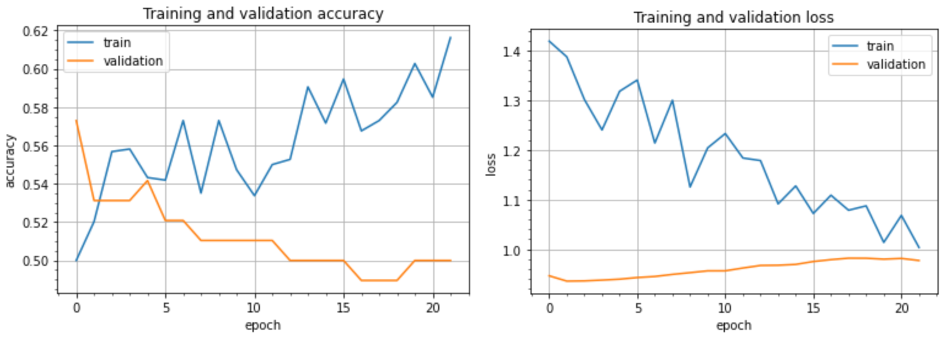

An **underfit model** as the training loss is noisy and decreasing continuously. This means that model did not learn from the training dataset.

**e)** When The learning rate is **1e-07** we achieved accuracy prediction of **85.8%**




An overfit model as the plot of the training loss continues to decrease.

**f)** When The learning rate is **1e-08** we achieved accuracy prediction of **0.5%**




An overfit model as the plot of the training loss continues to decrease.

## 2. Splitting "First Set" data into Train, Validation and Test sets.

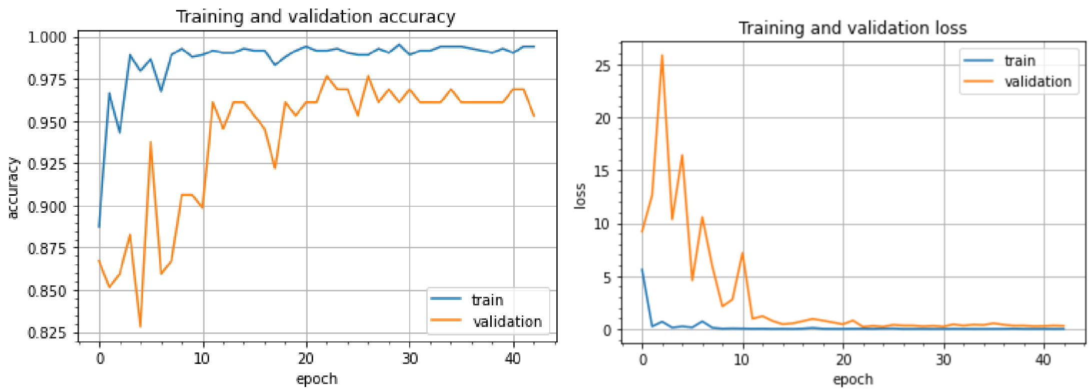
Data augmentation was done earlier only so

We have 438 normal images and 438 OSCC images in Train Dataset.

We have 10 Normal images and 45 OSCC images in Test Dataset.

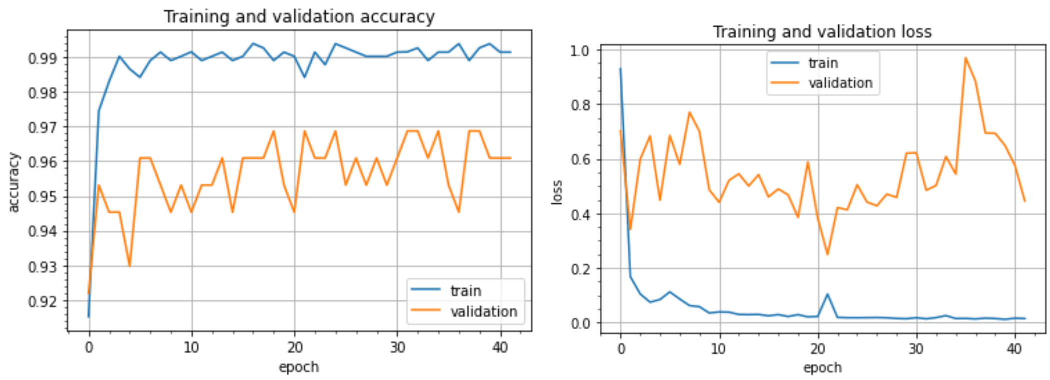
We have 23 Normal images and 124 OSCC images in Validation Dataset.

