Deep Fake Video Classification Using Deep Learning Techniques

George Washington University

Final Term Project - Individual Final Report

Author

Vishal Bakshi

Department of Data Science
George Washington University
Vishal.bakshi@gwu.edu

Mentor

Prof. Amir Jafari

1 Introduction

The rapid growth of multimedia content has transformed how we communicate and share information. With approximately 3.2 billion images and 720,000 hours of video shared daily, this explosion of content brings both opportunities and challenges. Among these challenges is the rise of misinformation and deepfake manipulation.
Deepfakes—synthetically generated or altered videos and images—present a significant threat to digital integrity. They have been exploited in various harmful ways, such as:

- **Political Propaganda**: Creating fake videos of politicians making inflammatory remarks.
- **Financial Fraud**: Facilitating identity theft through morphed images.
- **Misinformation Campaigns**: Spreading false narratives with highly convincing fake visuals.

The societal impact of deepfakes is vast, including eroding public trust, financial losses, and reputational harm. Detecting and combating these manipulations is essential to reducing their damaging effects.
This project addresses this issue by developing advanced deep learning models for forgery detection. Utilizing state-of-the-art datasets like CelebDF V2 and cutting-edge architectures, we aim to create a reliable system for identifying and mitigating manipulated multimedia content.

Folder Name	Subfolders	Description
Train	Real, Fake	This directory has 70% of train and test videos
Test	Real, Fake	This directory has 15% of train and test videos
Val	Real, Fake	This directory has 15% of train and test videos
Celeb-real	NA	All real videos
Celeb-synthesis	NA	All deepfake videos
YouTube-real	NA	All real videos collected from YT
Train_ela	Real, Fake	This directory has 70% of ELA converted train and test videos
Test_ela	Real, Fake	This directory has 15% of ELA converted train and test videos
Val_ela	Real, Fake	This directory has 15% of ELA converted train and test videos

2 Description of Individual Work

This research involved the development of 2D, (2+1D) CNN architectures. A thorough Error Level Analysis was conducted to understand and address model shortcomings. The project also focused on improving the efficiency of data processing by restructuring and developing code for a Streamlit application and exploring various TensorFlow data loaders. To enhance the accuracy of facial recognition, MTCNN was integrated to isolate facial regions, allowing the model to focus its training on relevant features.

3 Description of Individual Work

- There are 3 major areas that I have worked on this project
    - Data Loading
    - 2+1D CNN, and InceptionV3+ LSTM architecture
    - Hyperband Tuning
    - Error Level Analysis

- **2+1D CNN**
    - I developed a 2+1D CNN model designed to perform convolutions on 2D image frames while capturing temporal information across video sequences.
    - Using TensorFlow's video classification code as a starting point, I adapted the methodology to suit a binary classification task instead of the original multiclass classification. To address the challenge of vanishing gradients, particularly for the minority class, I integrated residual connections into the model architecture.
    - Despite these efforts, the model consistently produced recall and precision scores of 0 for the "real" class, leading to its eventual exclusion from the final system.

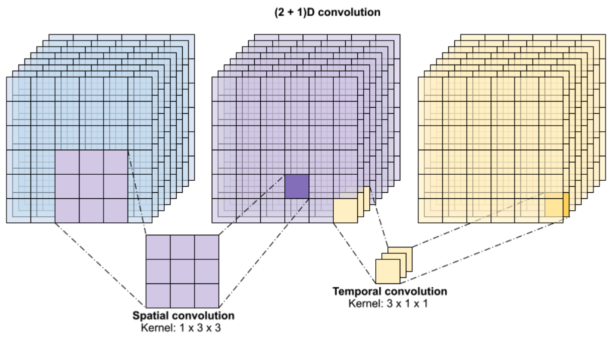


Fig 1. 2+1D CNN Architecture

- **InceptionV3+LSTM**
    - Bharat and I collaboratively worked on the InceptionV3 + LSTM model. My contribution involved leveraging InceptionV3 to extract 2048-dimensional features from individual frames, which were then passed to an LSTM model to capture temporal dependencies across the frames.
    - To ensure the LSTM could effectively model the temporal dynamics, I ensured that no frames were skipped during feature extraction, aiming to preserve crucial sequential information.
    - However, the model faced issues, as the recall for the "real" class remained at 0. Despite observing the training sequences and finding no signs of overfitting, I suspect the problem lies with the data loader (TensorFlow Generator).

