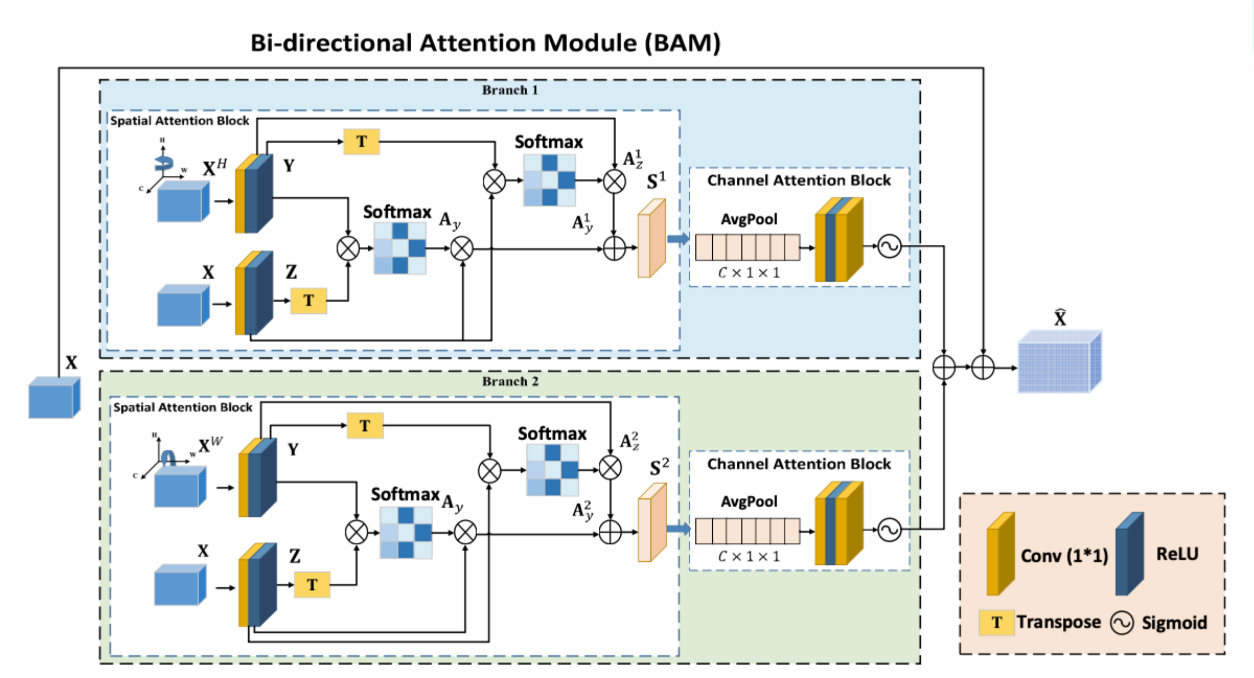
AUTOMATED FACE RECOGNITION IN THE WILD

PURPOSE OF THIS PROJECT

- Occlusion is a common artefact while capturing face images, where only a part of the face becomes visible due to the use of various accessories.
- Face recognition systems find it difficult to match occluded face images with a gallery full of complete face images.
- In recent times, due to use of surgical masks, he performance of face recognition systems has degraded more causing law-enforcement agencies to look for alternate solutions.
- This project aims to match the masked faces with non-occluded ones by utilising the spatial correlation between the periocular regions with the help of an attention framework.