a) When the learning rate is 1e-3 we achieved accuracy prediction of 92.7%



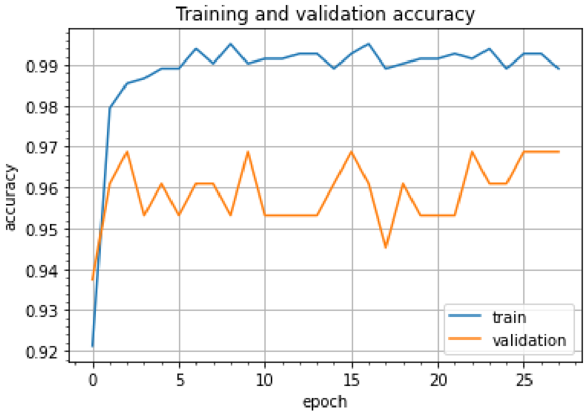

An underfit model as the training loss is flat.

b) When the learning rate is 1e-4 we achieved accuracy prediction of 94.5%




An underfit model as the training loss is flat.

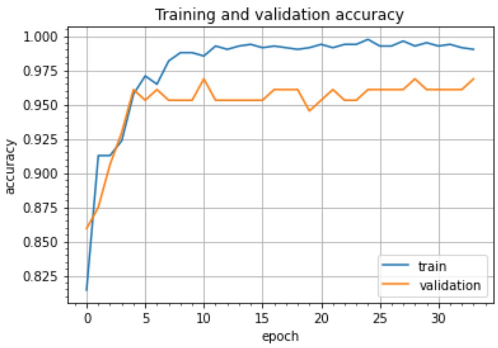
c) When the learning rate is 1e-5 we achieved accuracy prediction of 94.5%

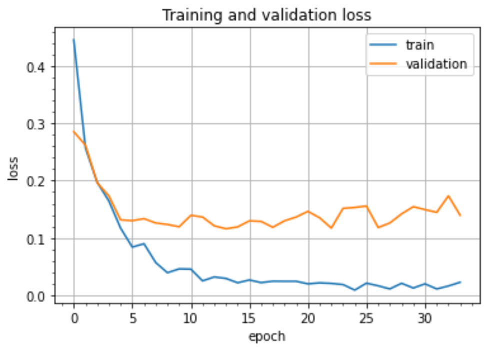




An underfit model as the training loss is noisy.

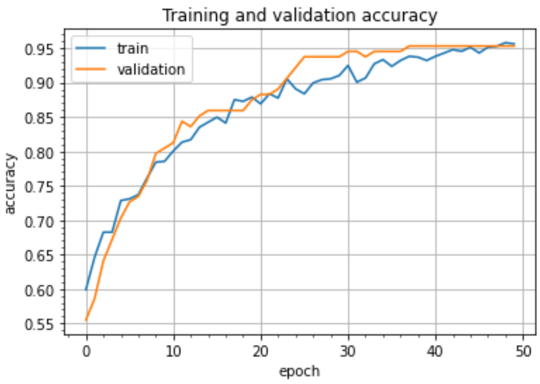
d) When the learning rate is 1e-6 we achieved accuracy prediction of 92.7%

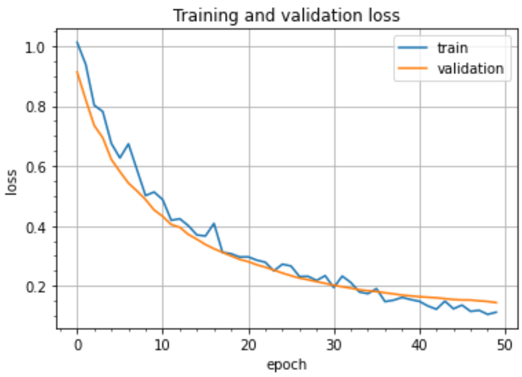



An underfit model as the training loss is flat and noisy.

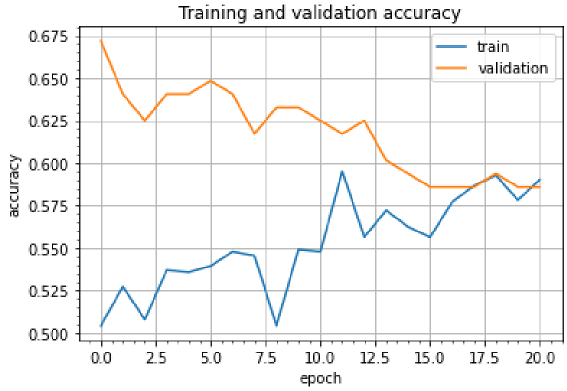
e) When the learning rate is 1e-7 we achieved accuracy prediction of 96.3%

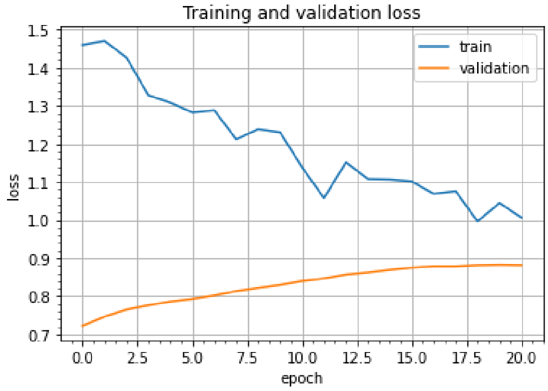



A good fit as training and validation loss are decreasing with a minimal gap between the two final loss values.

f) When the learning rate is 1e-8 we achieved accuracy prediction of 65.4%




An underfit model as the training loss is decreasing continuously.