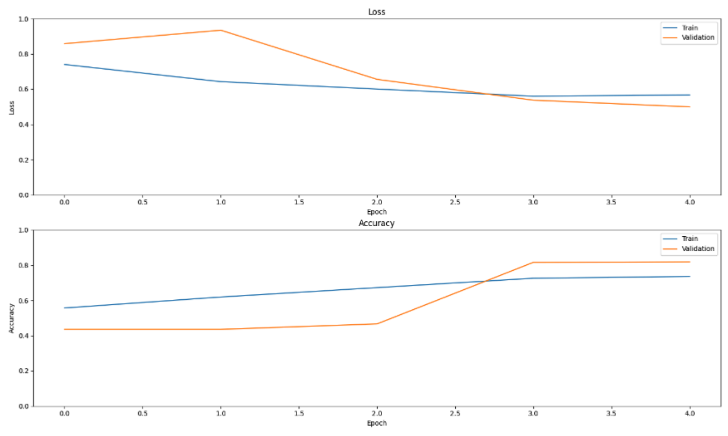
- **Hyperband Tuning**
    - I collaborated on Bharat's code for the CNN-RNN architecture and applied Hyperband tuning to optimize its performance. However, the results were suboptimal, leading to the decision to discard this approach.

```
# Hyperband tuner
tuner = kt.Hyperband(
    build_gru_model_with_tuning,
    objective="val_accuracy",
    max_epochs=10,
    factor=3,
    directory="hyperband_tuning",
    project_name="video_classification")
```

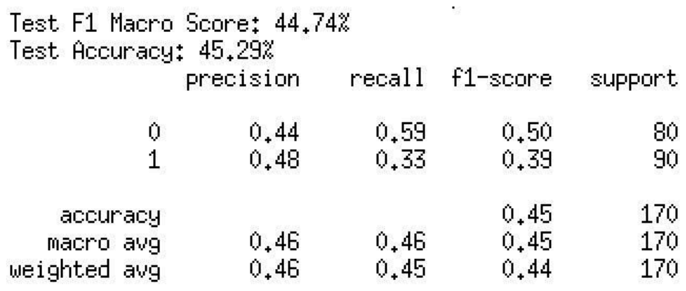
- **Error Level Analysis**
    - I explored Error Level Analysis (ELA) based on insights from a research paper focused on forensic data techniques. ELA works by compressing an image to 95% of its original quality and analyzing pixel intensity variations.
    - The key idea is that an authentic image, when compressed, shows minimal differences in pixel intensities. However, manipulated or fake images reveal discrepancies, particularly in regions that have been altered. For instance, in face-swapped images, the manipulated region typically appears darker or lighter compared to the surrounding pixels.
    - As part of this effort, I collaborated with Nena Beecham to train her model using ELA-converted images. This approach yielded promising results, demonstrating the effectiveness of ELA in detecting manipulated content.

- **Streamlit**
    - I structured the codebase to ensure it adhered to a clean and modular design using Classes, enhancing maintainability and readability. Additionally, I integrated functionality within the Streamlit application to display video files directly, providing a user-friendly and interactive interface.
    - To streamline usability further, I developed a script that, when executed, downloads all the necessary pre-trained models along with a curated set of fake and real video examples into the current working directory. This setup simplifies the deployment process, enabling users to quickly access and test the system without additional manual setup.

