

Deepfake Detection Using Multimodal Analysis

Project Timeline

CSE 299.5 - Spring 2025

Group: 2

Members:

Atique Shahrier Chaklader (2132882642)

Md. Wailul Sakib (2212570042)

Samin Yeasar Badhon (2212208042)

Week 1-2: Dataset Collection and Preprocessing

Goals:

Collect and preprocess datasets for visual, audio, and temporal analysis.

Set up the development environment.

Tasks:

Dataset Collection:

Download datasets:

DFDC (DeepFake Detection Challenge Dataset)

FakeAVCeleb Dataset

Celeb-DF

VoxCeleb

OpenSLR

Organize datasets into folders (real vs. fake).

Preprocessing:

Visual Data:

Extract frames from videos using OpenCV.

Resize and normalize frames for model input.

Audio Data:

Convert audio files to spectrograms using Librosa.

Temporal Data:

Extract sequences of frames for motion analysis.

Environment Setup:

Install required libraries (TensorFlow, PyTorch, OpenCV, Librosa, etc.).

Set up Google Colab for training models.

Week 3-5: Model Training

Goals:

Train models for visual, audio, and temporal analysis.

Tasks:

Visual Analysis:

Train a CNN (e.g., MobileNet, EfficientNet) on the DFDC and Celeb-DF datasets.

Use pre-trained models for faster convergence.

Evaluate model performance using metrics like accuracy and F1-score.

Audio Analysis:

Train a CNN on spectrograms to detect synthetic voices.

Use the VoxCeleb and OpenSLR datasets for training.

Temporal Analysis:

Train an RNN or Transformer on sequences of video frames.

Focus on detecting motion inconsistencies.

Model Evaluation:

Test each model on a validation set.

Fine-tune hyperparameters for better performance.

Week 6: Multimodal Fusion

Goals:

Combine outputs from visual, audio, and temporal models for robust detection.

Tasks:

Fusion Techniques:

Implement late fusion (combining model outputs at the decision level).

Experiment with attention mechanisms for better integration.

Evaluation:

Test the fused model on a validation set.

Compare performance with single-modality models.

Week 7: App Development (Backend and GUI)

Goals:

Develop the backend and frontend for the deepfake detection app.

Tasks:

Backend:

Create a Flask API to handle video uploads and process them using the trained models.

Integrate the multimodal fusion model into the backend.

Frontend:

Web App:

Use Flask to create a simple HTML interface for uploading videos and displaying results.

Desktop App:

Use Tkinter to create a GUI for video upload and result display.

Explainable AI:

Add visual heatmaps (using Grad-CAM) and spectrogram highlights to the results.

Week 8: Testing and Deployment

Goals:

Test the app and deploy it for use.

Tasks:

Testing:

Test the app with real and fake videos.

Evaluate accuracy, precision, recall, and F1-score.

Deployment:

Web App:

Host the Flask app on Heroku or Streamlit Cloud.

Desktop App:

Package the app using PyInstaller.

Week 9: Final Report and Presentation

Goals:

Prepare the final report and presentation.

Tasks:

Final Report:

Document the entire project, including:

Problem statement, methodology, and implementation.

Results and evaluation metrics.

Challenges and solutions.

Include screenshots of the app and sample outputs.

Presentation:

Create a PowerPoint presentation summarizing the project.

Highlight key features, results, and future work.

Key Deliverables:

Functional App:

A web or desktop app for deepfake detection.

Detailed Reports:

Include visual heatmaps, spectrogram highlights, and confidence scores.

Open-Source Codebase:

Share the code on GitHub for academic use.

Future Work:

Incorporate More Modalities:

Add text analysis for detecting inconsistencies in captions or subtitles.

Improve Robustness:

Train the models on adversarial examples to improve robustness.

References:

DFDC Dataset: <https://ai.facebook.com/datasets/dfdc/>

FakeAVCeleb Dataset: <https://github.com/DashanGao/FakeAVCeleb>

Celeb-DF Dataset: <https://github.com/yuezunli/celeb-deepfakeforensics>

VoxCeleb Dataset: <https://www.robots.ox.ac.uk/~vgg/data/voxceleb/>

OpenSLR Dataset: <https://openslr.org/>

Tools and Technologies:

Programming Languages: Python

Frameworks and Libraries:

Deep Learning: PyTorch

Visual Analysis: Mediapipe

Audio Analysis: Librosa, Matplotlib

Multimodal Fusion: Scikit-learn

Deployment: Tkinter (desktop app)

Explainable AI: Grad-CAM

Free Resources:

Cloud Training: Google Colab Free Tier

Deployment Hosting: Streamlit Cloud