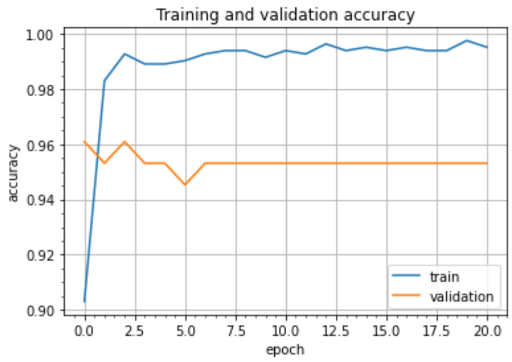
## 3. Using min and max learning rate for Train, Validation and Test sets.

a) When the learning rate is 1e-3 we achieved accuracy prediction of 94.54%



An underfit model as the training loss is flat.

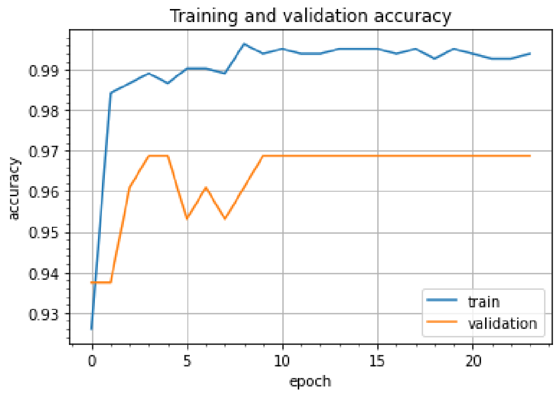
b) When the learning rate is 1e-4 we achieved accuracy prediction of 92.72%





An underfit model as the training loss is flat.

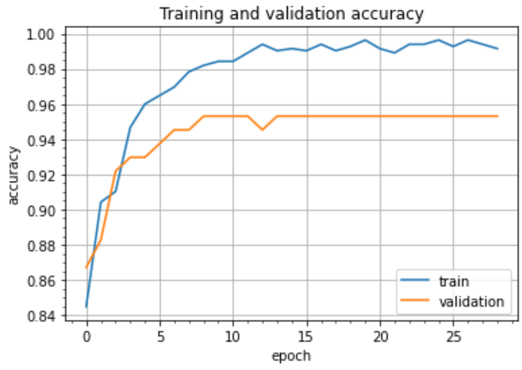
c) When the learning rate is 1e-5 we achieved accuracy prediction of 94.54%

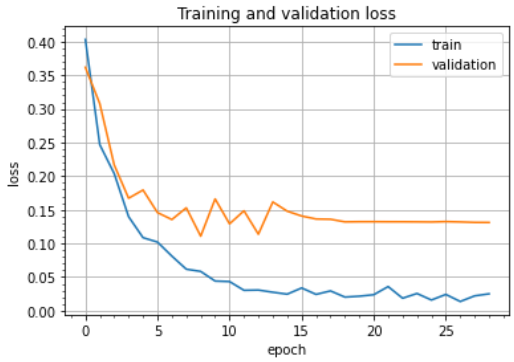




An underfit model as the training loss is noisy and decreasing.

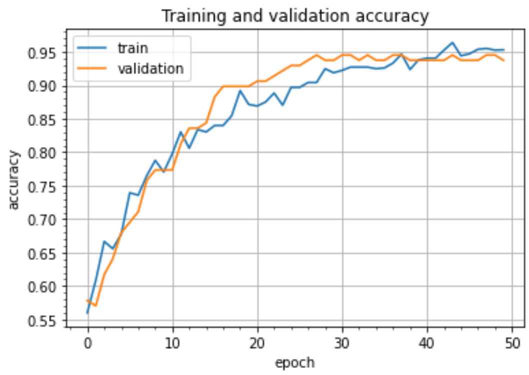
d) When the learning rate is 1e-6 we achieved accuracy prediction of 94.54%.




An underfit model as the training loss is decreasing.