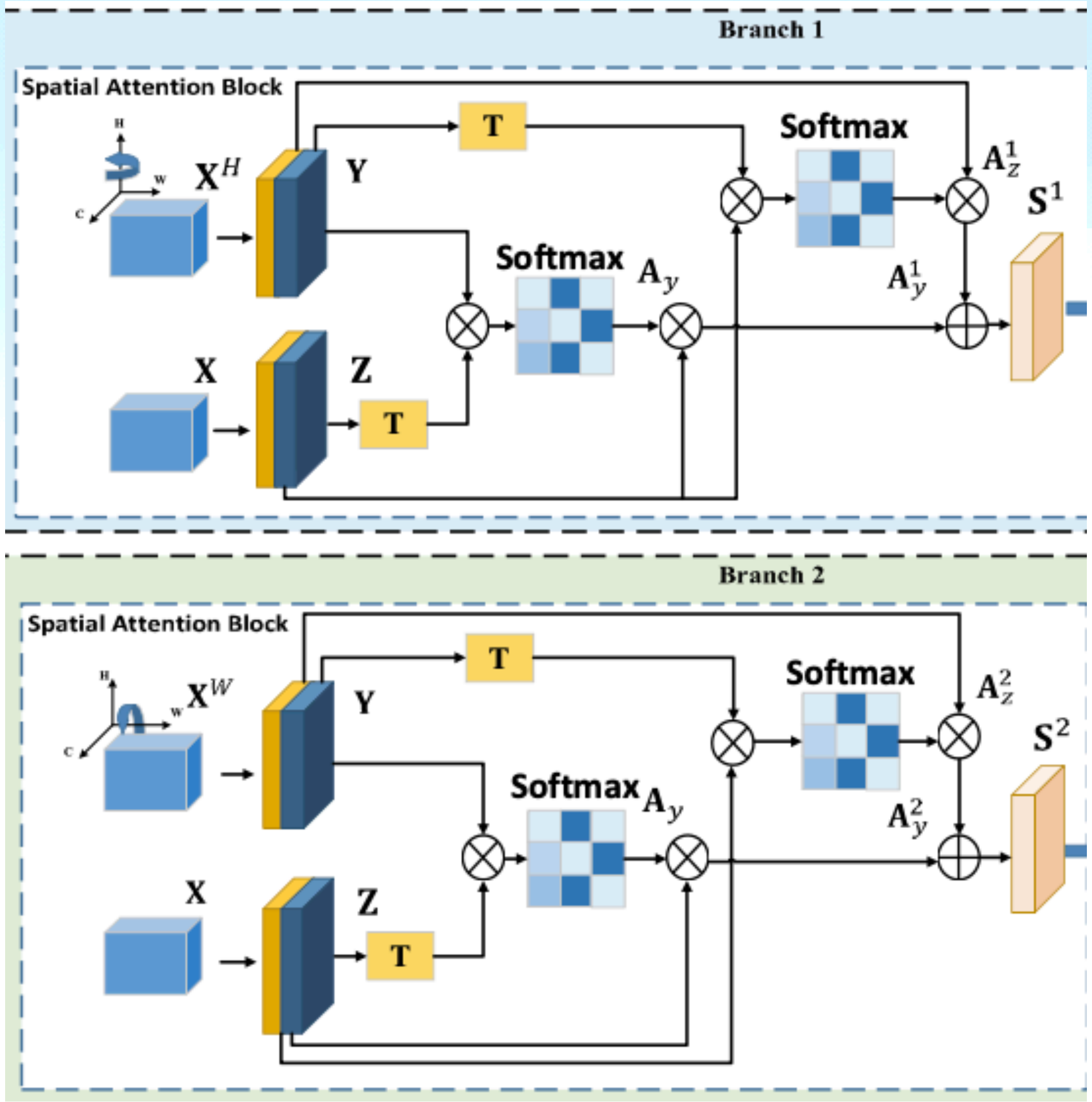
BIDIRECTIONAL ATTENTION MODULE

- The focus is to make sure that the computer face attention to all parts of the face equally.
- BAM incorporates spatial and channel attention blocks to highlight informative spatial locations and feature channels.



