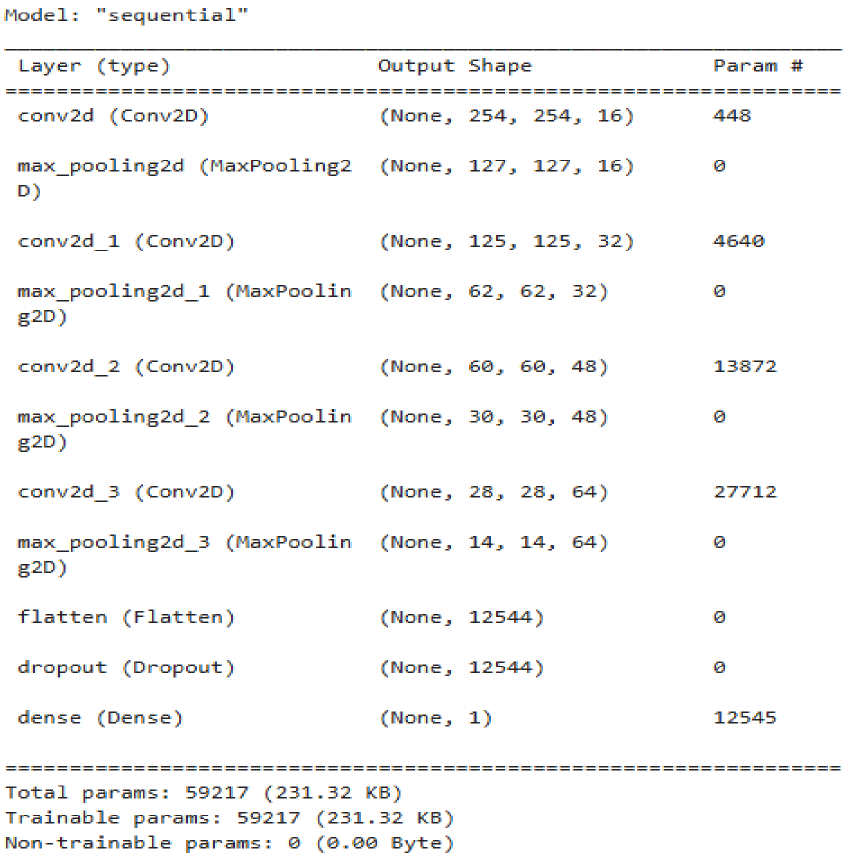
Deepfake Video Detection Model Using CNN

Introduction

Deepfake videos have become a significant challenge in today's digital world, posing threats in various sectors like media, security, and entertainment. Detecting these videos accurately is crucial for mitigating their negative impact. In this project, I developed a Deepfake Video Detection Model using Convolutional Neural Networks (CNN). This model is designed to distinguish between real and deepfake videos with a high degree of accuracy. Various techniques and model configurations were explored to improve the model's performance.

Model Overview

The final model used in this project is a CNN-based architecture, specifically a **Sequential Model**. The model architecture consists of several convolutional layers followed by max-pooling, dropout layers, and a dense layer for the final classification. The model was designed to process image data extracted from videos and classify them as either "real" or "deepfake."



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 16)      448

 max_pooling2d (MaxPooling2  (None, 127, 127, 16)      0
 D)

 conv2d_1 (Conv2D)           (None, 125, 125, 32)      4640

 max_pooling2d_1 (MaxPoolin  (None, 62, 62, 32)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 60, 60, 48)        13872

 max_pooling2d_2 (MaxPoolin  (None, 30, 30, 48)        0
 g2D)

 conv2d_3 (Conv2D)           (None, 28, 28, 64)        27712

 max_pooling2d_3 (MaxPoolin  (None, 14, 14, 64)        0
 g2D)

