

Environment Setup For Online Exam Cheating Detection

1. System Requirements

- Operating System: Windows 10/11, macOS, or Linux
- RAM: Minimum 8GB (Recommended 16GB for smooth execution)
- Python Version: Python 3.8 or later
- Additional Requirements: Webcam, Microphone

2. Install Python

1. Download Python 3.8 or later from [Python official website](#).
2. Install Python and ensure to check "Add Python to PATH" during installation.
3. Verify installation:

```
Python  
python --version
```

3. Create Virtual Environment (Optional but Recommended)

1. Open a terminal or command prompt.
2. Navigate to the project folder:

```
Python  
3. cd path/to/project
```

4. Create a virtual environment:

```
Python  
5. python -m venv exam_env
```

6. Activate the virtual environment:

- Windows:

```
Python  
○ exam_env\Scripts\activate
```

- macOS/Linux:

Python

- `source exam_env/bin/activate`

4. Install Required Dependencies

Run the following command to install all required Python libraries:

Python

```
pip install opencv-python numpy mediapipe pyaudio tensorflow psutil pywin32
```

Additional Installation Notes

- For Windows Users:
 - Install PyAudio dependencies:

Python

```
pip install pipwin
```

```
pipwin install pyaudio
```

- **Install TensorFlow with GPU (if applicable):**

Python

- `pip install tensorflow[and-cuda]`

- For macOS Users:
 - Install PyAudio using Homebrew:

Python

```
brew install portaudio
```

```
pip install pyaudio
```

- Install AppKit dependency:

Python

- `pip install pyobjc-framework-AppKit`

- For Linux Users:

- Install required system packages:

Python

- `sudo apt-get install python3-pyaudio portaudio19-dev`

- Then install dependencies:

Python

`pip install opencv-python numpy mediapipe pyaudio tensorflow`

5. Verify Installation

Run the following script to verify that all dependencies are correctly installed:

Python

```
import cv2
import numpy as np
import mediapipe as mp
import pyaudio
import tensorflow
import platform
import sys

print("All dependencies installed correctly.")
```

6. Running the Exam Monitoring System

1. Ensure your webcam and microphone are working.
2. Activate the virtual environment (if used):

Python

```
exam_env\Scripts\activate # Windows
```

```
source exam_env/bin/activate # macOS/Linux
```

3. Run the script:

Python

```
python exam_monitoring.py
```

7. Troubleshooting

- ModuleNotFoundError: Run `pip install <missing_module>`.
 - Camera Not Detected: Ensure no other application is using the webcam.
 - PyAudio Errors: Install system dependencies as per OS instructions above.
 - TensorFlow GPU Issues: Ensure compatible GPU drivers and CUDA are installed.
-

Creation :

This document outlines the *gadget requirements* for the *exam monitoring gadget* (EMS). The cause of this machine is to screen a web examination environment for suspicious sports along with cheating, distractions, or unauthorized movements. The system utilizes laptop vision, audio evaluation, and actual-time tracking techniques to make certain that the examination surroundings stays controlled.

Machine Evaluation :

The exam monitoring gadget (EMS) is designed to:

- detect suspicious sports, including:
 - Face orientation troubles
 - software switches.

- cellphone utilization during the examination.
- Loud noises.
- Environmental adjustments (e.g., movement within the history).
- trigger a countdown and cease the examination if suspicious activities are detected continuously.
- show real-time remarks, including a countdown timer and suspicious activity warnings.

System Requirements :

- **Functional Necessities :**

The EMS machine ought to meet the subsequent practical requirements:

1. Real-Time Video Processing:

- capture and method the video move from the webcam in actual-time.
- perform face tracking and detection to make certain the candidate's face is orientated efficiently.
- discover any changes within the candidate's surroundings (e.g., if the candidate moves or leaves the frame).

2. Audio monitoring :

- display the surroundings's audio ranges in real-time.
- hit upon loud noises or interruptions that could imply cheating behavior.

3. Smartphone Detection :

- discover the presence of mobile phones or comparable gadgets within the candidate's vicinity the use of object detection models (MobileNetV2).

4. Software switch Detection :

- screen the presently lively utility at the candidate's device.
- come across if the candidate switches faraway from the exam window.

5. Suspicious pastime managing :

- If suspicious behavior is detected (e.g., smartphone use, loud noise), log it.
- If suspicious conduct persists beyond a threshold, trigger a countdown to quit the examination.

6. Timer and exam tracking :

- show a countdown timer displaying the closing time of the examination.
- show alerts for detected suspicious sports (e.g., face turned, loud noise).

7. Consumer instructions display screen :

- show a tutorial display screen with examination guidelines, consequences, and begin instructions earlier than the examination starts offevolved.

● **Non-functional necessities**

1. Overall performance :

- The gadget ought to run without important lag, processing video and audio input in real time.
- frame price for video seize must be as a minimum *30 fps*.

