# Automated Monitoring System with Real-Time Visual Capture

Shubha Arya
*Data Science Instructor, Unicminds, Bangalore*

***Abstract*—This paper presents an automated noise detection system which has been designed for classroom settings. It monitors the sound level in an ambient environment, using a microphone and notifies the user when noise exceeds a certain threshold. On detecting excessive noise, the system records the time and noise level in dB in a CSV file and captures an image using the web camera attached to the system. I have created it using Python using libraries like sounddevice, numpy, tkinter, and OpenCV. This application provides a simple yet effective solution for maintaining discipline in the classroom environments and can be extended to other no noise zones like libraries and offices.**

***Index Terms*—OpenCV, Computer Vision, Class room Management, Real-time monitoring, Noise detection, tkinter GUI.**

## I. INTRODUCTION

Classroom management is of the utmost importance in schools. We all have studied in schools and so we all are aware of the significance of silence for creating an environment conducive to learning as it enhances focus and concentration, the inverse of which can definitely degrade overall academic performances. The objective of this project is to propose a technological solution to this problem by continuously monitoring the levels of noise in real time and then capturing the pictures of those who contribute to high decibel noises. This type of automated monitoring in classrooms is definitely a big support to teachers and administrators in promoting discipline, henceforth improving the results of the students. Not only this, I believe that if this system is installed at every place like cafes and restaurants also, and adjusting the volume decibels to be captured according to the environment settings we can even detect the chaotic elements in public settings.

## II. SYSTEM ARCHITECTURE

A. Overview

The system comprises four primary components:

1. Microphone Input: Captures ambient noise in real-time.
2. Volume Analyzer: Converts input audio into decibel levels.
3. Image Capture Unit: Uses a webcam to capture images on noise threshold breach.
4. Graphical User Interface (GUI): Displays real-time volume levels and alerts.

B. Libraries Used

- sounddevice: For accessing real-time audio input.
- numpy: For numerical operations on sound samples.
- tkinter: For GUI creation.
- OpenCV: For webcam access and image capture.
- csv and time: For logging.

## III. WORKING PRINCIPLE

A. Volume Detection

The system uses the sounddevice library to stream audio input. The norm of the incoming sound wave is computed and scaled to estimate volume in decibels:

```
def get_volume(indata):
  volume_norm = np.linalg.norm(indata) * 10
  return int(volume_norm)
```

B. Noise Logging

If the noise level exceeds the set threshold (default: 50 dB), the system logs the volume and current time to a CSV file:

```
def log_noise(volume):
  writer.writerow([time.strftime("%H:%M:%S"),
volume])
```

C. Image Capture

When a noise event is detected, the webcam captures an image and saves it with a timestamp:

```
filename = f"face_{timestamp}.jpg"
```

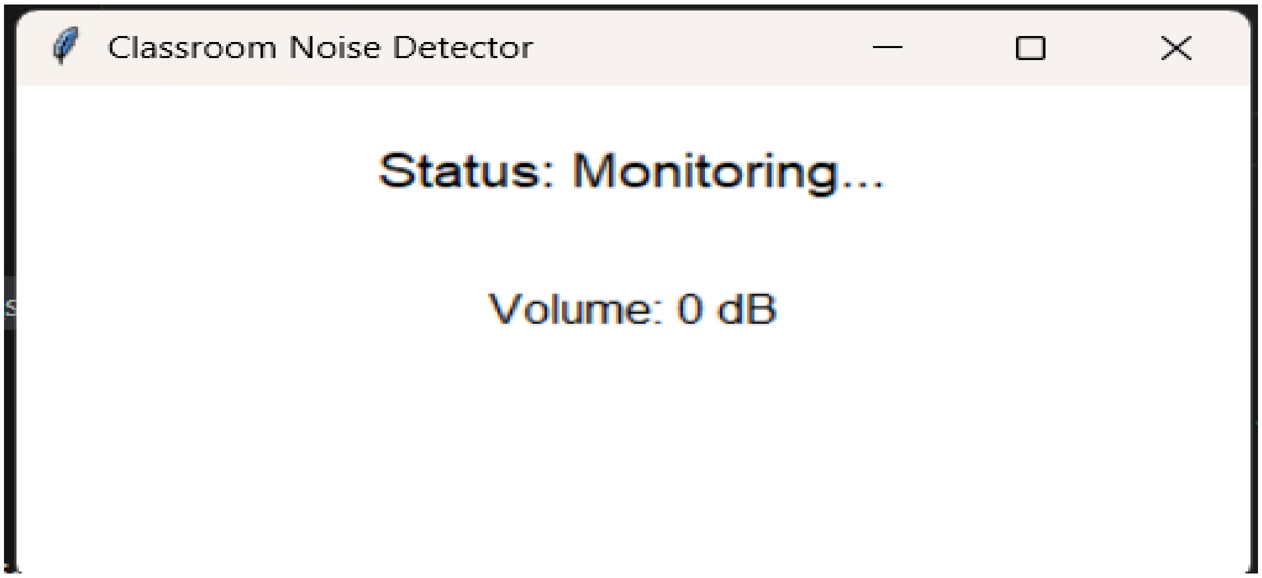
cv2.imwrite(filename, frame)