SPATIAL ATTENTION BLOCK

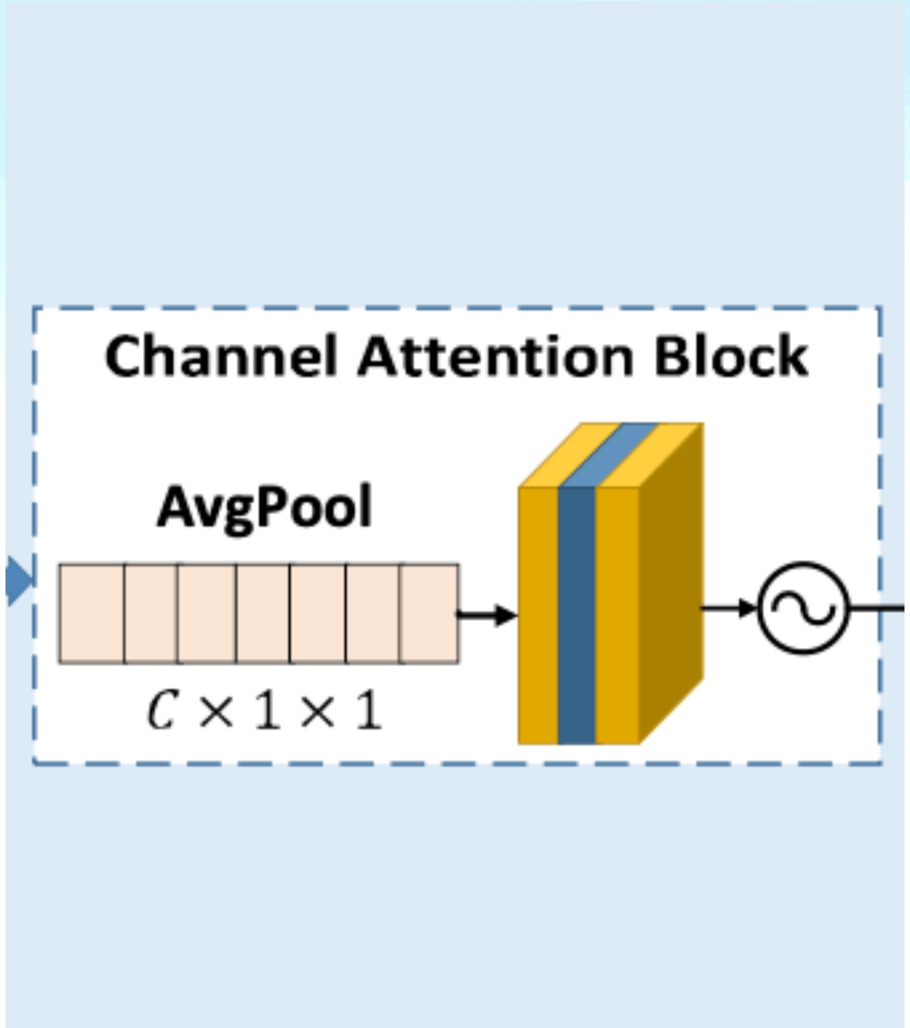
- The Spatial Attention Block (SAB) in the Bidirectional Attention Module (BAM) learns bidirectional attention maps by capturing inter-dimensional feature dependencies to highlight informative spatial locations for feature learning.



- It performs matrix multiplication between the transpose of the rotated feature map and the original feature map, followed by a softmax function to compute the spatial attention map.
- This enhances the network's ability to focus on relevant spatial regions for masked face recognition.

CHANNEL ATTENTION BLOCK

- The Channel Attention Block (CAB) in the Bidirectional Attention Module (BAM) calibrates spatial attention maps generated by the SAB by modelling interdependencies between feature channels.



- It applies global average pooling, followed by a channel-downscaling convolutional layer and ReLU activation, to assign higher weights to informative feature channels.

- This process enhances the network's ability to focus on relevant feature channels for masked face recognition.

TRAINING SETTINGS

- To ensure a fair comparison, we trained our model and other models from scratch using the same backbone architecture called ResNet50-IR.
- Loss functions are used namely softmax
- The learning rate used is 0.01 initially. After specific epochs, the learning rate is reduced by a factor of 5.

- During training, we processed batches of 128 images at a time.

- The momentum value used is 0.9 and the weight decay used is 5e-4.

- We implemented our method using the Tensorflow framework

ROC CURVE

- ROC stands for Receiver Operating Characteristic. It is a visual representation of a model's performance at classifying binary outcomes.

- In the graph shown, it is evaluating models for facial recognition.

