Project Abstract
DeepFake Detection

University: Egypt University of informatics

Major: Computer Science

Number of Team Members: 3

Country of residence: Egypt

Team Leader Name: Ahmed Abdelaziz

Team Leader Email: 21-101068@students.eui.edu.eg

Team Leader Gender: Male

Team Member 2 Name: Samy Mohamed Fattouh

Team Member 2 Email: 21-101076@students.eui.edu.eg

Team Member 2 Gender: Male

Team Member 3 Name: Adel Khaled

Team Member 3 Email: 21-101047@students.eui.edu.eg

Team Member 3 Gender: Male

Faculty Advisor Name: Prof. Fatty Mostafa Ahmed Salem

Faculty Advisor Email: fatty.salem@eui.edu.eg

Faculty Advisor University Job: Professor

جامعة مصر للمعلوماتية
EGYPT UNIVERSITY
OF INFORMATICS

Fatty

Which sector does your project tackle?

Sustainability

Which technology area are you using to tackle the addressed problem?

AI

1- Main problem we are solving

DeepFake technology, which uses artificial intelligence (AI) to manipulate video and audio to make them appear real, has received increasing attention due to its potential harm. DeepFakes use artificial intelligence to create realistic but fake media, which poses significant challenges, including: **Misinformation and Deception, Threats to Security and Privacy, Undermining Trust.**

Detecting these deepfakes is a complex challenge as the operations become more sophisticated, making it more difficult to distinguish between real and fake content.

Why we choose this problem

- **Rising Threat of DeepFakes:** The increasing sophistication and accessibility of DeepFake technologies make them a growing concern for society.
- **Social Impact:** Addressing this problem helps protect individuals and institutions from harmful consequences.
- DeepFake detection involves machine learning and deep learning techniques, which are valuable skills for our career.

2- Current Solutions

Current solutions for DeepFake detection can be categorized into several approaches:

1- **Traditional Machine Learning Techniques**

- **Feature-Based Methods:** These focus on identifying specific features in fake media

2- **Deep Learning Models**

- **Convolutional Neural Networks (CNNs):** Widely used for image-based DeepFake detection.
- **Recurrent Neural Networks (RNNs):** Applied for audio-based detection, analyzing temporal dependencies in speech patterns.

3- Frequency Domain Analysis

Focuses on identifying anomalies in the frequency spectrum of images or audio files. Fake content often introduces unique patterns in the frequency domain that are not present in real data.

4- Adversarial Training

- Using Generative Adversarial Networks (GANs) to simulate DeepFake creation during training helps the detection model learn to identify a broader range of fakes.

3- How will our proposed solution be used and operated in a Real-Life environment?

Our DeepFake Detection solution is designed to address the growing threats posed by DeepFake technologies in real-life scenarios. The system operates on two media types—**audio and images**—and can be deployed in multiple domains:

- Social media and Online Platforms:
  Integrated as a backend service for platforms like Facebook or YouTube. It scans uploaded images or audio files to detect potential DeepFakes in real time.
- Media verification for journalists:
  Media organizations can verify the authenticity of news content before publication.
- Public use via web or mobile application:
  individuals can upload content to verify its authenticity.

Real life workflow

- **Data Input:** Accepts static images, audio files, or video frames.
- **AI Processing:** Uses trained models (CNN for images, RNN for audio) to detect signs of manipulation.
- **Output and Feedback:** Produces results in real-time or near-real-time.

4- The high-Level architecture and the main software and hardware components of our system.

Fathy

1- High-Level Architecture

The system is divided into four main layers: Input Layer, Preprocessing Layer, Detection Layer, and Output Layer.

Input Layer

Function: Accepts input media (audio files, images, or video frames).
Components:

- **User Interface:** Web app, mobile app, or API for media upload.
- **Input Storage:** Temporary storage for uploaded files.

Preprocessing Layer

Function: Normalizes and prepares input data for analysis.
Components:

- **Image Preprocessing:** Resizing, denoising, and frame extraction (for video).
- **Audio Preprocessing:** Noise removal, spectrogram generation, and feature extraction.

Detection Layer

Function: Performs DeepFake detection using trained machine learning and deep learning models.
Components:

- **Image Detection Model:** Convolutional Neural Networks (CNNs) for identifying visual artifacts.
- **Audio Detection Model:** Recurrent Neural Networks (RNNs) or transformers to analyze temporal and spectral anomalies in audio.
- **Hybrid Models:** Combines audio and image insights for cross-modal validation (if applicable).

Output Layer

Function: Displays results and generates reports.
Components:

- **Visualization Module:** Highlights manipulated regions in images or anomalous segments in audio.

- **Report Generator:** Summarizes detection confidence and findings.
- **Feedback Interface:** Allows user interaction with results.

2- Main Software Components

1- **Programming Languages:** Python (for AI models), JavaScript (for front-end interfaces).

