

Autism Spectrum Disorder - Using Eye Scan path

16 Feb 2025

Data Augmentation

Vaishnav Krishna P

Report 2

Outline

1. Introduction
2. Objective
3. Methodology
4. Results
5. Conclusion

Introduction

The Orginal Dataset includes 547 images with 328 non ASD(Autistic Spectrum Disorder) and 219 ASD images. From the baseline accuracy testing very clear that this many images are not enough to build a good model. So, we apply Data Augmentation techniques.

Data Augmentation Techniques are used to generate synthetic samples by image transformations. Some of the Data Augmentation techniques includes,

1. Horizontal flipping
2. Gaussian Noise
3. Vertical flipping
4. Hue/Saturation
5. Rotation

Objective

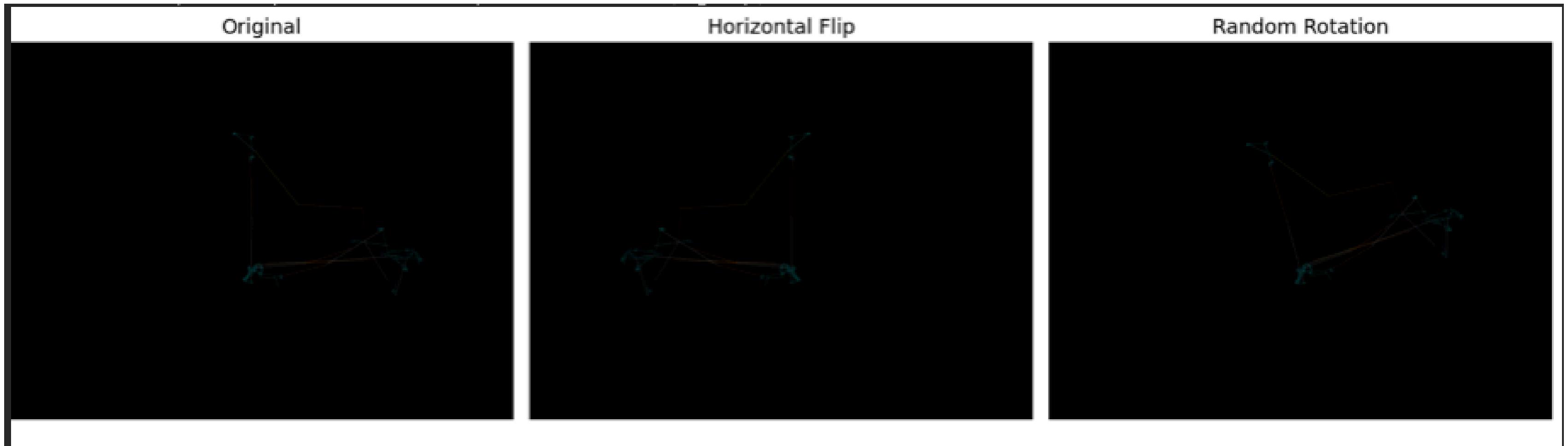
The objective of this experimental setup is to apply various Data augmentation techniques to enlarge our dataset from 547 to 2564 around 4-5 times the orginal dataset by applying various data augmentation techniques.

Note: Since we are dealing with the eye track scan path, we cannot perform horizontal flipping, vertical flipping, large rotation, etc., because it changes the meaning of the data and train model with false data.