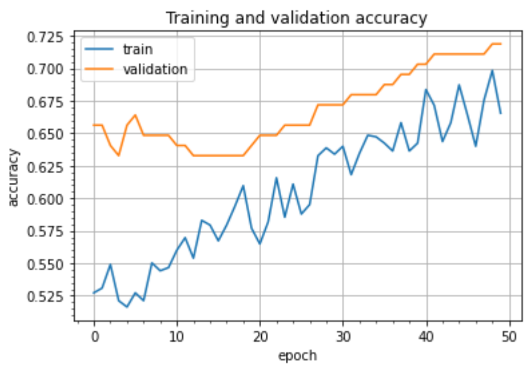
e) When the learning rate is 1e-7 we achieved accuracy prediction of 94.54%





A good fit as the training and validation loss is decreasing and there is a minimal gap between the two.

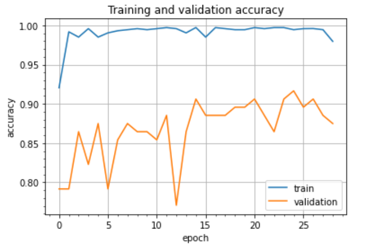
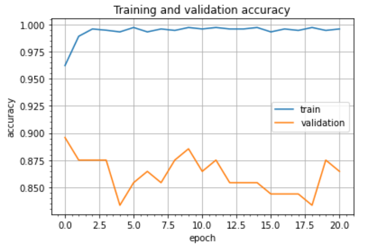
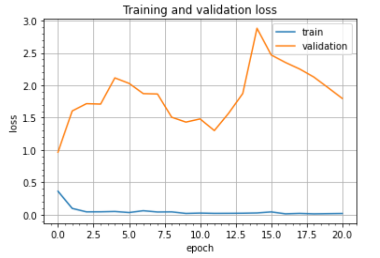
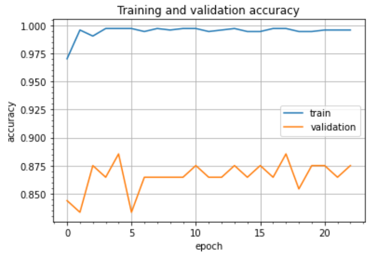
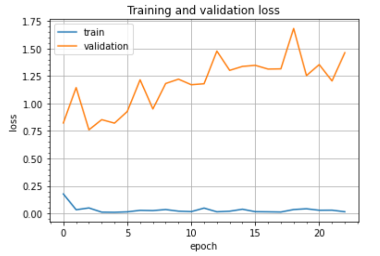
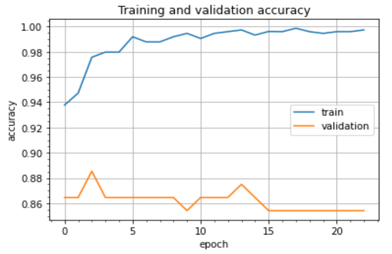
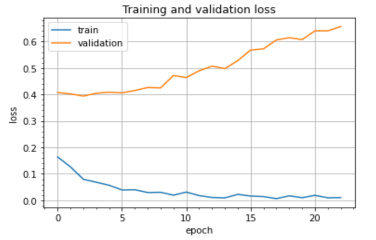
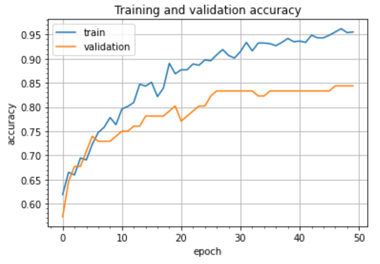
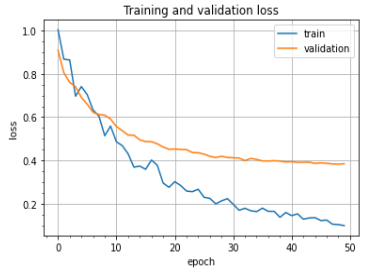
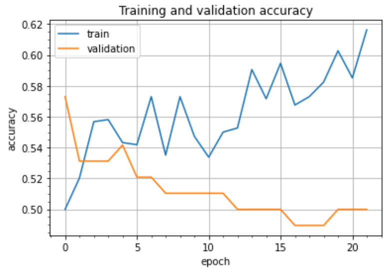
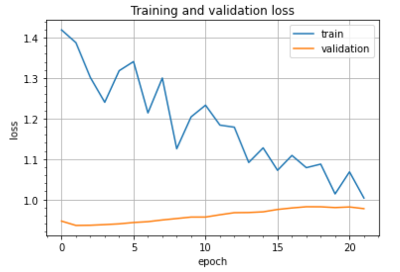
f) When the learning rate is 1e-8 we achieved accuracy prediction of 65.45%