D. User Interface

The GUI displays current volume levels and alerts the user when a noise event is detected. It resets after a brief cooldown.

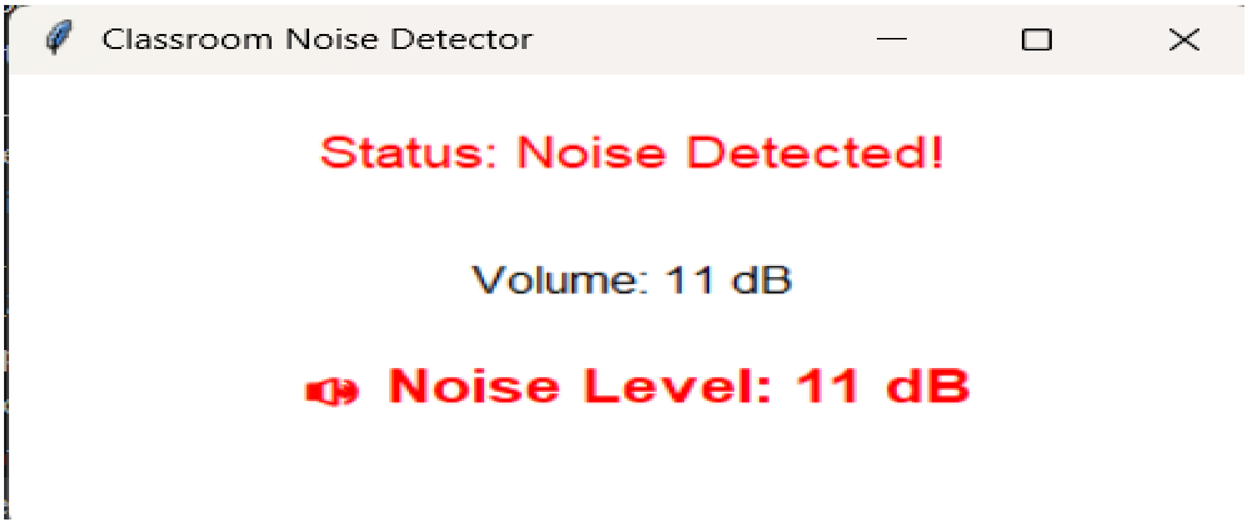
## IV. RESULTS

The application was tested in a simulated classroom environment. It successfully:
- Detected noises above 50 dB.
- Captured and saved images.
- Logged accurate timestamps and volume levels.
- Maintained a responsive GUI interface without lag.



Status: Monitoring...

Volume: 0 dB



Status: Noise Detected!

Volume: 11 dB

Noise Level: 11 dB

Sample CSV log:

```
Time      | Volume (dB)
------------------------
10:32:45  | 62
10:34:21  | 73
```

Captured image filenames followed the format face_HHMMSS.jpg.

## V. APPLICATIONS

- Classrooms: For monitoring discipline.
- Libraries: For identifying noise disturbances.
- Hospitals/Offices: For maintaining silence zones.

## VI. CONCLUSION

The project gives an idea of a lightweight and effective solution to monitor and control classroom noise using Python. Future improvements could include machine learning for face recognition and cloud integration for remote monitoring.

## REFERENCES

[1] Python Documentation- https://docs.python.org/3/
[2] OpenCV Library- https://opencv.org/
[3] NumPy Documentation- https://numpy.org/
[4] Tkinter GUI Programming - https://docs.python.org/3/library/tkinter.html
[5] Sounddevice Library- https://python-sounddevice.readthedocs.io/