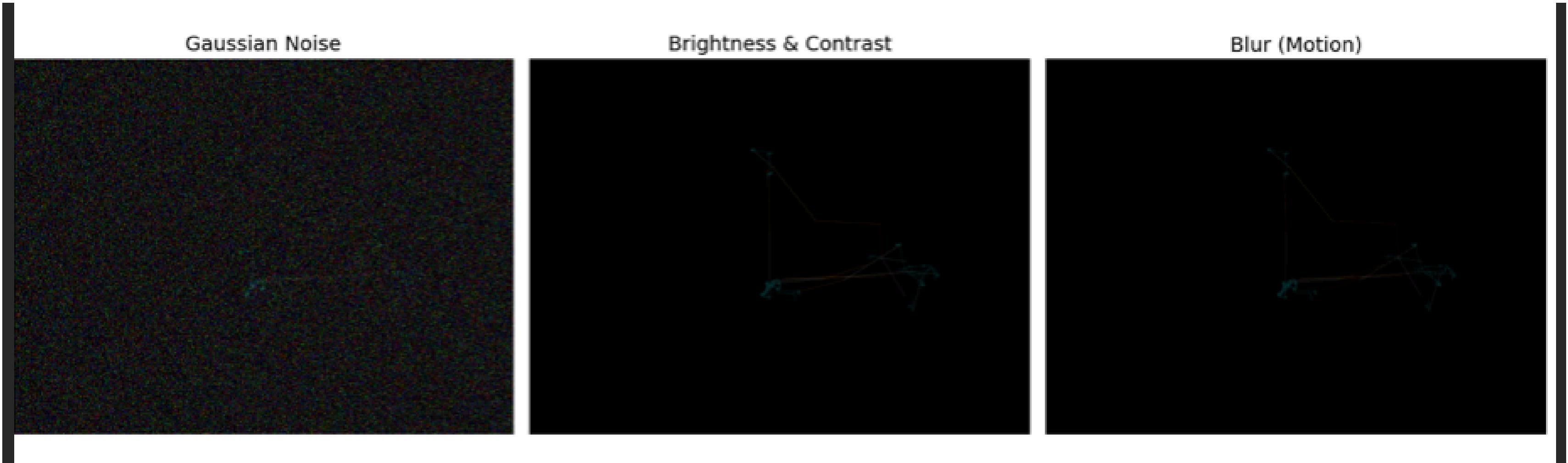
- Train the various pre-trained models with the data augmented data,
 - MobileNetv2
 - ResNet50
 - DenseNet121
 - EfficientNetv2

Methodology

Data Augmentation Techniques,



Methodology



Methodology

- The offline Data augmented dataset is generated from the original dataset; the no of images after the augmentation was about 2564 images.
- each image of about 4-5 augmentation

The data augmentation techniques applied in our case are,

1. slight rotation
2. small horizontal shift
3. small vertical shift
4. small zoom
5. slight brightness change

Experimental Setups

Entire experiments were conducted on Google Colab with a free GPU resources. The parameters used across the experiments includes,

- Test size: 0.2
- Image size: 128*128*3 / 224*224*3(original image: 640*480*3)
- Batch size: 16
- Train, Test image size: 2052,512
- hidden layer: 128 neurons(rectified linear unit activation function)
- Dropout(0.5)
- output layer: 1(sigmoid activation function)
- epochs: 20(applied early stopping - callback)
- metrics - accuracy, precision, recall, f1-score, MCC

Base accuracy - Data Augmentation

BASE MODEL EVALUATION(AUGMENTED DATA)					
Model	Training Accuracy	Testing Accuracy	Precision	Recall	MCC
ResNet50(128*128)	93.45	91.6	91.71	86.76	0.824
ResNet50(224*224)	92.44	90.42	93.78	81.37	0.8009
MobileNetv2(128*128)	94.48	92.57	98.82	82.35	0.8493
MobileNetv2(224*224)	94.08	93.36	97.2	85.78	0.863
DenseNet121(128*128)	88.78	87.89	89.88	78.43	0.7762
DenseNet121(224*224)	87.57	85.74	87	75.49	0.7
EfficientNetV2(128*128)	84.87	87.5	90.23	76.96	0.7386
EfficientNetV2(224*224)	85.04	86.13	88.88	74.51	0.7094

Other Experimental results

The ROI(Region of interest) & Skeleton image processing techniques were effective techniques for the original dataset. But, when applied after Data augmentation the accuracy drops.