4 Results



- **Observation**
  - During model training, no signs of overfitting were observed, as the training and validation loss followed a consistent pattern. However, despite this, the model consistently produced a recall and precision of 0 for the “real” class.
  - This indicates that the model struggled to correctly identify instances of the “real” class, failing to make accurate predictions even when the training appeared stable.



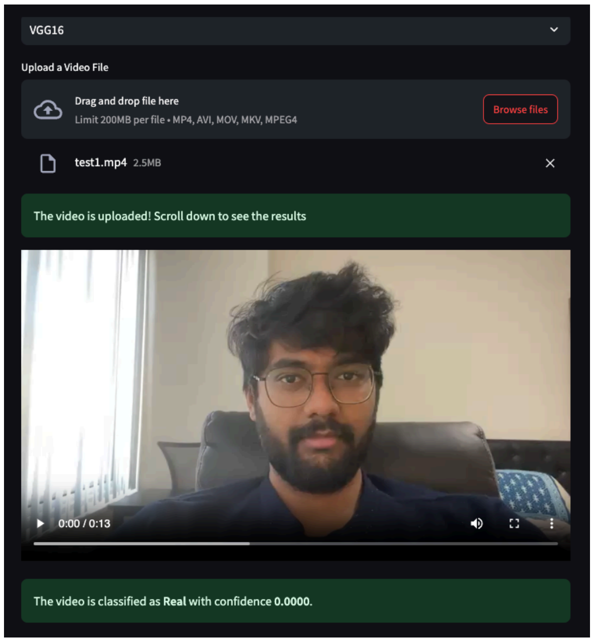
```
Test F1 Macro Score: 44.74%
Test Accuracy: 45.29%
              precision    recall  f1-score   support

           0       0.44      0.59      0.50        80
           1       0.48      0.33      0.39        90

    accuracy                           0.45       170
   macro avg       0.46      0.46      0.45       170
weighted avg       0.46      0.45      0.44       170
```

Fig 3. Model Evaluation Metrics using Nena’s InceptionV3 Model on ELA Converted Frames

- **Observation**
  - This is the evaluation metrics from the InceptionV3 model for ELA converted images. This is the raw image that has been passed to the model for training, if we preprocess the image properly we might gain more precise results.



VGG16

Upload a Video File

Drag and drop file here

Limit 200MB per file • MP4, AVI, MOV, MKV, MPEG4

The video is uploaded! Scroll down to see the results

The video is classified as **Real** with confidence **0.0000**.

Fig 4. Streamlit Output

5 Summary and Conclusions

Based on my research and experimentation, Error Level Analysis (ELA) shows significant potential for detecting manipulated content. ELA analyzes pixel intensity variations by compressing an image to 95% quality and highlighting inconsistencies. In authentic images, compression results in uniform pixel intensity differences. However, in manipulated images, such as those with face swaps, altered regions often stand out due to mismatched pixel intensities. These differences, such as darker or lighter patches in manipulated areas, provide critical clues for detecting forgery.

Given its ability to reveal subtle patterns in altered images, ELA could be a valuable tool for improving the precision and reliability of forgery detection systems.

6 Code Percentage

For this project, approximately 70% of the code was adapted from various online resources, including platforms like TensorFlow documentation, Reddit discussions, and Quora posts. These references provided a solid foundation and guidance for implementing specific techniques and overcoming technical challenges. The remaining 30% of the code was custom-written, focusing on tailoring the algorithms, optimizing the workflow, and integrating the components to meet the specific requirements of the project. This includes modifications to pre-existing code, implementing novel solutions, and adding features to enhance functionality.

By leveraging available resources and combining them with custom development, I was able to efficiently build a robust system while ensuring it met the project's unique needs.

7 References

1. https://fotoforensics.com/tutorial-ela.php
2. https://www.nature.com/articles/s41598-023-34629-3
3. https://ieeexplore.ieee.org/abstract/document/9441519