FINAL PROJECT PROPOSAL

Project Title: Deepfake Detection System

Track name: AI & Data Science - Microsoft Machine Learning Engineer

Instructor: Eman El-galad Round

Code: GIZ3_AIS2_S2

Submitted by:

- Mohab Hamdy Saleh Mustafa (Team Leader)
- Ahmed Sief Al-Aslam
- Mohamed Ragab Abo-Baker
- Youssef Ezzat Abd-Shafy
- Omar Mustafa Omar
- Osama Abd-Rahman Saad

Date: 10/2025

1. Project Description

The Deepfake Detection System project aims to develop an intelligent AI-powered tool capable of detecting manipulated videos and images generated using deep learning techniques. The system leverages PyTorch, a ResNet50 backbone combined with an LSTM network to analyze temporal and spatial features across video frames.

To accurately classify content as real or fake, the project incorporates frame sampling (N_FRAMES = 10 frames per video), face detection and cropping using Haar cascades, and image normalization, ensuring the model focuses on relevant facial features while reducing background noise.

The system is trained on the FaceForensics++ dataset, focusing on four types of deepfakes: Deepfakes, Face2Face, FaceSwap, and FaceShifter, with balanced data for robust performance. The framework is extensible, allowing future adaptation to additional manipulation types.

During inference, users can upload videos or images through a Streamlit-based web interface, and the model provides real-time predictions with confidence scores for each media file. The project also includes visualization of training and validation metrics, such as loss curves, AUC, accuracy, precision, recall, and F1-score, supporting evaluation and transparency.

Purpose & Practical Impact:

- Protect privacy by identifying manipulated personal media.
- Improve security by preventing the use of fake videos for malicious purposes.
- Fight misinformation by detecting falsified media in news and social platforms.
- Verify media authenticity to support informed decisions by individuals, journalists, and organizations.

The ultimate goal is to provide a reliable, user-friendly tool that helps verify digital content authenticity and mitigate the spread of misinformation.

2. Group Members & Roles

Names	IDs	Roles & Contributions
Mohab Hamdy Saleh Mustafa (Team Leader)	21065472	• Team coordination and task planning • Data preprocessing and feature extraction • Model design, training, and performance evaluation • Model deployment and integration supervision • Documentation and presentation management
Ahmed Sief Al-Aslam	21011557	• Data cleaning and augmentation • Model training and optimization using TensorFlow • Evaluation and comparison between different models • Contribution to project documentation
Mohamed Ragab AboBaker	21000751	• Dataset preparation and balancing • Support in model development and testing • Performance monitoring and result analysis • Report writing and data visualization
Youssef Ezzat AbdShafy	21009275	• Data preprocessing and integration with model pipeline • Support in result interpretation and visualization • Preparing project presentation materials • Documentation and teamwork coordination
Omar Mustafa Omar	21098126	• Research on deepfake detection techniques and related work • Assistance in data labeling and organization • Model deployment and integration supervision • Contributing to report and presentation design
Osama Abd-Rahman Saad	21011952	• Data preprocessing and augmentation • Model testing and evaluation support • Contribution to performance tracking and reporting • Documentation and result summarization

Collaboration Methodology: All team members work collaboratively in pairs, with each pair building upon the work of previous pairs. Every member contributes across all project phases, ensuring comprehensive involvement and knowledge sharing throughout the development lifecycle.

4. Objectives

a. Collect and Prepare Data:
Gather and preprocess videos and images from FaceForensics++, focusing on the four deepfake types: Deepfakes, Face2Face, FaceSwap, and FaceShifter. Perform frame sampling, face detection/cropping, and normalization to ensure high-quality input for the model.

b. Develop and Train Detection Models:
Design and train a deep learning model combining ResNet50 for spatial feature extraction and LSTM for temporal modeling using PyTorch. Optimize hyperparameters to accurately distinguish between real and fake media.

c. Evaluate and Optimize Performance:
Assess model performance using metrics such as accuracy, F1-score, precision, recall, and AUC, and refine preprocessing or model architecture to enhance reliability and reduce false positives/negatives.

d. Build a User-Friendly Interface:
Develop a Streamlit-based web application allowing users to upload videos or images and receive real-time deepfake analysis with confidence scores, along with visualization of model predictions.

e. Deploy and Test the Complete System:
Integrate the trained model into a cloud or local deployment environment, ensuring scalability, usability, and stability for practical use.

f. Promote Awareness and Ethical AI Use:
Highlight the importance of detecting deepfakes to support digital safety, privacy, authenticity, and trust in online media, and encourage responsible AI adoption.

5. Tools & Technologies

a. Programming Languages: Python.
b. Frameworks & Libraries: PyTorch, Torchvision, OpenCV, scikit-learn, Matplotlib, Pillow, NumPy, tqdm.
c. Dataset: FaceForensics++ focusing on four deepfake types: Deepfakes, Face2Face, FaceSwap, and FaceShifter.
d. Environment: Kaggle / Local machine (Windows) with GPU support for training and evaluation.
e. Deployment: Streamlit.

6. Stakeholder Analysis

The Deepfake Detection project involves multiple stakeholders who influence the design, implementation, and impact of the system. The following table outlines each stakeholder, their interests, influence level, needs, and engagement approach.