OTHER EXPERIMENTAL RESULTS(Skeleton & ROI)					
Model	Training Accuracy	Testing Accuracy	Precision	Recall	MCC
MobileNet + 128*128 + skeleton	83.72	79.49	76.76	69.6	0.5671
densenet + skeleton +128*128	77.49	76.76	75.76	61.27	0.5058
MobileNet + ROI + 128*128	72.76	75.2	74.4	50.81	0.4492

Optimisation of parameters

The MobileNetv2 outperform all the other models with an accuracy of 98.02 after adjusting some of the parameters,

- Test size: 0.2
- Random seed: 41
- Image size: 224*224*3
- Batch size: 16
- Train, Test image size: 2052,512
- epochs: 20(applied early stopping - callback)

Accuracy - metrics

Accuracy: 0.98046875

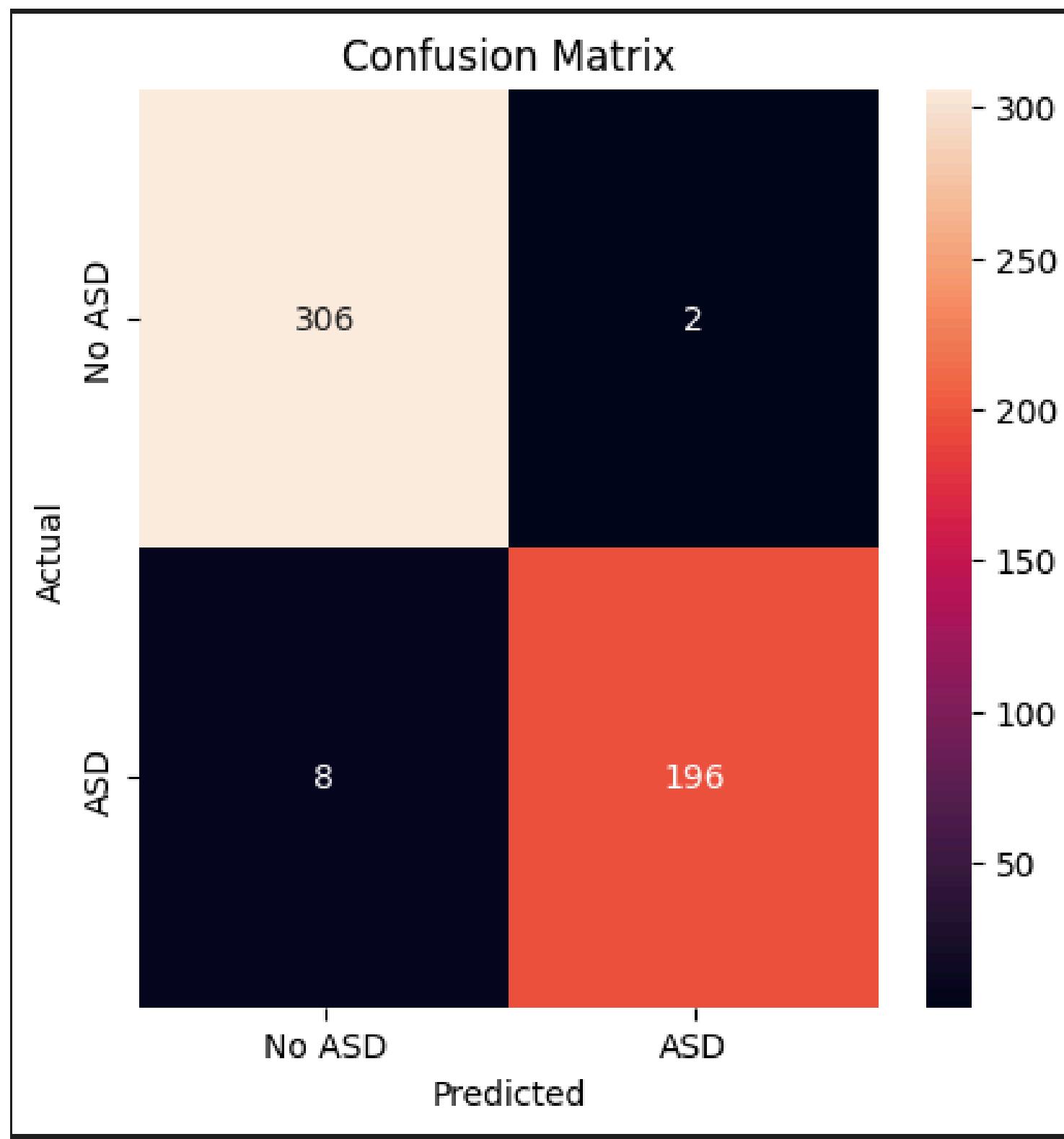