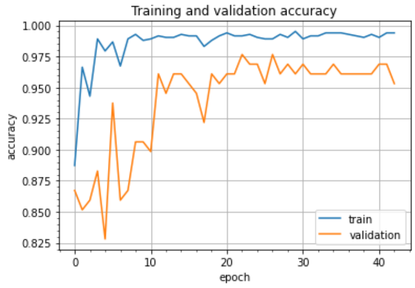
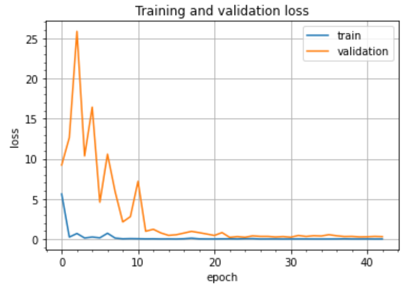
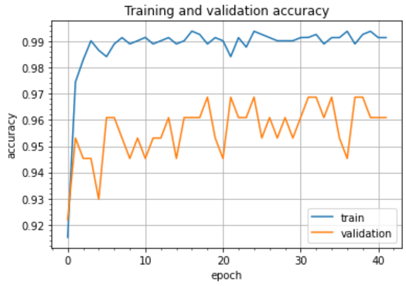
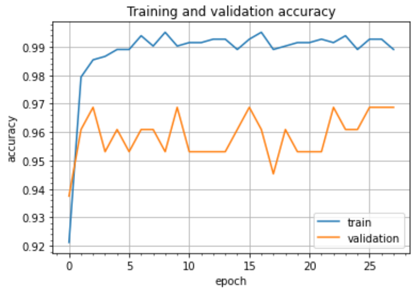
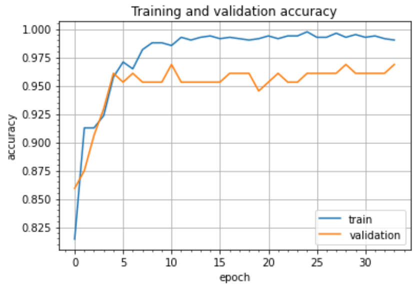
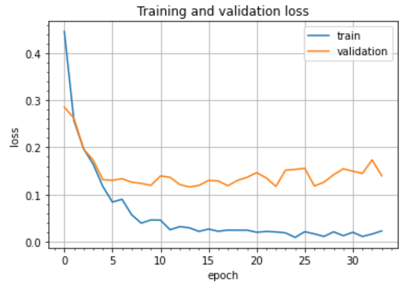
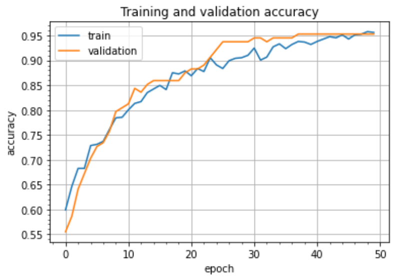
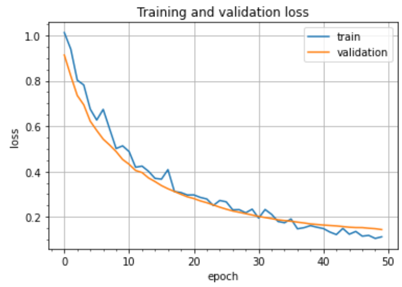
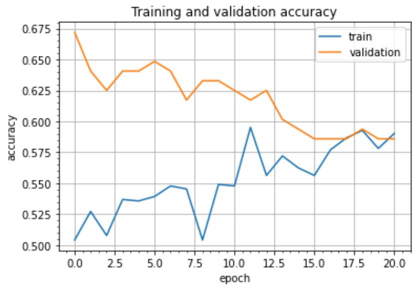
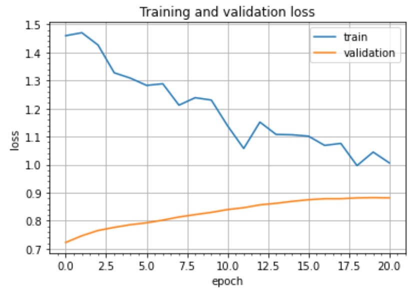