2- **Frameworks and Libraries:**

- TensorFlow/PyTorch: Model training and inference.
- OpenCV: Image preprocessing.
- Librosa: Audio preprocessing.
- Flask/Django: Backend development.

3- **Database:**

- SQL/NoSQL database to store logs and detection results.

4- **Cloud Services:**

- AWS/GCP/Azure for scalable deployment, storage, and real-time inference.

3- Main Hardware Components

- o Processing Units:
  GPUs (RTX 3060) for training and inference of deep learning models.
- o Storage:
  High-capacity SSDs for storing datasets, models, and temporary files.
- o Servers:
  Cloud-based instances for hosting the system.
- o Input Devices:
  File upload via desktop/mobile devices.

5- The technology platforms that we intend to use in building our system

- Operating System: we will use Windows

- Programming Languages:

Python: For implementing machine learning and deep learning models.
JavaScript/TypeScript: For developing the frontend and backend APIs.
SQL/NoSQL Queries: For database operations.

- Backend and frontend stacks:
  Backend: Flask or FastAPI for creating APIs
  Frontend: javascript for building the web interface and Flutter for cross-platform mobile applications.
  Database: SQL/NoSQL database

- AI Models and tools:
  TensorFlow/Keras
  OpenCV
  Librosa

- Deployment and cloud platforms:
  cloud Services:AWS/GCP/Azure

6- The new ideas that we intend to develop and include in our solution

- Our solution focuses on both audio and images, detecting manipulations across modalities. Unlike many existing models that focus on one type of media
- While many solutions operate offline and require significant processing time, we aim to incorporate real-time processing for live audio streams
- Our solution will include explainability features, such as visual heatmaps for image analysis and spectrogram anomalies for audio analysis.
- We plan to enhance our system's robustness against adversarial attacks and newer, more convincing DeepFake methods by training on up-to-date datasets

7- Our team system development methodology and quality assurance process

We will adopt an **Agile Development Methodology** for building our DeepFake Detection system. This approach allows flexibility, iterative improvements, and active collaboration among team members.

fathy

Quality Assurance Process

To ensure the reliability and accuracy of the system, we will follow a comprehensive quality assurance process:

- Validate the datasets for accuracy and Remove noisy or biased data that might affect model performance.
- Model Evaluation and Optimization
- Testing:
  Unit Testing
  Integration Testing
  Stress Testing
  Real-World Testing
- Implement logging mechanisms to track errors
- User Feedback
- Security Testing

8- Initial project management plans

1. **Distribution of Responsibilities and Tasks**
   The tasks will be distributed among the team members based on their expertise and interests. Each team member will have a key responsibility area, but there will be regular collaboration across all tasks to ensure consistency and integration.

   Team Member 1: AI Model Development and Training
   - Responsible for selecting and implementing AI models
   - Data collection, preprocessing, and model optimization.
   - Testing and evaluation of model performance.

   Team Member 2: Backend and Integration
   - Responsible for backend development
   - Integrating AI models with the backend services and ensuring smooth interaction between components.

   Team Member 3: Frontend Development and User Interface

- Developing the user interface (UI) for web using HTML, CSS and javascript
- Making sure users can upload media and view results.
- Building an interactive dashboard for showing detection results

2. Milestone Schedule

The project will be completed in 4 months, with defined milestones at each stage

Month 1:
Milestone 1: Complete system design and architecture.
Milestone 2: Data collection and preprocessing.
Milestone 3: Setup development environments and initial codebase.

Month 2:
Milestone 4: Train AI models and test on sample datasets.
Milestone 5: Complete basic backend API integration.
Milestone 6: Develop a basic UI prototype.

Month 3:
Milestone 7: Fine-tune models and optimize performance.
Milestone 8: Full integration between AI models, backend, and frontend.
Milestone 9: Conduct initial user testing.

Month 4:
Milestone 10: Final testing, deployment preparation, and bug fixes.
Milestone 11: Launch prototype for real-world testing and feedback.
Milestone 12: Prepare final documentation and project report.

3. Training Plan

Each team member will receive training in specific areas to ensure the necessary skills are developed.

Team Member 1 (AI Specialist):
Training Areas: Advanced TensorFlow/PyTorch, deep learning model optimization, audio and image preprocessing.

Team Member 2 (Backend Developer):
Training Areas: API development, backend frameworks (Flask/FastAPI), database management, Docker, cloud services (AWS).

Team Member 3 (Frontend Developer):
Training Areas: HTML, CSS, javascript, Flutter for mobile apps, UI/UX best practices.

4. **Risk Management and Contingency Plans**

   We have identified potential risks and have established strategies to less the risks as much as possible:

   Risk 1: Data quality or availability issues
   Solution: utilize publicly available datasets and work on data augmentation.