Stakeholder	Role / Interest	Influence / Priority	Needs / Expectations	Engagement /Communication
End Users (Journalists, Investigators, General Public)	Use the tool to verify the authenticity of videos or images before sharing or publishing.	High	Accurate detection (Real/Fake), confidence score, simple and fast user interface, data privacy.	User-friendly interface, educational content, and post-use feedback surveys.
Media Agencies / Social Media Platforms / Cybersecurity Firms	Integrate the detection tool into their systems to prevent misinformation and verify content authenticity.	High	High accuracy, detailed detection reports, accessible API, strong data protection and privacy policies.	Demonstrations, API documentation, progress reports, and collaboration meetings.
Development Team / Researchers	Responsible for building, training, and maintaining the deepfake detection model and web system.	Medium–High	Clean and balanced dataset, effective development environment, performance monitoring, and version control.	Weekly sync meetings, GitHub issue tracking, and internal technical documentation.

Instructor / Evaluator	Provides academic supervision, evaluation, and feedback on project progress and deliverables.	High	Clear objectives, consistent progress updates, and final deliverables including trained model, web demo, and report.	Progress reports, milestone reviews, and presentation meetings.

7. Milestones & Deadlines

Milestones	Description	Deadlines
M1: Project Planning & Research	Conduct a literature review on deepfake technologies, existing detection methods, and relevant datasets. Finalize the project scope and requirements.	29 Sep.2025
M2: Dataset Collection & Preprocessing	Gather real and fake media samples from DFDC and FaceForensics++ datasets. Perform cleaning, labeling, and augmentation for balanced data.	6 Oct.2025
M3: Model Design & Training	Develop and train deep learning models (CNN, XceptionNet, or ViT) for deepfake classification. Experiment with hyperparameters for best performance.	25 Oct.2025

M4: Evaluation & Optimization	Test the model using validation data, measure accuracy, F1-score, and latency, and optimize for improved detection performance.	1 Nov.2025
M5: Web Application Development	Build and integrate a user-friendly web interface (React + Flask) for uploading and analyzing videos or images.	7 Nov.2025
M6: Deployment & Final Testing	Deploy the model to a cloud platform (e.g., Render or AWS), perform user testing, and prepare final documentation and presentation.	10 Nov.2025

8. KPIs (Key Performance Indicators)

A. Data Quality

Metric	Description	Expected Result
Missing Values Handled	Percentage of missing or incomplete data successfully identified and handled during preprocessing.	98%
Data Accuracy After Preprocessing	Degree of correctness and consistency of data after cleaning, normalization, and augmentation.	95%
Dataset Diversity	Measures the variation in samples, ensuring the dataset includes different genders, ethnicities, lighting, and backgrounds.	90%

B. Model Performance

Metric	Description	Expected Result
Model Accuracy	Proportion of correct classifications among total predictions (real vs. fake).	87.76%
F1-Score	Harmonic mean of precision and recall, representing model balance and reliability.	88.68%
Latency per Image	Average time required to process and classify one image or video frame.	2000ms

AUC	Measures the model's ability to distinguish between real and fake samples by evaluating the area under the ROC curve. A higher AUC indicates stronger discrimination capability and more reliable classification across different threshold values.	97.67%

C. Deployment & Scalability

Metric	Description	Target / Result
API Uptime	Measures the system's availability and reliability during continuous operation.	≥ 99% (Goal)
Response Time	Time taken for the API to return detection results after receiving input.	≤ 2000ms (Goal) 1000:3000ms (Result)

D. Practical Impact

Metric	Description	Target / Result
Reduction in Manual Verification	Percentage decrease in time or effort needed for manual fake-content verification compared to baseline.	%
User Satisfaction	Percentage of positive feedback received from users during testing and evaluation phases.	%

9. UI/UX Design (Using Streamlit)

1. Objective:

The goal of the UI/UX design is to provide a clean, intuitive, and responsive interface that allows users to interact with the deepfake detection system efficiently. Streamlit was chosen because it enables rapid development of interactive web applications with real-time updates, easy layout control, and strong visualization capabilities.

2. Interface Structure:

The Streamlit application is organized into four main pages to ensure clarity, simplicity, and smooth navigation:

1. Dashboard (Home Page – app.py)

This is the main landing page of the system. It provides:

- A brief introduction to the project
- Overview of system features
- Key performance metrics (accuracy, F1-score, AUC)
- Navigation cards or buttons leading to other pages
- A clean dashboard-style layout to welcome users

2. Detection Page

This is the core functionality of the system where the deepfake detection occurs.
Features include:

- Video upload interface
- Preprocessing steps (frame extraction, face detection)
- Real-time model inference
- Display of results with:
    - Final classification (Real / Fake)
    - Confidence score
    - Inference time
- Display sample extracted frames for transparency

3. Visualization Page

This page is dedicated to model analysis and performance insights. It includes:

- Accuracy and loss curves
- Confusion matrix
- ROC curve and AUC value
- Bar charts comparing different deepfake types (Deepfakes, Face2Face, FaceSwap, FaceShifter)
- Visual summaries that help users understand how the model behaves

4. About Page

Provides background information about:

- The project and its purpose
- The team members and their roles
- Motivation behind deepfake detection
- Technologies used (PyTorch, ResNet + LSTM, OpenCV, Streamlit)
- Contact information

3. User Experience (UX) Considerations

To ensure smooth and efficient user experience, the following design principles are applied:

- Simple, minimal, and consistent layout for clear navigation
- Clean use of colors, spacing, and typography for readability
- Interactive components such as upload buttons and result panels
- Real-time result display without page reload
- Clear validation messages (invalid file, processing error, etc.)
- Lightweight and responsive design to ensure fast interaction