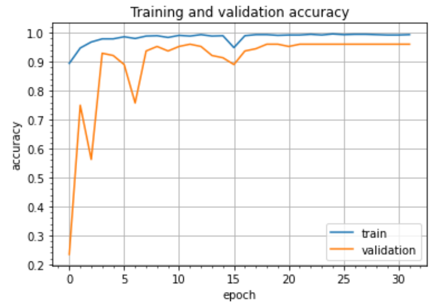
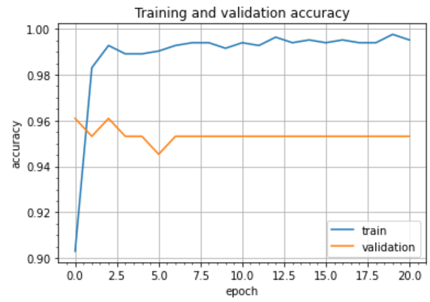
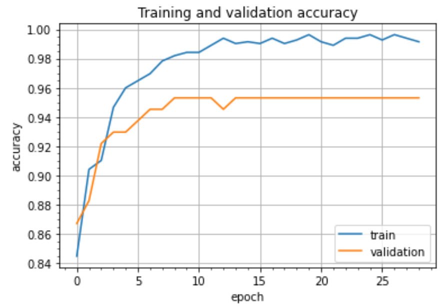
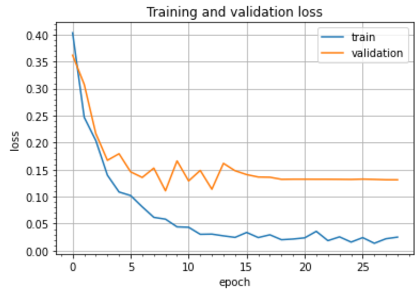
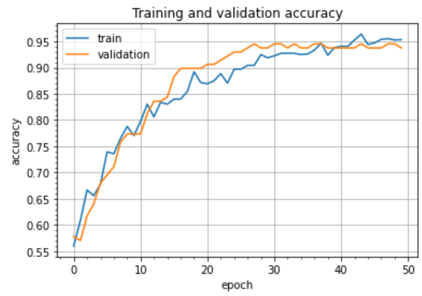
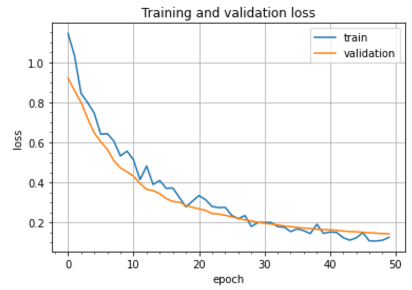
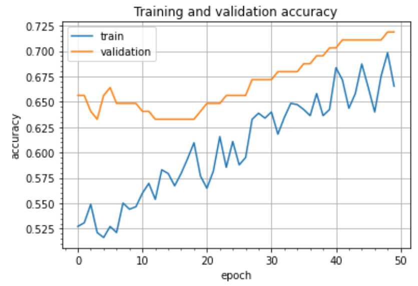
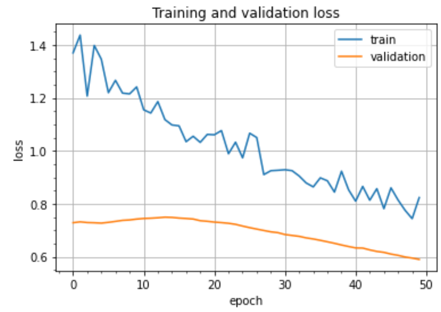


An underfit model as the training loss is decreasing continuously.

# Summary of the experiment done so far.

| Dataset split | Learning rate | Prediction accuracy % | Accuracy curves | Loss curves |
|---|---|---|---|---|
| Train and Val | 1e-3 | 88.6 curve is not good |  |  |
| | 1e-4 | 86.7 |  |  |
| | 1e-5 | 88.6 |  |  |
| | 1e-6 | 88.6 |  |  |
| | 1e-7 | 85.8 |  |  |
| | 1e-8 | 0.5 |  |  |