- In the graph shown, x-axis denotes FPR and y-axis denotes TPR.
- FPR is the proportion of negative cases that were classified positive and TPR is the proportion of positive cases that were correctly classified as positive.
- A perfect classifier would have TPR of 100% and FPR of 0%.
- The area under the ROC curve is a numerical measure of the classifier's performance. Larger the area, better is the classifier's performance.

VISUALISATION OF ATTENTION MAPS LEARNED BY BAM

- Attention maps are a way to see what parts of an image a model pays attention to when making a decision.
- The brighter an area is in an attention map, the more attention the model is paying to that area.
- By visualising attention maps, we can gain insights into how different models work and identify areas where they may be making mistakes.

Results

Model Training

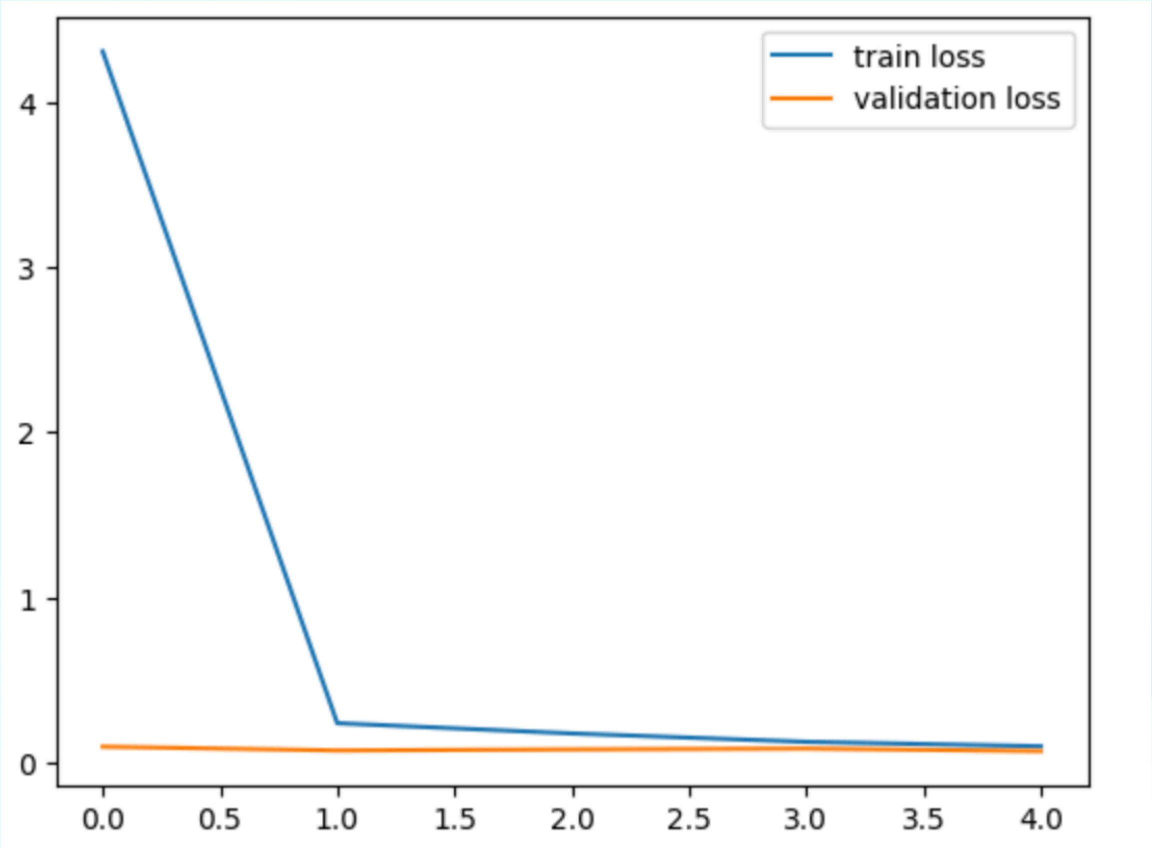
```
Epoch 1/5
48/48 [==============================] - 453s 9s/step - loss: 4.3076 - acc: 0.9058 - val_loss: 0.0980 - val_acc: 0.9762
Epoch 2/5
48/48 [==============================] - 492s 10s/step - loss: 0.2403 - acc: 0.9490 - val_loss: 0.0751 - val_acc: 0.9749
Epoch 3/5
48/48 [==============================] - 507s 11s/step - loss: 0.1782 - acc: 0.9528 - val_loss: 0.0813 - val_acc: 0.9788
Epoch 4/5
48/48 [==============================] - 492s 10s/step - loss: 0.1274 - acc: 0.9634 - val_loss: 0.0867 - val_acc: 0.9709
Epoch 5/5
48/48 [==============================] - 490s 10s/step - loss: 0.1005 - acc: 0.9743 - val_loss: 0.0712 - val_acc: 0.9689
```