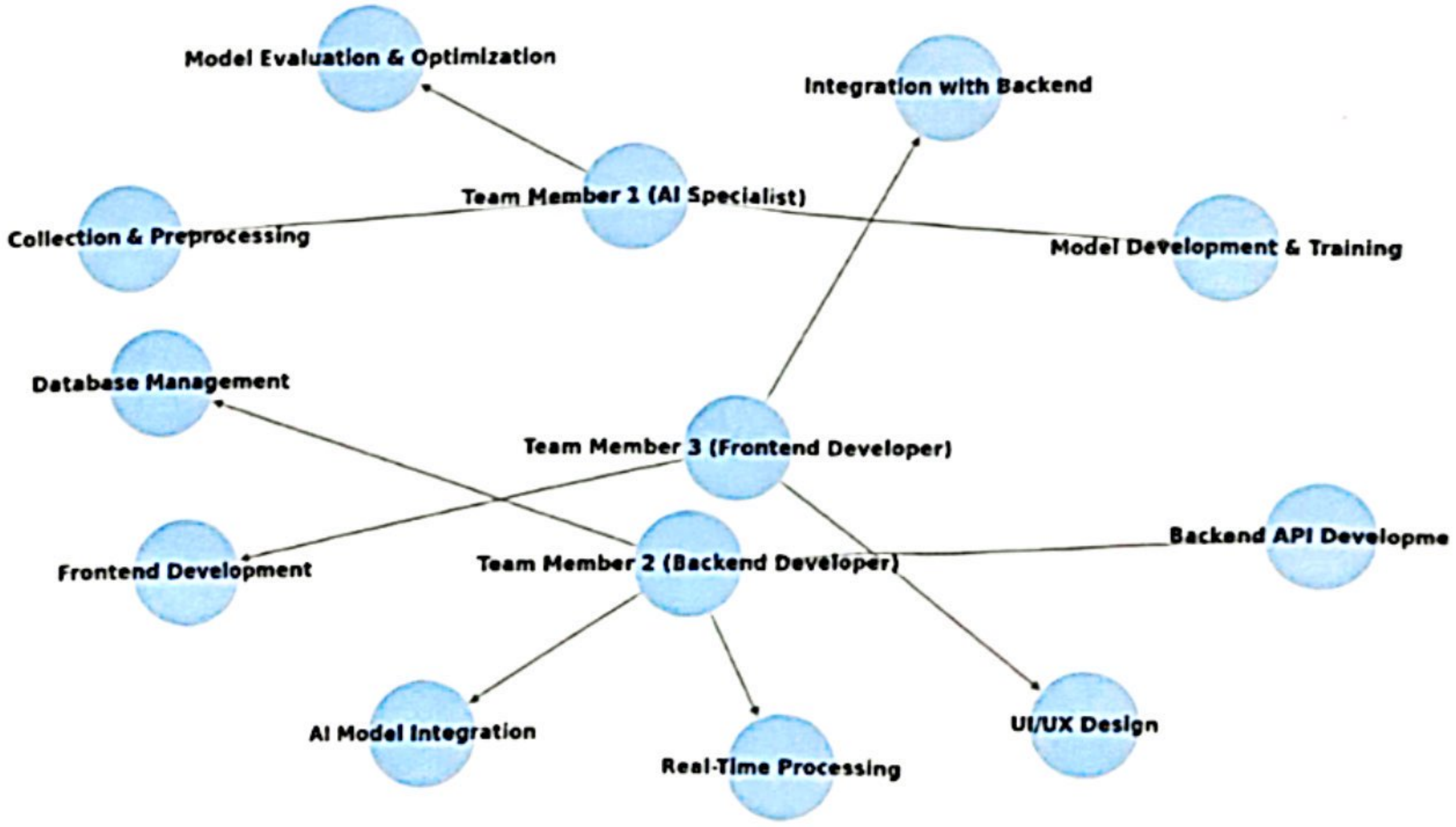
   Risk 2: Model performance not meeting accuracy expectations.
   Solution: continuously fine-tune the models and experiment with different architectures.

   Risk 3: Integration challenges between backend and AI models.
   Solution: regular testing of individual components before full integration.

5. **Change Control Procedures**

   To manage changes efficiently, we will:
   - Use Git/GitHub for version control
   - Weekly Team Meetings to review progress and discuss potential changes
   - Use Jira to manage tasks and any changes to deadlines and responsibilities

Task Distribution Flowchart


Model Evaluation & Optimization
Integration with Backend
Team Member 1 (AI Specialist)
Collection & Preprocessing
Model Development & Training
Database Management
Team Member 3 (Frontend Developer)
Backend API Developme
Frontend Development
Team Member 2 (Backend Developer)
AI Model Integration
Real-Time Processing
UI/UX Design

Here is the Task Distribution Flowchart, which shows the distribution of responsibilities and tasks among the 3 team members. Each team member is assigned key areas of the project such as AI model development, backend integration, and frontend development. This ensures that each member has a clear role while also maintaining collaboration across all parts of the system.