 flatten (Flatten)           (None, 12544)             0

 dropout (Dropout)           (None, 12544)             0

 dense (Dense)               (None, 1)                 12545

=================================================================
Total params: 59217 (231.32 KB)
Trainable params: 59217 (231.32 KB)
Non-trainable params: 0 (0.00 Byte)
```

This model consists of four convolutional layers and max-pooling layers, which allow the network to learn hierarchical features from the images. A dropout layer is used to prevent overfitting, followed by a dense layer for classification.

Performance Metrics

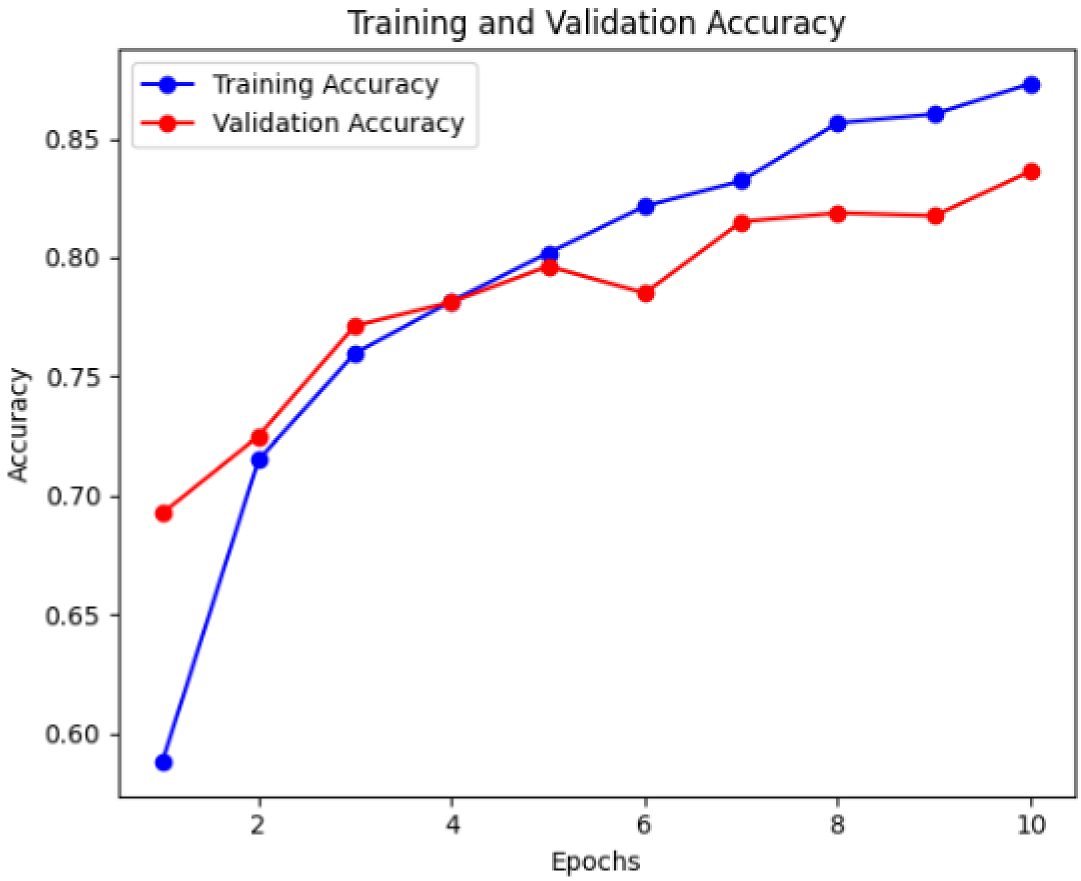
The model's performance was evaluated on the test dataset, achieving the following results:

- **Test Loss:** 0.3835
- **Test Accuracy:** 83.63%

These results indicate that the model is able to successfully classify real and deepfake videos with a high level of accuracy. Below are the plots of accuracy and loss during training:

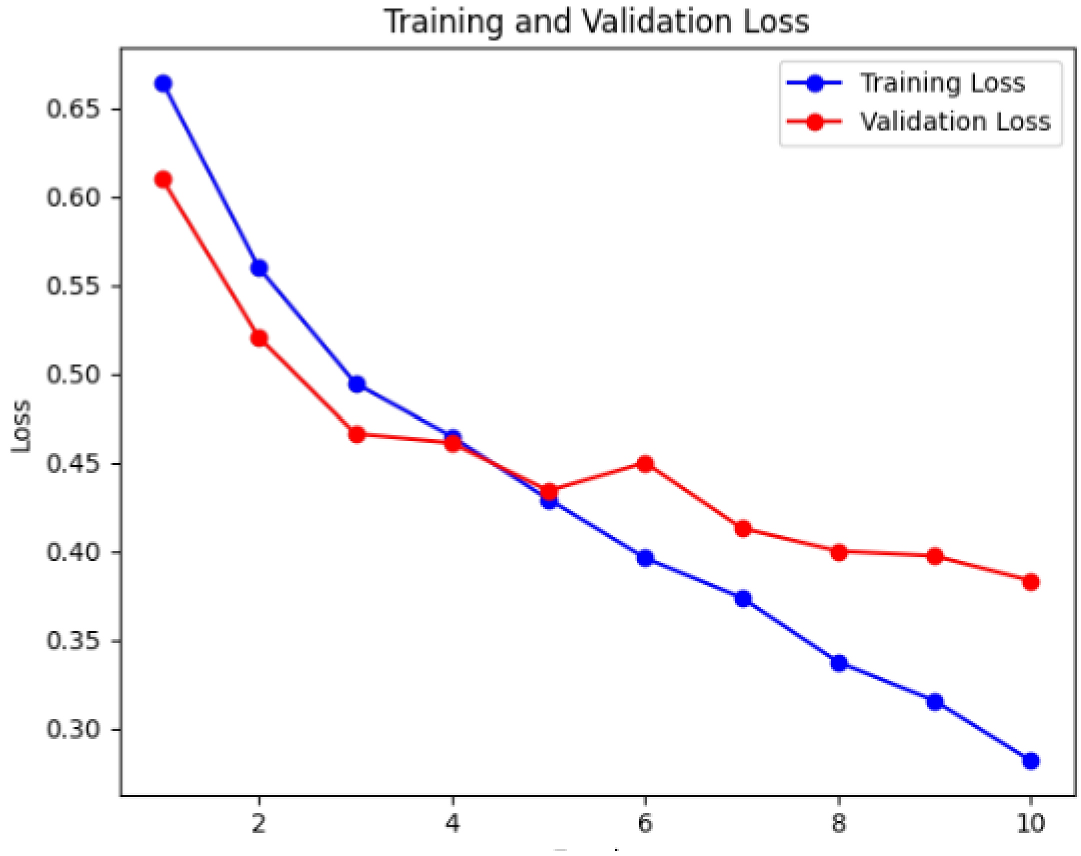
Accuracy Plot

The accuracy plot demonstrates the model's performance over the epochs, showing how the accuracy improves as training progresses. The final accuracy of the model was 83.63%, which is a promising result for this type of task.