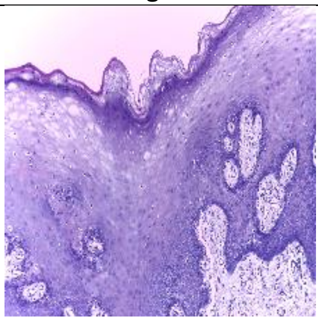
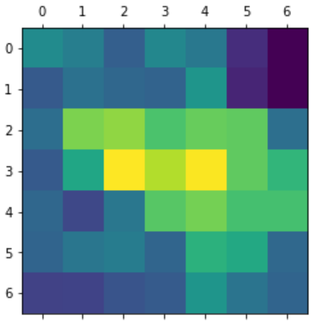
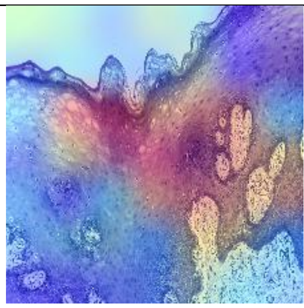
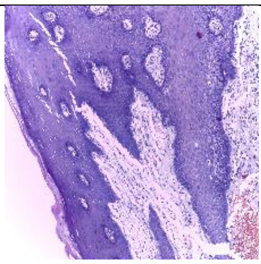
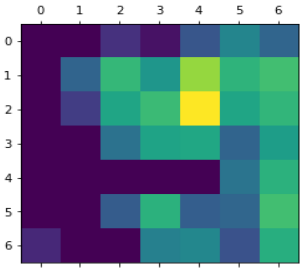
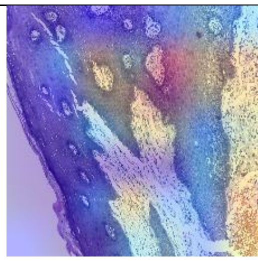
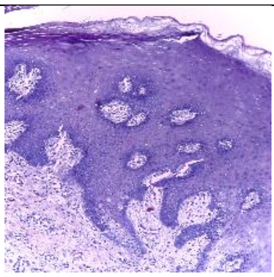
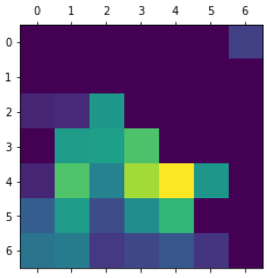
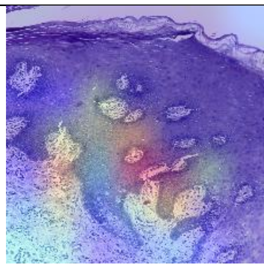
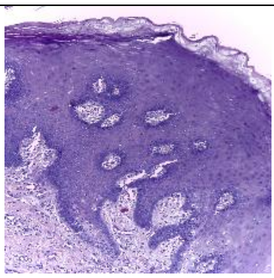
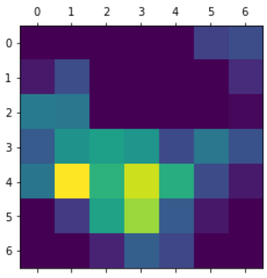
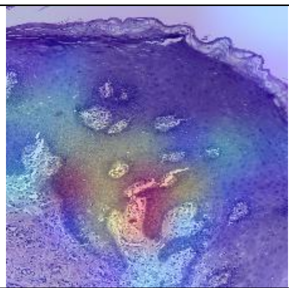
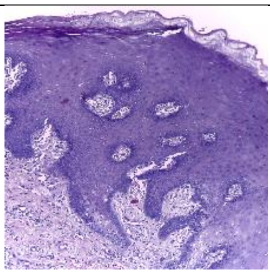
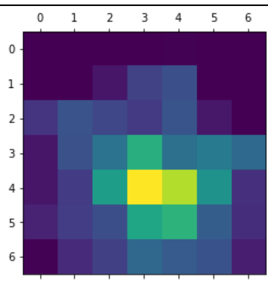
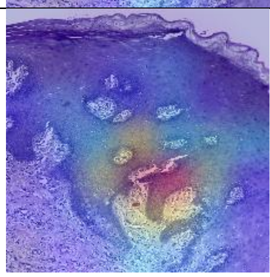
| | | | | |
|---|---|---|---|---|
| | | | | |
| Train Val and Test | 1e-3 | 92.7 |  |  |
| | 1e-4 | 94.5 |  |  |
| | 1e-5 | 94.5 |  |  |
| | 1e-6 | 92.7 |  |  |
| | 1e-7 | 96.3 |  |  |
| | 1e-8 | 65.4 |  |  |
| | | | | |

