Project: PEGASUS FACEGUARD - Facial Recognition System

Overview

PEGASUS FACEGUARD is a comprehensive facial recognition system designed to enhance security by detecting and alerting for specific activities such as the presence of motorcycle helmets, weapons, and theft incidents. The project involves optimizing performance, enhancing the user interface, and packaging the system into an executable file.

See the Current EXE for Better Understanding.

https://we.tl/t-omnuS7Zv8k

Project Requirements

1. **Smooth Video Playback**
   - Ensure continuous video feed without pauses or lags.
2. **Executable File Creation**
   - Package the entire project into a standalone .exe file.
3. **User Interface Enhancement**
   - Develop an aesthetically pleasing and functional user interface.
4. **Licensing Implementation**
   - Integrate a system for license keys to manage software installation and usage.
5. **Notification System**
   - Implement a robust system to send WhatsApp notifications based on detections.

Functionalities to Implement

1. **User Interface**
   - Allow clients to change the phone number for WhatsApp notifications.
   - Enable adding and deleting photos of alleged criminals.
   - Enhance the graphical interface to be more user-friendly and visually appealing.
2. **Detection Features**
   - **Helmet Detection**: Identify individuals entering with a motorcycle helmet.
   - **Weapon Detection**: Detect individuals carrying weapons.
   - **Theft Detection**: Detect when items are taken from a shelf or display case.
   - **Body Language Analysis**: (Optional) Analyze nervous or restless behavior.
   - **Alarms**: Trigger alarms and send WhatsApp messages upon detection of helmet, weapon, or theft.
3. **Performance Optimization**
   - Improve detection accuracy and speed by optimizing the models and code.
   - Ensure the system can run smoothly on various hardware configurations.

Detailed Requirements

1. **WhatsApp Message Management**
    - Implement functionality to add or update WhatsApp messages that are sent upon detection events.
2. **Photo Management**
    - Enable the addition and removal of photos of alleged thieves through the user interface.
3. **Thieves Detection and Response**
    - Detect alleged thieves and sound an alarm.
    - Send a WhatsApp message and store the picture to Google Cloud upon detection.
4. **Helmet Detection**
    - Detect if someone is wearing a helmet and sound an alarm.
    - Send a WhatsApp message upon detection.
5. **Weapon Detection**
    - Detect if someone is carrying a weapon and sound an alarm.
    - Send a WhatsApp message upon detection.
6. **Theft Detection**
    - Sound an alarm if someone picks an item from the shelf.
    - Send a WhatsApp message upon detection.

Provided Resources

1. **Current Code Base**
    - The existing Python code for facial recognition.
2. **Model Files**
    - Trained models such as FaceNet, YOLOv3 for detection.
3. **Requirements File**
    - A `requirements.txt` file listing all necessary Python libraries.
4. **License Keys**
    - Example license key files for managing installations.
5. **Sample Dataset**
    - A set of images for testing the system's functionality.

Tasks for Developer

Initial Setup

1. **Code Review**
    - Review the existing codebase to understand the current implementation and identify optimization areas.
2. **Environment Setup**
    - Set up the development environment using the provided `requirements.txt` file.
    - Ensure all necessary libraries and dependencies are installed.

Core Development

1. **Smooth Video Playback**
    - Investigate and resolve the issues causing video frame pausing.
    - Optimize code to ensure continuous and smooth video feed.
2. **Executable File Creation**
    - Package the project into a standalone executable file using tools like PyInstaller.
    - Ensure the executable includes all necessary files and libraries.
3. **User Interface Enhancement**
    - Develop a more visually appealing and user-friendly interface.
    - Implement functionality to change WhatsApp numbers and manage photos of alleged criminals.
    - Add branding elements such as the PEGASUS FACEGUARD logo.
4. **Detection Features**
    - **Helmet Detection**: Implement and test the algorithm for detecting motorcycle helmets.
    - **Weapon Detection**: Implement and test the algorithm for detecting weapons.
    - **Theft Detection**: Implement and test the algorithm for detecting item theft.
    - **Alarm System**: Integrate an alarm system that triggers upon detection of helmets, weapons, or thefts.
5. **Notification System**
    - Ensure WhatsApp notifications are sent reliably upon detections.
    - Troubleshoot and resolve any issues with Twilio or alternative messaging services.
6. **Performance Optimization**
    - Refine detection models to improve accuracy and reduce lag.
    - Test the system on different hardware configurations to ensure performance.

Testing and Deployment

1. **Final Adjustments**
    - Make necessary adjustments based on testing feedback.
2. **Deployment**
    - Assist in deploying the final executable on the client's hardware.
    - Ensure the system runs smoothly and meets all specified requirements.