2. Compatibility :

- The device must be well matched with the subsequent operating systems:
 - *macOS* (via AppKit for software detection).
 - *windows* (thru win32gui, psutil for application detection).

3. Safety :

- The device must now not add any personal or touchy information (video/audio) until required for logging functions.
- All information accrued must continue to be local until an specific, person-described movement requires facts upload.

4. Reliability :

- The device need to deal with errors gracefully and preserve functioning if viable.
- The machine have to clean up assets (e.g., microphone and webcam) when the utility closes or crashes.

5. Usability :

- The system need to provide clear commands to the consumer approximately the exam guidelines and any suspicious hobby detection.
- The UI ought to offer a clean-to-understand warning if suspicious hobby is detected.
- The examination instructions screen needs to be displayed earlier than the examination starts.

6. Scalability :

- The gadget should be able to operate for lengthy durations (up to one hour) without failure.
- The structure ought to allow for destiny scalability, enabling integration with a larger exam system or multi-candidate environments.

7. Maintainability :

- Code ought to be modular, allowing clean updates and renovation, specifically for detection models and threshold values.
- error logging has to be implemented to ensure quick diagnostics.

Machine components and hardware requirements :

The system is composed of the subsequent additives:

1. Webcam :

- excessive-definition (HD) webcam for taking pictures real-time video.
- minimum decision: *640x480*.
- recommended decision: *1080p (1920x1080)* for clearer face detection.

2. Microphone :

- A microphone to seize environmental sound.
- The machine must detect loud noises and strange audio tiers.

3. PC necessities :

- working gadget :
- macOS* (10.14 or later) or *windows 10* or better.
- CPU :
 - minimum: Intel center i5 or equivalent.
 - recommended: Intel center i7 or higher for smoother processing.
- RAM :
 - minimum: 4 GB
 - endorsed: 8 GB for higher performance with video and audio processing.
- GPU (for stronger performance):
 - optional but endorsed: *NVIDIA GPU* with CUDA assist for faster photograph processing.
- garage:
 - minimal: 10 GB of to be had storage for software and temporary documents.
- community:
 - whilst community connection isn't always strictly required for neighborhood functionality, net get right of entry to can be wished for initial setup, software updates, or cloud integration (if applicable).

4. Software requirements:

- Python 3.x (advocated: Python three.7 or better).
- Required libraries:
 - OpenCV: For video processing and show.
 - MediaPipe: For face detection.
 - TensorFlow: For the MobileNetV2 item detection version.
 - pyaudio: For audio flow managing.
 - psutil, win32gui, and AppKit (depending on OS) for utility transfer detection.
- Dependencies :
 - extra Python libraries for set up:

bash

```
pip install opencv-python mediapipe pyaudio tensorflow psutil  
pyobjc-framework-AppKit
```

Consumer Interface (UI) necessities 1 :

1. Exam instructions display :

- complete-display window showing the examination policies, tips, and results.
- display textual content need to be clean to study with proper contrast.
- provide instructions to start the examination and how to go out.

2. Examination monitoring screen :

- show real-time video feed from the webcam.
- show a countdown timer displaying the ultimate exam time.
- show indicators and detect suspicious sports (e.g., "cell phone Detected", "Loud Noise Detected").

3. Alert gadget :

- If suspicious activity is detected, show a big, sizable caution message at the screen (e.g., "Suspicious interest Detected!").
- If the examination is set to quit, show a countdown for the exam closure.

4. Exit Mechanism :

- allow the user to cease the examination by means of urgency by pressing the "Q" or "Tab" key.

Mistakes coping with and Logging :

The machine must deal with mistakes gracefully, specifically in the following cases:

- Digicam error : If the webcam is unavailable or cannot be accessed, show a clean mistakes message.
- Audio error : If the microphone can't be accessed, display a warning to the person.
- Model Failure : If the phone detection or face detection models fail, deal with the error through logging it and trying to get better.

Moreover, all mistakes and vital events have to be logged to a record for post-examination evaluation.

Trying out and Validation :

The subsequent checks have to be achieved to validate the gadget:

1. Unit checking out : every feature (e.g., face orientation check, audio monitoring) ought to be tested independently.
2. Integration trying out : ensure all additives paintings collectively, from video capture to hobby detection and alerting.
3. Give up-to-cess testing : Simulate the whole exam procedure, from starting the examination, detecting suspicious activity, to finishing the examination.
4. Overall performance testing : make sure the system performs in actual-time underneath normal situations and heavy load (e.g., lengthy examination periods, various environmental conditions).

Conclusion :

The Examination Monitoring Gadget is designed to preserve integrity at some point of online tests by tracking suspicious sports including face orientation, smartphone use, noise levels, and surroundings changes. This record outlines the required hardware, software, and device specifications vital for the a hit implementation of the system. The machine is designed to run easily, be user-friendly, and make sure the examination manner remains truthful and uninterrupted.