Model Testing

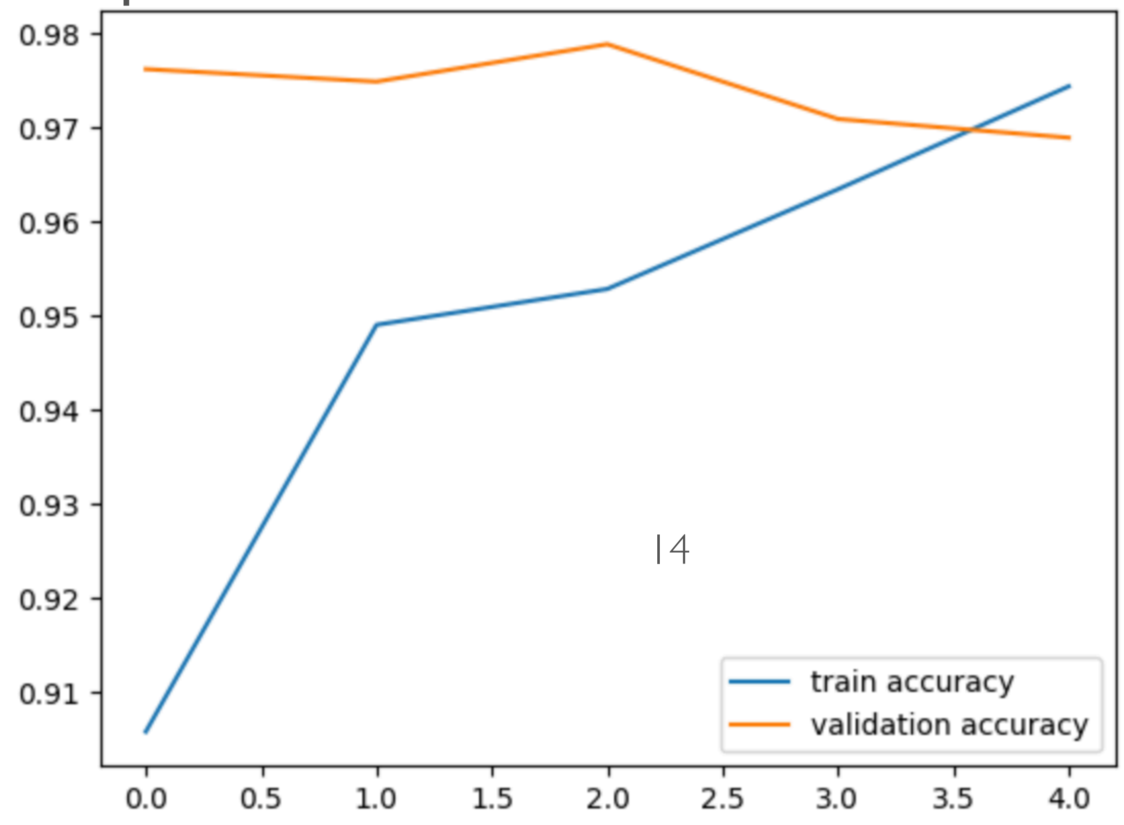
```python
loss, accuracy = model_y.evaluate(X_test, Y_test)
print('Test Accuracy =', accuracy)
```

```
48/48 [==============================] - 79s 2s/step - loss: 0.0712 - acc: 0.9689
Test Accuracy = 0.9688947796821594
```