Loss Plot

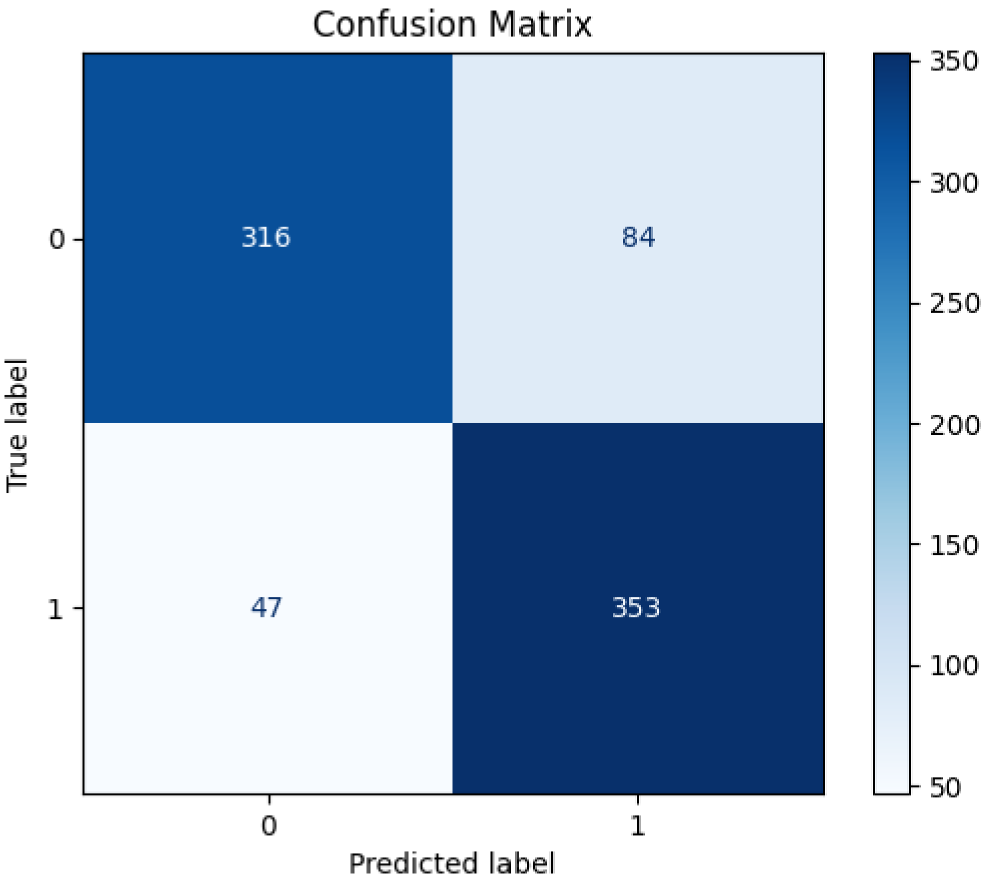
The loss plot shows the loss value decreasing over time, indicating that the model was successfully learning during training. A lower loss value suggests that the model was able to reduce errors and make more accurate predictions.



Confusion Matrix

The confusion matrix below provides a detailed breakdown of the model's classification performance. It shows the number of true positives (real images classified as real), false positives (deepfake images classified as real), true negatives (deepfake images classified as deepfake), and false negatives (real images classified as deepfake).

The confusion matrix helps in understanding where the model is making errors and provides insights into how it can be further improved.



Conclusion

In this project, I successfully built a CNN-based model for detecting deepfake videos. After experimenting with various architectures and training configurations, I was able to achieve an accuracy of 83.63% on the test set. This demonstrates the potential of CNNs for video manipulation detection. The model's performance can be further improved by fine-tuning hyperparameters, adding more layers, or incorporating additional techniques such as transfer learning.