Proxy Patrol: Dual Voice and Face Detection System

Table of Contents

Introduction

Proxy Patrol is a dual voice and face detection system designed to monitor and detect instances of cheating during remote exams or interviews. It utilizes a combination of voice activity detection (VAD) and face detection algorithms to identify multiple voices and unnatural eye movements, reporting any suspicious behavior.

Installation

To run this project, you need to install the following dependencies:

Python 3.8+

Ensure Python is installed on your system. You can download it from the official Python website.

Pip

The Python package installer. It usually comes with Python, but you can install it using the instructions here.

Dependencies

Install the necessary Python packages using pip:

sh
Copy code

```
pip install pyaudio webrtcvad numpy opencv-python pillow
```

Additional Setup for pyaudio:

For Windows:

sh
Copy code
```
pip install pipwin
pipwin install pyaudio
```

For macOS:

sh
Copy code
```
brew install portaudio
pip install pyaudio
```

Code Overview

The project is composed of a Python script that integrates various libraries to achieve real-time audio and video processing.

Main Components

- **pyaudio:** Captures audio input from the microphone.
- **webrtcvad:** Analyzes audio data to detect voice activity.
- **numpy:** Handles numerical operations on audio data.
- **cv2 (OpenCV):** Captures and processes video frames for face and eye detection.
- **tkinter:** Creates a GUI for displaying video feed and detection messages.
- **PIL (Pillow):** Processes images for the GUI.

Code Structure

The script includes several key functions and classes:

- **start_detection:** Initializes audio and video streams and starts the detection thread.
- **stop_detection:** Stops audio and video streams.
- **detect_faces:** Detects faces in the video frames.
- **preprocess_frame:** Preprocesses video frames for better detection accuracy.
- **detect_dual_voices:** The main detection loop that processes audio and video data to detect multiple voices and face movements.

ML Algorithms Used

Voice Activity Detection (VAD)

The project uses the WebRTC VAD algorithm, which is a robust, real-time voice activity detection algorithm. It processes audio chunks to determine whether they contain speech.

Face Detection

The project uses Haar Cascade Classifiers from OpenCV for face and eye detection. These classifiers are pre-trained models that can detect faces and eyes in an image or video frame.

Execution

To run the project, save the script in a Python file, e.g., `proxy_patrol.py`, and execute it using Python:

sh
Copy code
```
python proxy_patrol.py
```

GUI Interaction

- **Start Detection:** Begins monitoring audio and video streams.
- **Stop Detection:** Stops the monitoring process.

Detailed Explanation

Parameters

- **FORMAT:** Audio format (16-bit PCM).
- **CHANNELS:** Number of audio channels (mono).
- **RATE:** Sampling rate (16 kHz).
- **CHUNK_DURATION_MS:** Duration of each audio chunk in milliseconds (30 ms).
- **CHUNK_SIZE:** Number of audio samples per chunk.
- **WINDOW_DURATION:** Duration of the sliding window for voice detection (5000 ms).
- **WINDOW_SIZE:** Number of chunks in the sliding window.
- **VIDEO_WIDTH:** Width of the video frame (640 pixels).
- **VIDEO_HEIGHT:** Height of the video frame (480 pixels).

Initialization

- The VAD is set to the most aggressive mode (3) to maximize detection sensitivity.

Face Detection

- The Haar Cascade classifiers for face and eye detection are loaded from OpenCV's pre-trained models.

GUI Setup

- A Tkinter window is created with a background image. Video frames are displayed in this window along with detection messages.

Detection Logic

- **Audio Processing:** Audio chunks are captured from the microphone and analyzed by the VAD to detect speech.
- **Face and Eye Detection:** Video frames are captured from the camera and processed to detect faces and eyes. The detection results are used to determine if cheating is occurring.

Cheating Detection

- **Dual Voices:** The system checks if multiple voices are detected within a sliding window of audio data.
- **Unnatural Eye Movement:** The system monitors eye movements and positions relative to the face to detect unnatural behavior.

Build Tools

In the development of Proxy Patrol, specific build tools are necessary to ensure the proper setup and functionality of the various components, especially for dependencies like dlib. Below are the build tools and their installation instructions:

CMake

CMake is an open-source, cross-platform family of tools designed to build, test, and package software. It is required for building the dlib library from source.

Installation:

Windows:

1. Download the installer from the CMake website.
2. Run the installer and follow the installation instructions.
3. Ensure that you add CMake to your system PATH during installation.

macOS:

If you have Homebrew installed, you can install CMake using the following command:
sh

Copy code

```
brew install cmake
```

1.

Linux:

You can install CMake using your package manager. For example, on Ubuntu:
sh
Copy code

```
sudo apt-get update
sudo apt-get install cmake
```

1.

dlib

dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real-world problems. It has Python bindings that can be installed via pip, but it requires CMake for building.

Installation:

Install dependencies:

Windows:

- Install Visual Studio with C++ development tools.
- Ensure CMake is installed and added to your PATH.

macOS:

sh
Copy code

```
brew install cmake
brew install boost
```

Linux:

sh
Copy code

```
sudo apt-get update
sudo apt-get install build-essential cmake
sudo apt-get install libboost-all-dev
```

Install dlib using pip:

sh
Copy code
```
pip install dlib
```

If you encounter issues with the above command, you may need to build dlib from source:

sh
Copy code
```
git clone https://github.com/davisking/dlib.git
cd dlib
mkdir build
cd build
cmake ..
cmake --build .
cd ..
python setup.py install
```

Python Wheels

Wheels are the standard format for Python distribution and are preferred for installing packages because they contain compiled code that is ready to use.

Installation:

Ensure you have pip installed:
sh
Copy code
```
python -m ensurepip --upgrade
```

1.

Use pip to install Python packages:
sh
Copy code
```
pip install package_name
```

2.

To install a specific wheel file:
sh
Copy code
```
pip install path_to_wheel.whl
```

3.

Common Python Packages:

sh
Copy code
```
pip install numpy
pip install opencv-python
pip install Pillow
pip install webrtcvad
pip install pyaudio
pip install threading
```

Additional Notes

- Ensure that all dependencies are compatible with your Python version.
- Regularly update your packages to the latest versions to benefit from bug fixes and improvements.
- For complex dependencies, refer to the official documentation for installation instructions and troubleshooting.

By following these steps, you can set up the necessary build tools for developing Proxy Patrol, ensuring a smooth development process.

This documentation provides a comprehensive overview of the Proxy Patrol project, including installation instructions, code structure, and detailed explanations of the implemented algorithms and their functions. For any further assistance, please refer to the respective library documentation or seek support from the community.