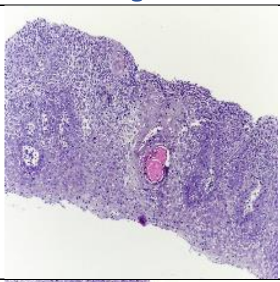
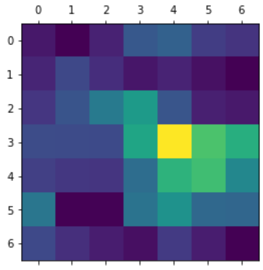
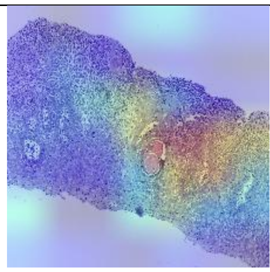
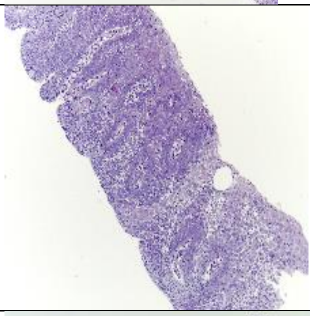
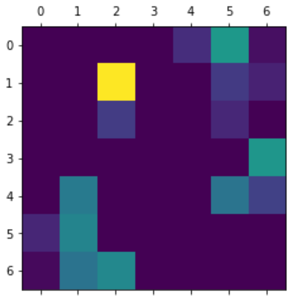
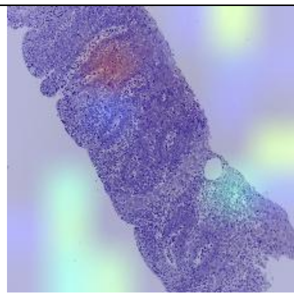
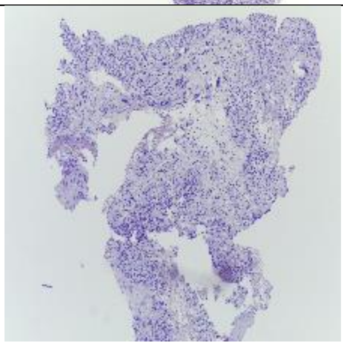
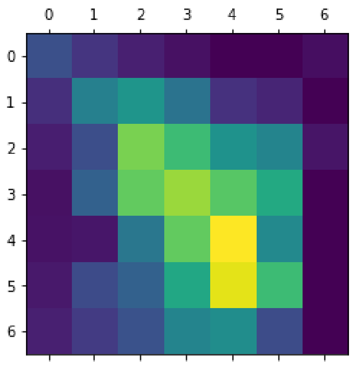
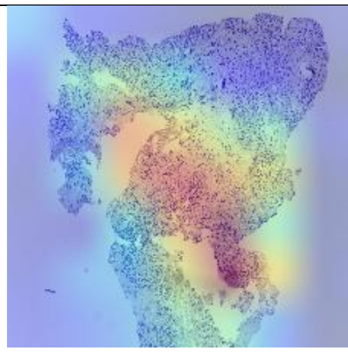
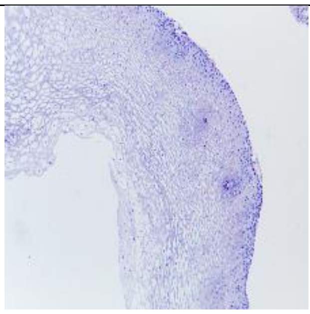
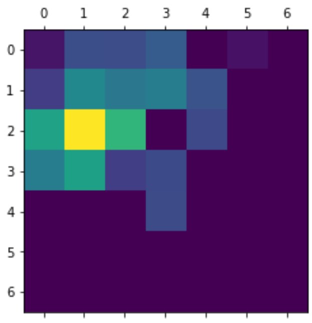
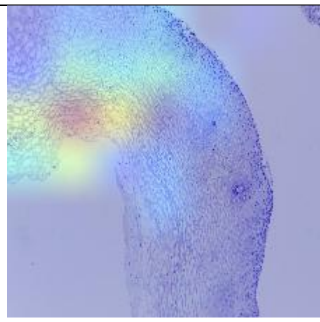
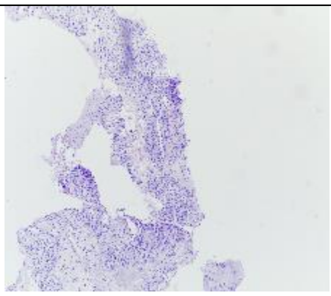
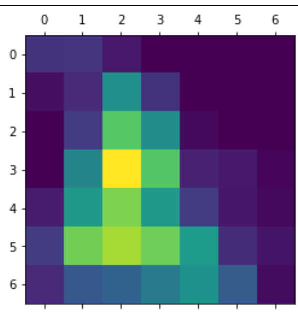
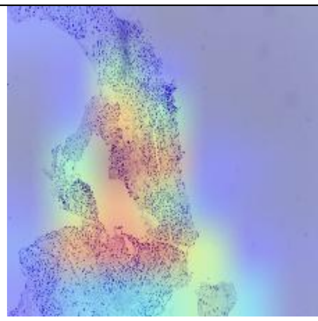
| | | | | |
|---|---|---|---|---|
| Train Val and Test with Reduce plateau | 1e-3 | 94.54 |   |   |
| | 1e-4 | 92.72 |   |   |
| | 1e-5 | 94.54 |   |   |
| | 1e-6 | 94.54 |   |   |
| | 1e-7 | 94.54 curve is better |   |   |
| | 1e-8 | 65.45 |   |   |

## 4. Tried Gradient-weighted Class Activation Mapping (Grad-CAM)

Grad cam is an algorithm from which we can get a sense of what regions of an image the model or the CNN is looking at to make a prediction.

This is done by seeing which region of the image that have been processed are now accounting for most gradients and most updated to the neural network.

| S.No | Normal Images | Heatmap | Grad Cam |
|---|---|---|---|
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

| S.No | OSCC images | Heatmap | Grad Cam |
|---|---|---|---|
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |