Piezoelectric-Powered Air Quality Monitoring and Accident Detection System

Pallab Roy[1]; Subhajit Mondal[2]; Subhajit Pal[3]; Suman Biswas[4]; Atanu Gayen[5]; Amit Mondal[6]; Sumit Bera[7]; Rahul Dutta[8]; Partha Ray[9]

[1]Department of Electrical Engineering JIS College of Engineering Kalyani, India
[2]Department of Electrical Engineering JIS College of Engineering Kalyani, India
[3]Department of Electrical Engineering JIS College of Engineering Kalyani, India
[4]Department of Electrical Engineering JIS College of Engineering Kalyani, India
[5]Department of Electrical Engineering JIS College of Engineering Kalyani, India
[6]Department of Electrical Engineering JIS College of Engineering Kalyani, India
[7]Department of Electrical Engineering JIS College of Engineering Kalyani, India
[8]Department of Electrical Engineering JIS College of Engineering Kalyani, India
[9]Department of Electrical Engineering JIS College of Engineering Kalyani, India

Publication Date: 2025/05/29

Abstract: This work develops a smart energy-efficient air quality monitoring system operated by piezoelectric technology harvesting energy from foot pressure. The system is monitoring environmental toxic gasses such as CO, CO_2, which provides real -time data to assess environmental conditions. In parallel, this work also detects car accident using YOLOv11. This double function system not only promotes environmental awareness and public health but also improves traffic safety by enabling rapid emergency reactions. The self -directed design ensures little maintenance and high reliability, making it a viable solution for smart city infrastructure and sustainable urban development.

Keywords: *Piezoelectric, Foot Step Power Generation, Air Quality Monitoring, Accident Detection System, DHT11 & MQ135.*

How to Cite: Pallab Roy; Subhajit Mondal; Subhajit Pal; Suman Biswas; Atanu Gayen; Amit Mondal; Sumit Bera; Rahul Dutta; Partha Ray (2025). Piezoelectric-Powered Air Quality Monitoring and Accident Detection System. *International Journal of Innovative Science and Research Technology,* 10(5), 2258-2267. https://doi.org/10.38124/ijisrt/25may1425

I. INTRODUCTION

Walking is the most common activity in day-to-day life. Humans have been walking for millions of years and antedating form of transportation. Using this simple process, creating electricity through piezo-electric sensors is possible. In walking, human loses his energy to the road surface in the form of pressure, through foot pressure on the ground during every step. In this context, the occurrence of piezoelectricity, certain materials to generate electrical energy under foot pressures presents a promising opportunity to develop permanent and independent surveillance systems. By using the pressure, piezoelectric convert it into usable electrical power, and possibly eliminate the need for a battery or external power supply. In this work the harvested piezoelectric energy is stored in DC battery and is used for air quality monitoring as well as in webcam for detects any kind of serious conditions on road and sends the data to specific emergency contacts. This paper checks viability and the development of a new Piezoelectric-driven air quality monitoring using the ESP8266 Microcontroller and accident detection system through webcam using YOLOv11.

Air quality monitoring has become important due to increasing level of pollution, especially in the urban environment. Systems usually use sensors such as MQ-135 and DHT11 to detect pollutants such as CO, CO_2, and NH_3 and monitor temperature and humidity. The air quality monitoring systems proposed in [1] cannot provide satisfactory solutions with low costs in real time. In [2], the authors have given an insight of a system which will have ARM7, LPC2138 which is interfaced with the microcontroller for the air pollution monitoring system. The obtained values are given to smart phone using Bluetooth. The use of Bluetooth slows down the data transmission compared to Wi-Fi transmission. In [3] authors focused on collision avoidance by informing users of the distance between their vehicle and objects. This system mis alerts the user during traffic jams, when the distance is too low between the vehicles. The air quality monitoring systems proposed in [4] they are not energy-efficient and sustainable. The conventional techniques do not make the system efficient for remote monitoring and cause inadequacy in detecting the rise in pollution concentration. In [5] the authors noted that the significant problems faced by the traditional air monitoring system are the relatively modern hardware innovation, unsafe operation, high cost and bulky instruments. In [6] the author has used YOLOv8 for accident de3tection and tracking. But [7] introduces YOLOv11 as the breakthrough in real time object detection delivering superior performance in various computer vision tasks, including object detection, feature extraction, instance segmentation, pose estimation, tracking, and classification.

The proposed system is mainly to overcome the problems, as well as provide a provision to visualize the data from the air pollution monitoring system in a graphical and tabular manner. The system is designed to detect various harmful gases present in air apart from detecting particle pollutants. It also keeps track of the temperature and humidity of the surrounding. The system also provides an alarm so that the user is made aware of the fact that the surrounding air is not safe for living and the user should take necessary actions to help reduce air pollution. This makes it a simple system which effectively provides valuable inputs to the user regarding the air quality.

II. METHODOLOGY

The figure 1 presents a comprehensive flow chart for the implementation of a piezoelectric-powered air quality monitoring and accident detection system. The working flowchart is divided into several interconnected blocks that perform specific tasks.

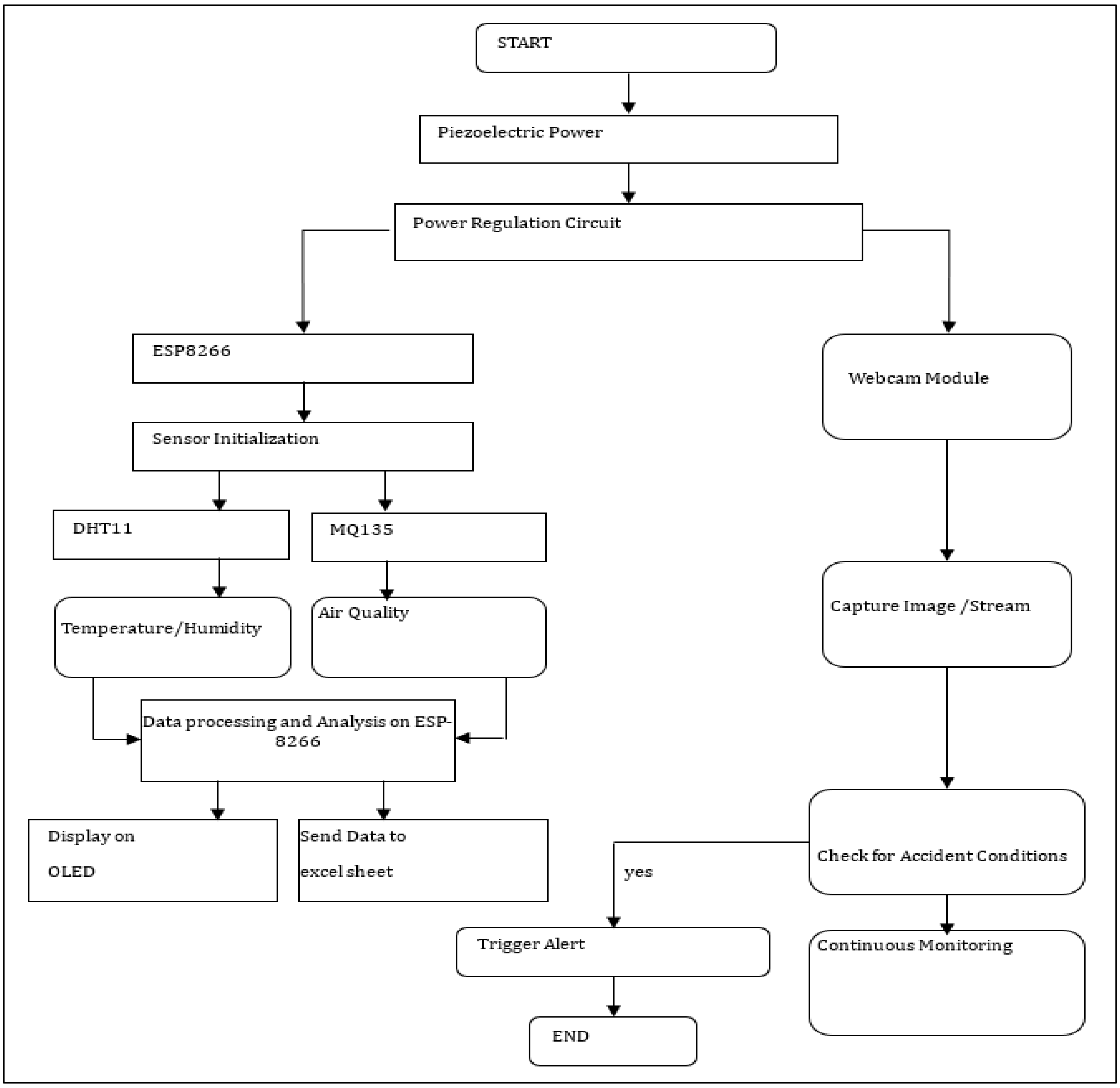


Fig 1 Working Flowchart

Firstly the system is initiated with piezoelectric power, which is collected from footstep pressure. This power is fed into a power regulation circuit to ensure stable voltage levels for operating the electronic components, ESP8266 microcontroller and webcam module. A digital humidity and temperature sensor (DHT11) and an air quality gas sensor module (MQ135) monitor environmental humidity and temperature and detect the harmful gases respectively. The temperature, humidity, and air quality data are gathered and processed by the ESP8266 and send the data to Organic Light-Emitting Diode (OLED) Screen and saved in excel sheet. OLED screen is used to display the real time local visualization of the air quality parameter such as harmful gases, temperature, and humidity and all the records of ttemperature, time, humidity and air quality index (ppm) are saved in excel sheet for further analysis. Simultaneously for accident detection case power regulator transfer the stable voltage to the web camera module. A web camera module is used to capture live video streams of the monitored area. These visual inputs are processed to detect accident conditions using trained models YOLOv11. The YOLOv11 can achieve a high level of accuracy, which is especially important for this project as multiple objects such as vehicles and pedestrians need to be identified in a scene. Moreover, YOLOv11 performs single pass detection through the neutral network, resulting in faster interference. If an accident is not detected, the system returns to continue monitoring both environmental conditions and camera feeds. If an accident is detected, the system immediately triggers an alert.

A. *Software Implementation for Air Quality Monitoring System.*

➢ *Setting up the Arduino IDE Code (Shown in APPENDIX-I):*

- *STEP 1-Initializing OLED.*

Initializing serial communication at 9600 baud, allowing the Arduino to communicate with computer via the Serial Monitor. This code is used for the Arduino IDE to initialize an OLED display and a DHT sensor (likely a DHT11 for temperature and humidity).

- *STEP 2- Reading data from MQ135 sensor and DHT11 Sensor.*

After every 32 seconds, the Arduino does the following:

✓ Reads gas sensor (MQ135) value.
✓ Reads temperature and humidity (DHT11).
✓ Sends the data using a function called sendData().

- *STEP 3-Display on OLED*

This code shown reads data from MQ135 and DHT11 sensor renders everything onto the OLED screen. Until this is called, nothing appears.

- *STEP 4-Connect to Wi-Fi*

On running the code CONNECTING TO WIFI on the ESP8266, ESP8266 attempts to connect to a Wi-Fi network using the given Service Set Identifier (SSID) and password and show the connection progress to the serial monitor. Once connected, it confirms success and shows the device's IP address on the network.

- *STEP 5-Sends data to PC*

This function reads three readings - temperature, humidity, PPM, sets up a Wi-Fi client to connect to a server via HTTPS to establish a (non-secure) HTTPS connection with the given host and port. If the connection fails, it logs the failure and exits.

- *STEP 6- Sends the collected data to the script*

This code converts sensor values to strings. It uses the given URL to send those values to the Google Apps Script endpoint by sending a GET request to that URL over HTTPS. Then HTTPS reads and checks the response and finally, it prints debug info to the serial monitor.

➢ *Script for Excel Input (as Shown in APPENDIX-II):*

In this Google Apps Script, function named do Get(e) is acting as a web app backend to receive data (from ESP8266 using HTTP GET) and log it into a Google Spreadsheet whenever the web app receives a GET request. 'e' is the event parameter that contains the query parameters and sends them to the excel sheet URL. It returns response to client (ESP8266) by sending a plain text response ("Temperature Written on column C, Humidity Written on column D, Air PPM Written on column E") back.

B. *Software Implementation for Accident Detection:*

The custom dataset is labelled using a graphic image annotation tool LabelImg for labeling object bounding boxes in an image during tasks such as object detection, segmentation, and classification. The classification of the training data is done with the highly efficient YOLOv11 algorithm. YOLOv11 can achieve a high level of accuracy, which is especially important for this project as multiple objects such as vehicles and pedestrians need to be identified in a scene. Moreover, YOLOv11 performs single pass detection through the neutral network, resulting in faster interference. Each mobile object i.e. cars, bikes, trucks, pedestrians, etc., present in the image is individually annotated with a class name. The frame with an accident event is given the "accident" class name. The process is then continued till all frames in the dataset have been labelled. The model is finally implemented using rob flow, which uses GPU, making it much faster than local environments.

C. *Hardware Implementation:*

- MQ135 Gas Sensor: MQ135 sensor is used for detecting the gasses like CO, CO_2, NH_3. The sensor will continuously monitor the surrounding atmosphere.
- DHT11 Sensor: The DHT11 sensor provides real -time data on temperature and humidity around. This information can be valuable reference to explain air quality data, as temperature and humidity can affect the behaviour and concentration of some pollutants.
- ESP8266: ESP8266 act as the brain of the system. It will read data from MQ135 and DHT11 sensor. This sensor calculates the data of abnormal gas levels and

temperature/humidity and to handle the power controlled by potentially piezoelectric sensors.

- Webcam: The webcam is the most important component in our accident detection system part as it is responsible for capturing continuous real-time video frames from monitored environment.
- OLED: OLED screen is used to display the real time local visualization of the air quality parameter such as harmful gases, temperature, and humidity.

III. DESIGNED MODEL

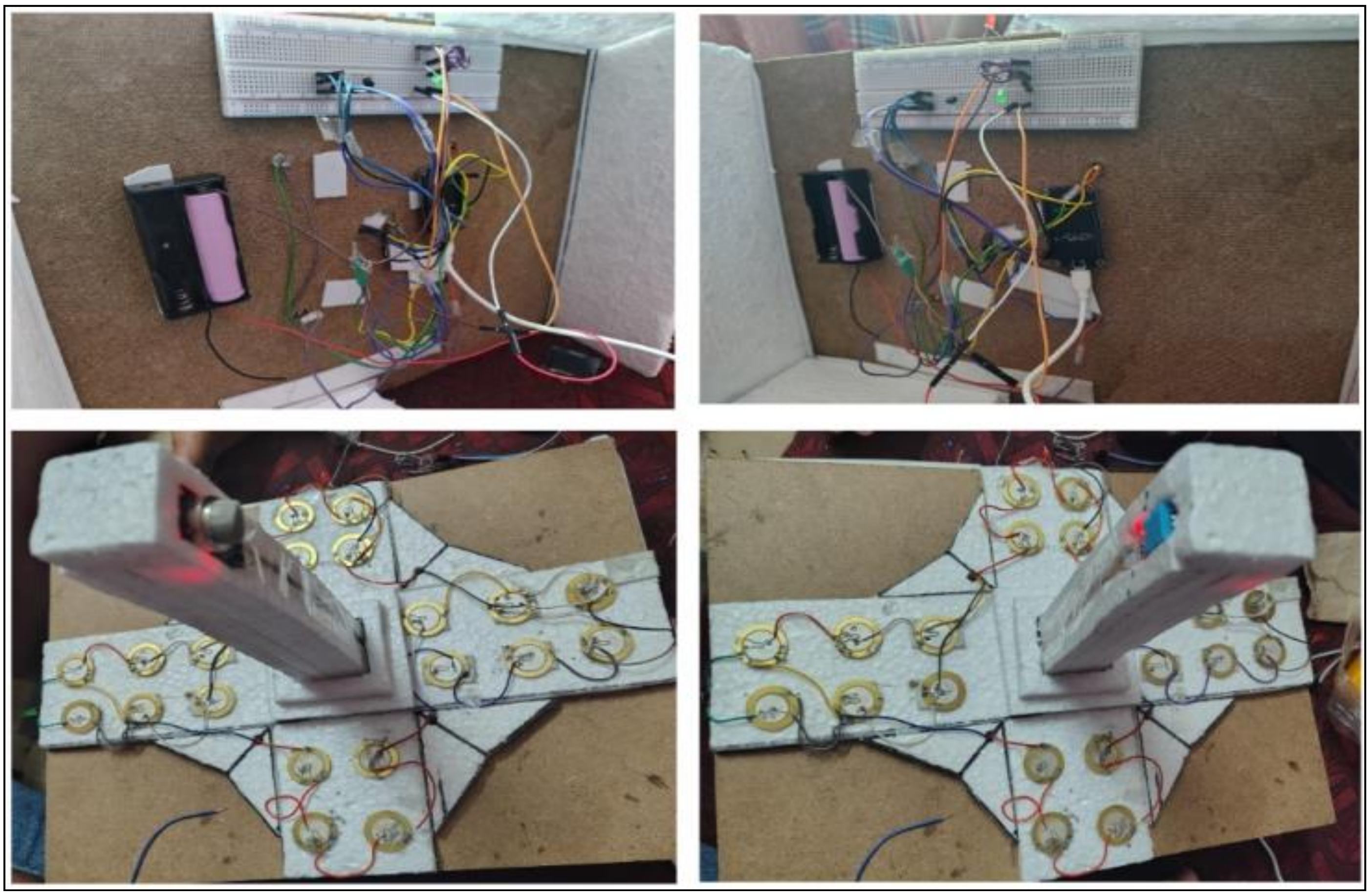


Fig 2 Project Prototype

The Figure-3 demonstrates our working project prototype. Here the combination of harvesting renewable energy, sensing the environment with IoT technology, and utilizing computer vision AI splits a smart city's infrastructure problem into manageable components and provides room for scaling as a sustainable solution.

IV. RESULT AND DISCUSSION

The proposed model uses a temperature and humidity DHT11 sensor, and an MQ135 sensor for air quality monitoring (PPM). These sensors receive atmospheric data and send it to the ESP8266 Wi-Fi module in the form of analog signal. The received analog signals are interpreted by ESP8266 to determine the gas concentration, temperature and humidity. The microcontroller then sends the data to the Google Script. That Google Script logs the values in one of the cells in a Google Spreadsheet, as illustrated in table I, thus enabling automated, real-time tracking without manual effort.

Table 1 Excel Sheet Output

Date	Time	Temperature (in °C)	Humidity	AirPPM
14/05/2025	16:18:24	35	52	161
14/05/2025	16:18:59	35	53	375
14/05/2025	16:19:35	35	53	340
14/05/2025	16:20:11	34	55	322
14/05/2025	16:20:46	34	55	310
14/05/2025	16:21:22	34	56	298

14/05/2025	16:22:06	34	56	250
14/05/2025	16:23:27	34	57	250
14/05/2025	16:24:02	34	57	250
14/05/2025	16:24:38	33	56	250
14/05/2025	16:25:13	33	57	250

Informational graphs further below, provide a more systemized view for the end users.

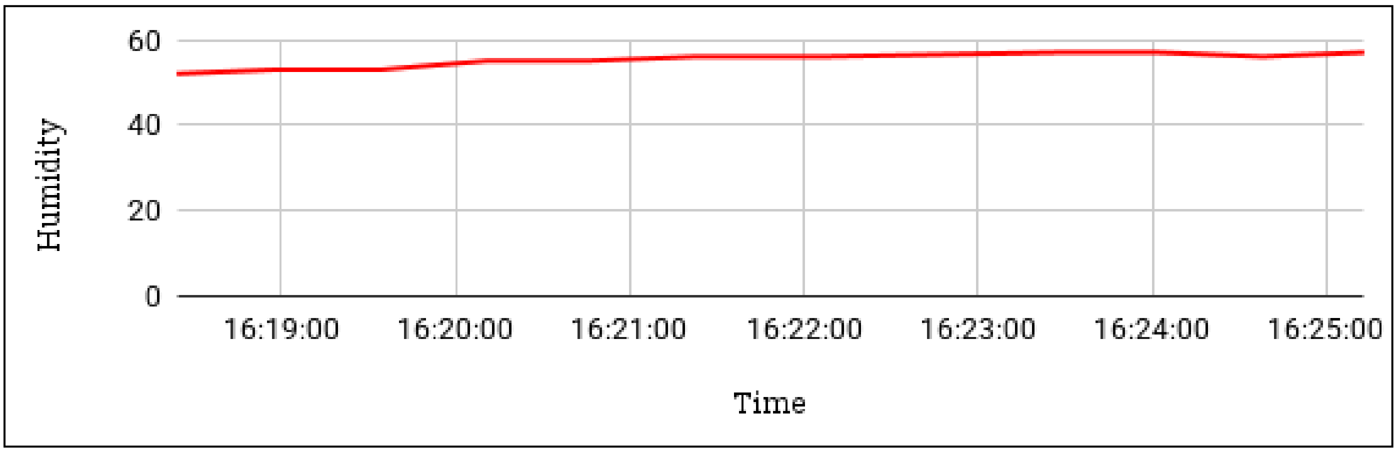


Fig 3 Humidity vs. Time Graph

Data representation for the system is classified by the information being collected by the system. These illustrations (in figure 6, 7 and 8) demonstrate the extent to which the system is capable of monitoring environmental conditions over time. The observed decrease in temperature alongside stable air quality readings affirms the effectiveness and sensitivity of the monitoring setup. Such data at different geographical locations can strengthen the case for supporting Air Quality Index (AQI) evaluations, thus proving beneficial for researchers and policymakers.

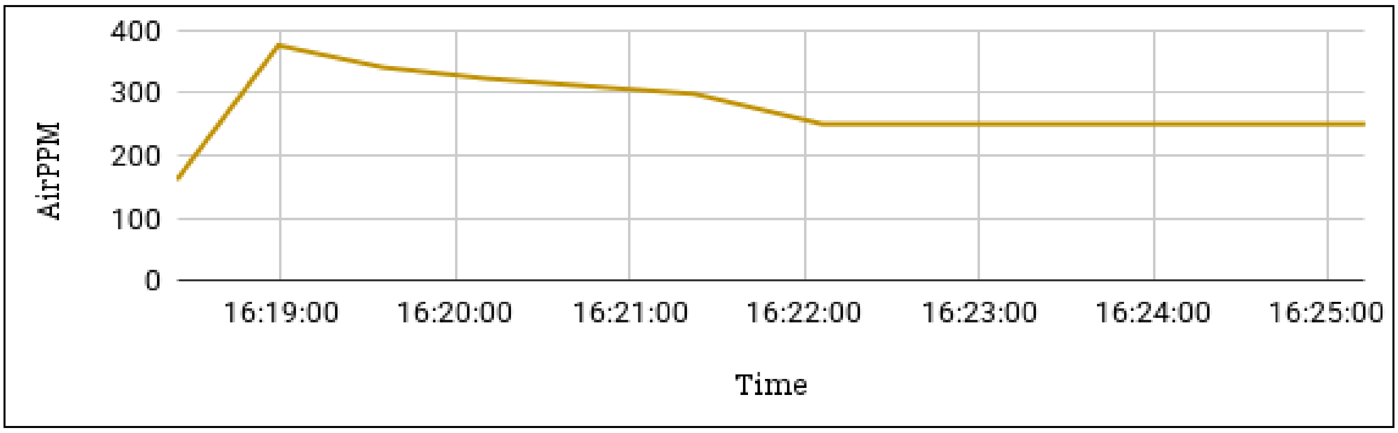


Fig 4 Air PPM vs. Time Graph

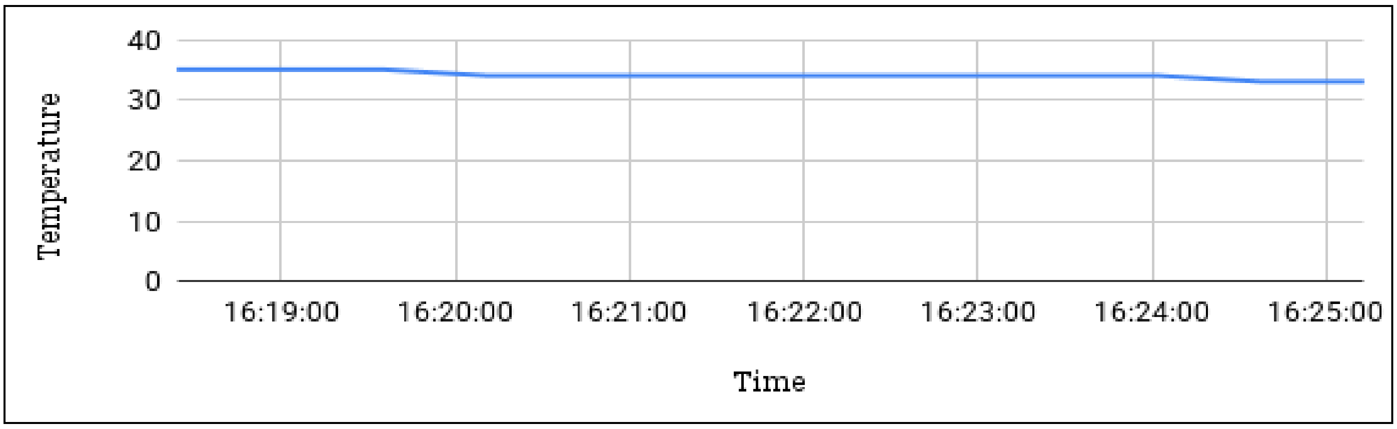
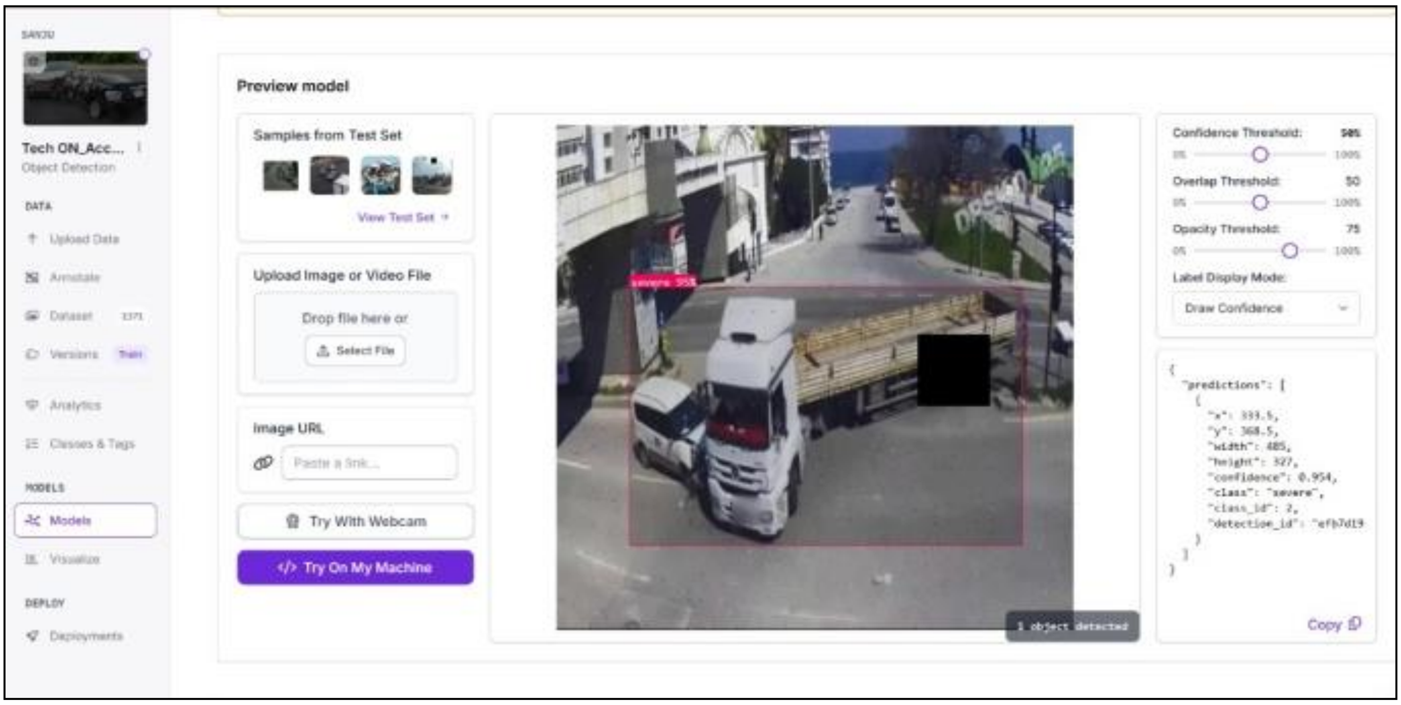


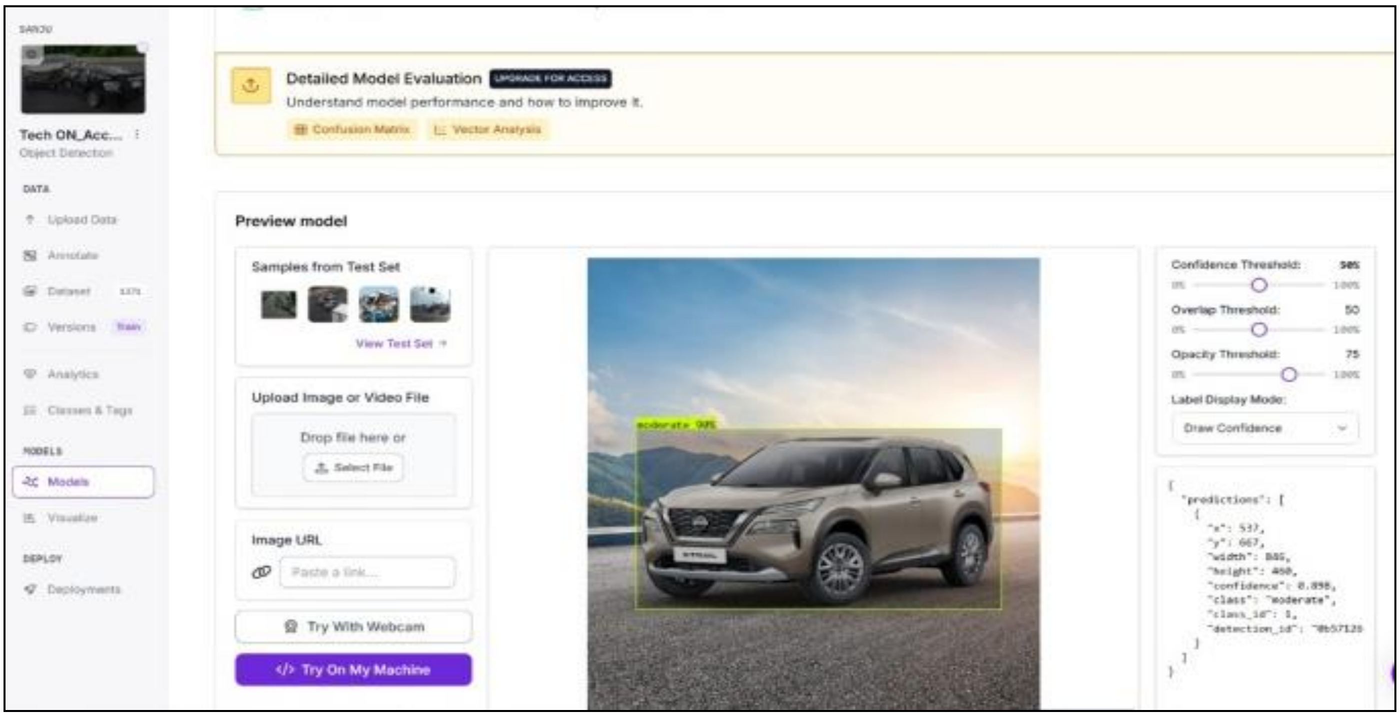
Fig 5 Temperature vs. Time Graph



Confidence Threshold: 50%
Overlap Threshold: 50
Opacity Threshold: 75
Label Display Mode: Draw Confidence

```
{
  "predictions": [
    {
      "x": 333.5,
      "y": 368.5,
      "width": 485,
      "height": 327,
      "confidence": 0.954,
      "class": "severe",
      "class_id": 2,
      "detection_id": "efb7d19
    }
  ]
}
```

(a)



Confidence Threshold: 50%
Overlap Threshold: 50
Opacity Threshold: 75
Label Display Mode: Draw Confidence

```
{
  "predictions": [
    {
      "x": 537,
      "y": 667,
      "width": 846,
      "height": 460,
      "confidence": 0.898,
      "class": "moderate",
      "class_id": 1,
      "detection_id": "0b57128
    }
  ]
}
```

(b)

Fig 6: (a) Accident Detected through Footage (b) No-Accident Detection through Footage

To detect accidents, YOLOv11 model has been customized with additional data and it performed robustly in classifying normal vs. anomalous scenarios as shown in figure 8(a) and 8(b). For example, the system processes the image of an undamaged vehicle as shown in figure 8(a) and correctly mark incident as moderate. On the other hand, for a collision scenario as shown in figure 8(b), the model accurately captures and marks the necessary components of different levels of impact. Real-time object detection is critical for autonomous cars, surveillance, and medical imaging, and YOLOv11's architecture performs entire image processes in a single forward pass, providing immediate relations with high accuracy.

V. CONCLUSION

In the proposed model, piezoelectric sensors effectively generate sufficient energy to power the devices like DHT11 & MQ135 sensors and a webcam. The integration of YOLOv11 enables accurate and real-time accident detection, enhancing road safety. The system promotes sustainability by utilizing wasted kinetic energy from foot traffic, reducing dependency on conventional power sources. This model has significant potential for smart city applications, offering a cost-effective, eco-friendly solution for urban monitoring and safety. The information on the OLED screen spreads awareness to the people on the street. People can take effective measures to prevent breathing polluted air. The excel data will beneficiate government surveys. It can be deployed in high-footfall areas like sidewalks, transit hubs, and public spaces to improve energy efficiency and public security. In future we may can enhance energy storage capacity to ensure uninterrupted operation during low foot traffic, expand the AI model to detect a wider range of incidents (e.g., crimes or emergencies), integrating IoT for centralized data analysis and automated alerts to authorities. By merging renewable energy harvesting with AI-driven surveillance, this model paves the way for smarter, safer, and more sustainable cities, proving that even small-scale innovations can have a far-reaching impact on urban living.

REFERENCES

[1]. K. Zheng, S. Zhao, Z. Yang, X. Xiong and W. Xiang, "Design and Implementation of LPWA-Based Air Quality Monitoring System," in IEEE Access, vol. 4, pp. 3238-3245, 2016

[2]. Khodve, Shilpa R., and A. N. Kulkarni. "Web Based Air Pollution Monitoring System (Air Pollution Monitoring Using Smart Phone)." International Journal of Science and Research (IJSR) 5.3 (2016): 266-269.

[3]. Tejal Lengare, Pallavi Sonawane, Prachi Gunjal, Shubham Dhire, "Accident Detection & Avoidance System in Vehicles" Engpaper Journal.

[4]. Kelvin Rinaldi da Luz, João Elison da Rosa Tavares, Jorge Luis Victória "RoadLytics: Road Accidents Analytics Using Artificial Intelligence" Engpaper Journal 2021.

[5]. Foggia, Pasquale, et al. "Audio surveillance of roads: A system for detecting anomalous sounds." IEEE transactions on intelligent transportation systems 17.1 (2015): 279-288.

[6]. J. A. Rosales, J. G. Samson and C. Maderazo, "Motor Vehicle Crash Detection Using Yolov8 Algorithm," 2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), Malang, Indonesia, 2023, pp. 365-371

[7]. Hidayatullah, Priyanto & Syakrani, N. & Sholahuddin, Muhammad & Gelar, Trisna & Tubagus, Refdinal. (2025). YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review.

APPENDIX - I

Setting up the Adriano IDE code:

Step 1 - initializing OLED.

```
void setup() {
Serial.begin(9600);
if(Idisplay.begin(SSD1306_SWITCHCAPVCC, 0x3D)) {
Serial.println(F("SSD1306 allocation failed"));
for(;;);
}
display.display();
delay(2000);
display.clearDisplay();
dht.begin();
```

Step 2 - reading data from MQ135 sensor and DHT11 sensor.

```
void loop() {
delay(2000);
// Read MQ135 sensor (analog value)
int sensorValue = analogRead(A0);
Float p= map(sensorValue, 0, 1024, 0, 1000); // Adjust calibration as needed
// Read DHT11 sensor
float hdht.readHumidity();
floatt dht.readTemperature();
Serial.print("Temp");
Serial.print(t);
Serial.print("HUM");
Serial.println(h);
Serial.print(" PPM= ");
Serial.println(p);
it (int) t;
ih= (int) h;
ip (int) p;
sendData(it, ih, ip);
delay(30000);
```

Step 3- connect to wi-fi

```
Serial.print("connecting to ");
Serial.println(ssid);
WiFi.mode (WIFI STA);
WiFi.begin(ssid, password);
while (WiFi.status() != WL CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
```

Step 4 - display on OLED

```
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306 WHITE);
display.setCursor(0,0);
display.println("Air Quality Monitor");
display.println("--
display.print("CO2: "); display.print(p); display.println(" ppm");
display.print("Temp: "); display.print(t); display.println("C");
display.print("Humidity: "); display.print(h); display.println("%");
display.display();
```

Step 5 - send data to PC

```
void sendData(int tem, int hum, int ppm)
{
client.setInsecure();  // Skip certificate validation (for development use)
Serial.print("Connecting to ");
Serial.println(host);
if (!client.connect(host, httpsPort)) {
Serial.println("Connection failed");
return;
}
```

Step 6 - sends the collected data to the script

```
String string temperature String(tem, DEC);
String string_humidity String(hum, DEC);
String stringAirPPM = String(ppm, DEC);
String url="/macros/s/" + GAS_ID + "/exec?Temperature=" + string temperature + "&Humidity=" + string humidity+
"&AirPPM=" + string AirPPM;
Serial.print("requesting URL: ");
Serial.println(url);
client.print(String("GET") + url + " HTTP/1.1\r\n" +
"Host: "+ host + "\r\n" +
"User-Agent: BuildFailureDetectorESP8266\r\n" +
"Connection: close\r\n\r\n");
Serial.println("request sent");
while (client.connected()) {
String line = client.readStringuntil('\n');
if (line = "\r") {
Serial.println("headers received");
break;
}
}
String line= client.readStringUnti1('\n');
if (line.startswith("{\"state\":\"success\"")) {
Serial.println("esp8266/Arduino CI successfull!");
} else{
Serial.println("esp8266/Arduino CI has failed");
}
Serial.println("reply was:");
Serial.println("=====");
Serial.println(line);
Serial.println("==========");
Serial.println("closing connection");
}
```

APPENDIX - II

Script for excel data entry:

```
var sheet _id= '1HSCcvn4chgIt-zMrqNZXBACv1CDRovvsyKRqwGFvqd8': // Spreadsheet ID
var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
var newrow= sheet.getLastRow() + 1;
var rowData=[]:
var Curr_ Date =new Date();
rowData[0] = Curr Date; // Date in column A
var Curr-Time =Utilities.formatDate(Curr_Date, "Asia/Calcutta", "HH:mm:ss');
rowData[1] = Curr_Time; // Time in column B
for (var param in e.parameter) {
Logger.log('In for loop, param=' + param);
var value = stripQuotes(e.parameter[param));
Logger.log(parame+':'+e.parameter [param]);
switch (param) {
case 'Temperature':
rowData[2]= value; // Temperature in column C
result= 'Temperature Written on column C';
break;
case 'Humidity':
rowData[3] = value; // Humidity in column D
result += ' , Humidity Written on column D';
break;
case AirPPM:
rowData[4] = value; // Air PPM in column E
result += ' , AirPPM Written on column E';
break;
default:
result "unsupported parameter";
}
```