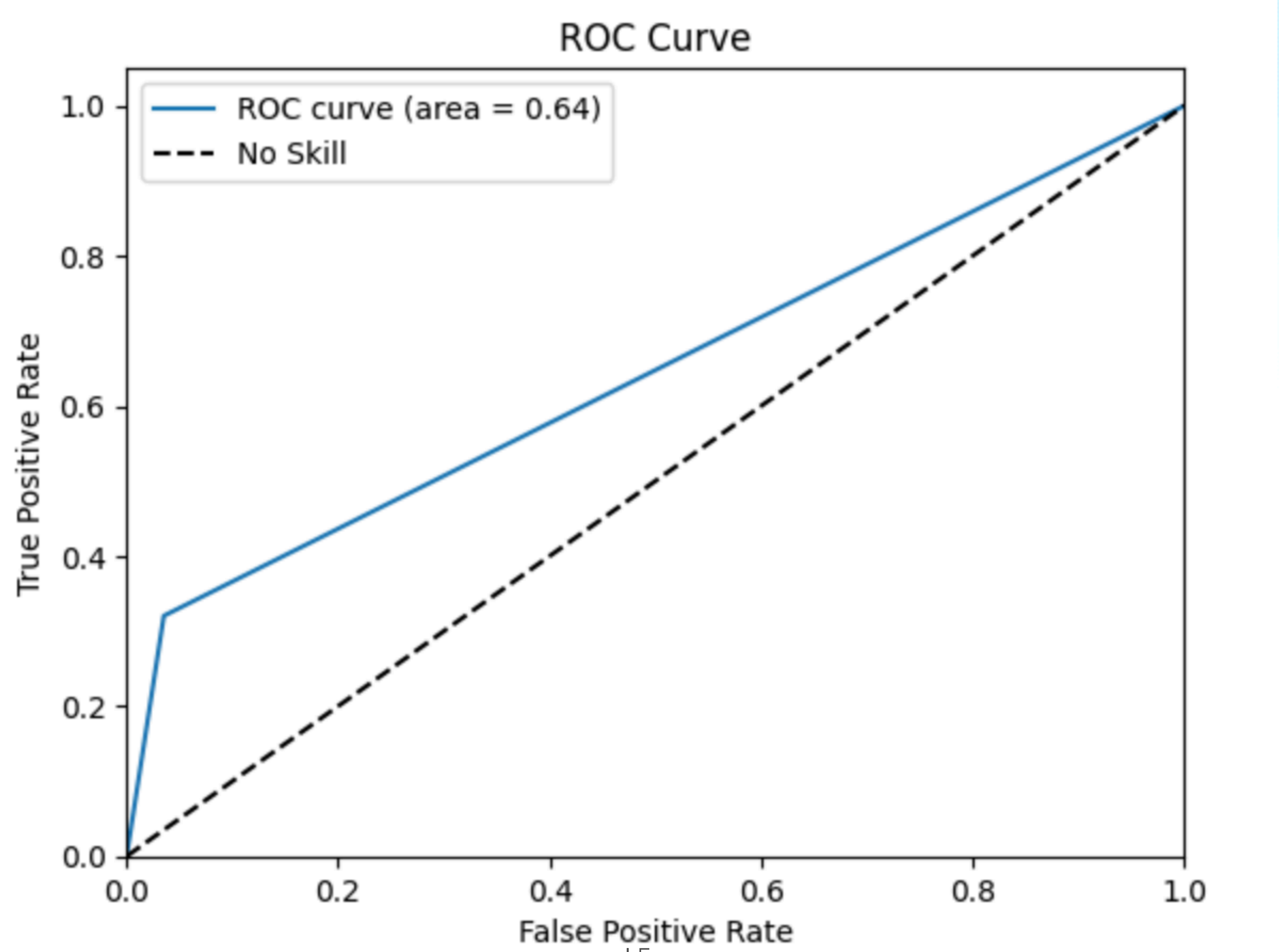
Loss Graph



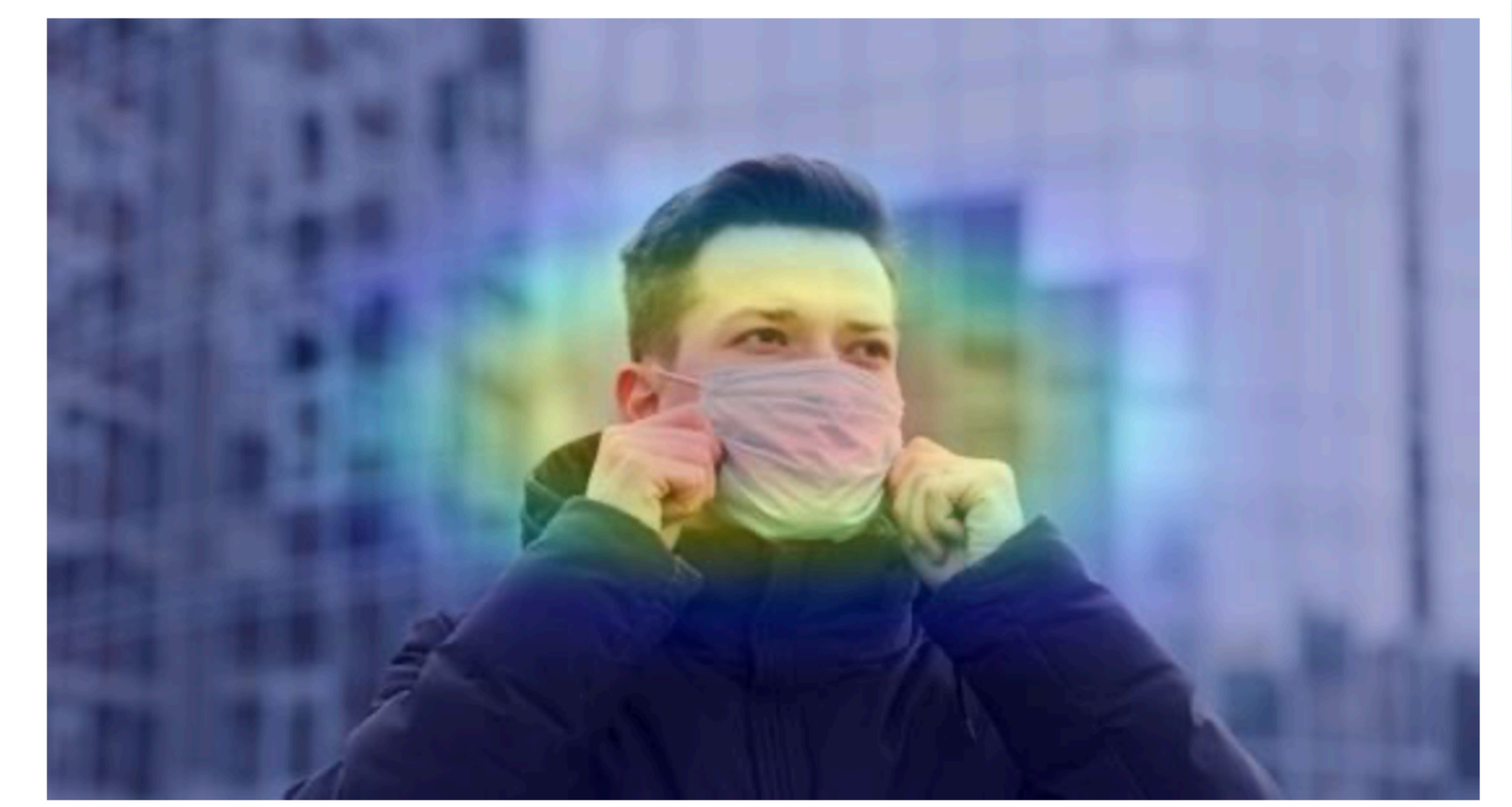
Accuracy Graph



ROC Curve for BAM



Visualisation of attention maps



THANK YOU!