Precision: 0.98989898989899

Recall (Sensitivity): 0.9607843137254902

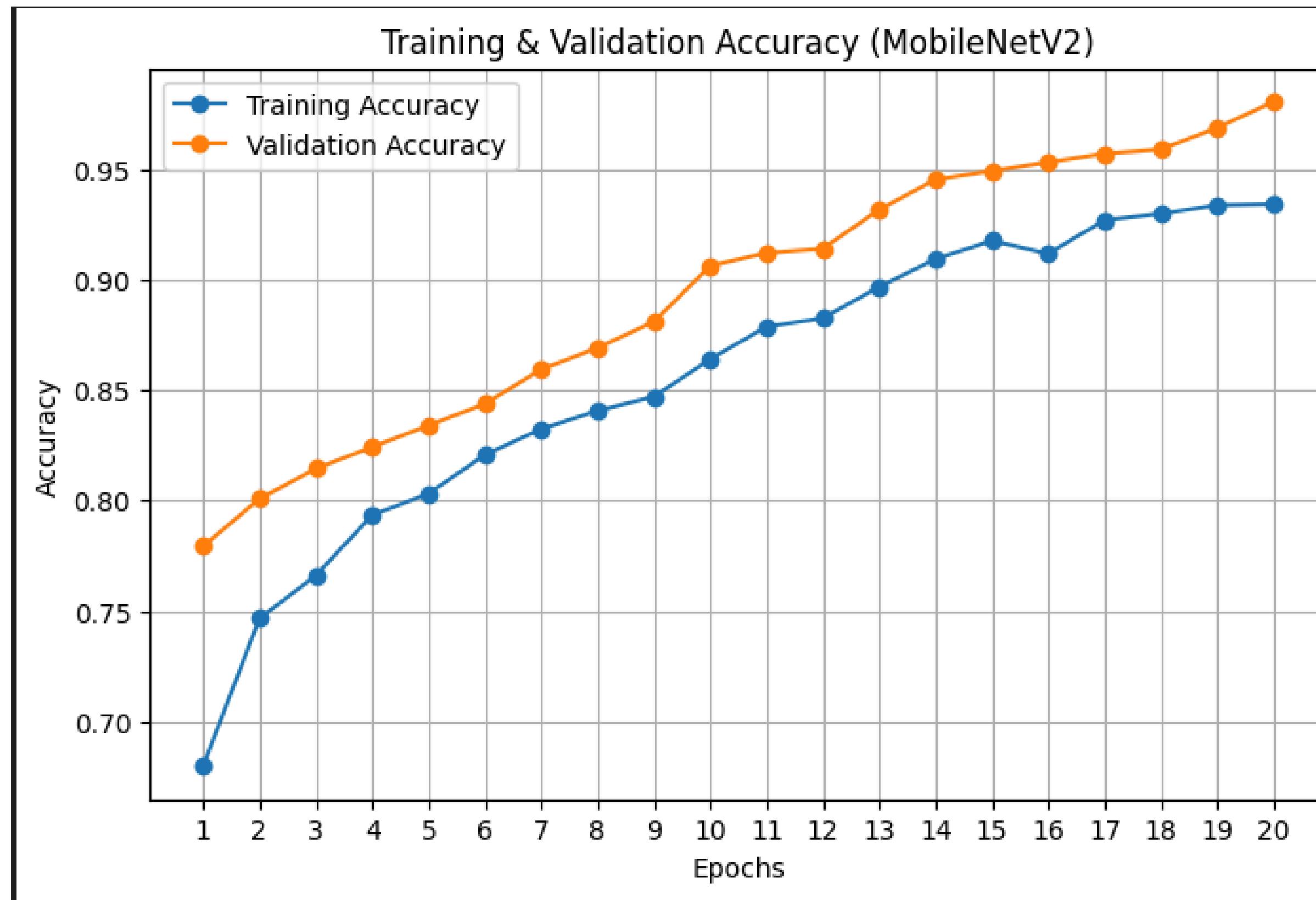
F1 Score: 0.9751243781094527

MCC: 0.9593426731691362

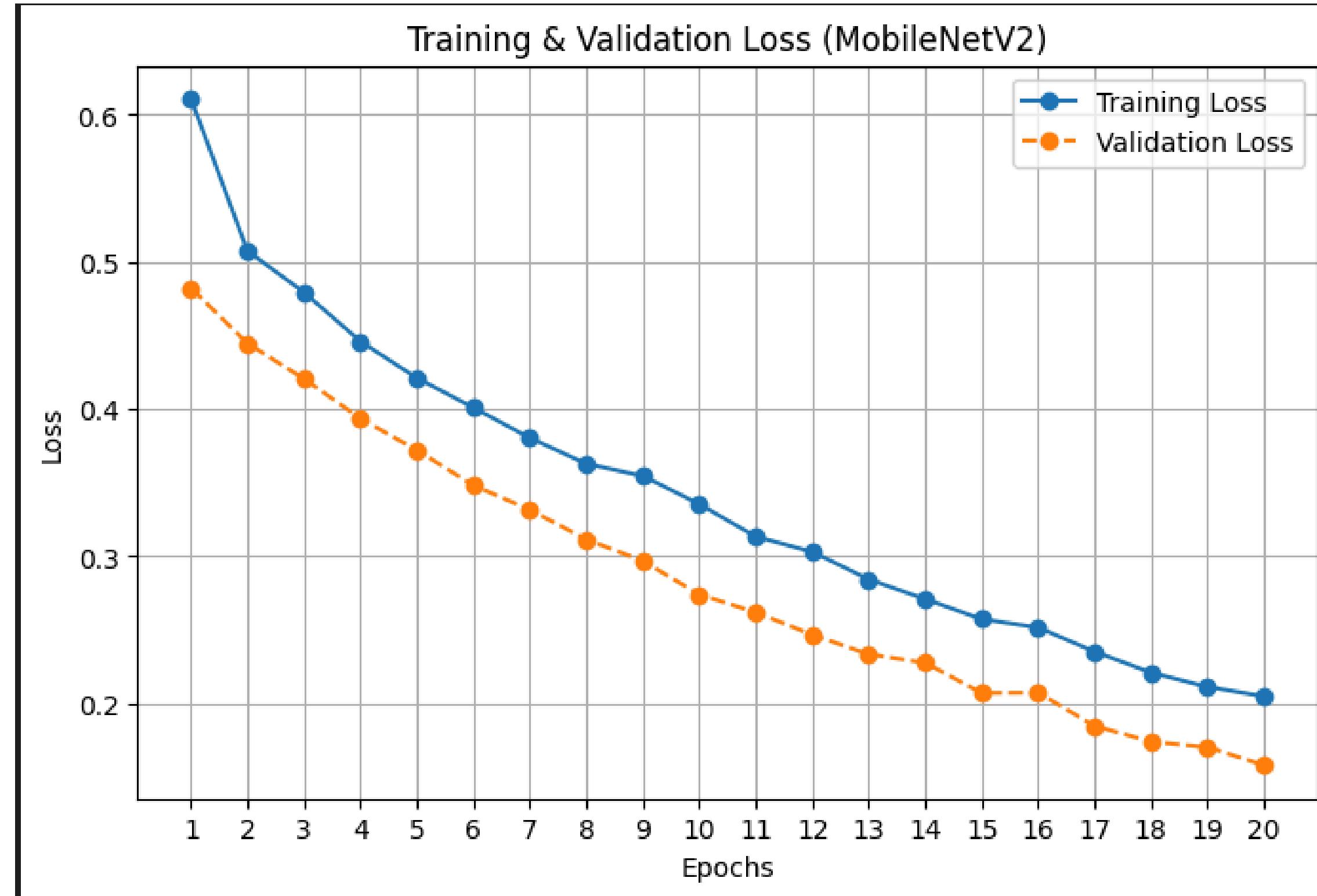
Accuracy - confusion metrics



Training & Validation accuracy curve



Training & Validation loss curve



Thank you