Article

Use of IoT with Deep Learning for Classification of Environmental Sound and Detection of Gases

Priya Mishra , Naveen Mishra , Dilip Kumar Choudhary , Prakash Pareek , Manuel José Cabral dos Santos Reis *

Posted Date: 4 July 2024

doi: 10.20944/preprints202407.0389.v1

Keywords: IoT; deep learning; CNN model; environmental sounds; gases detection; MQ6 sensor; MQ135 sensor; DHT11 sensor; IFTTT

Article

Use of IoT with Deep Learning for Classification of Environment Sounds and Detection of Gases

Priya Mishra [1], Naveen Mishra [1], Dilip Kumar Choudhary [1], Prakash Pareek [2] and Manuel José Cabral dos Santos Reis [3,*]

[1] Department of Communication Engineering, School of Electronics Engineering, Vellore Institute of Technology, Vellore, 632014, T.N., India; pmish2002@gmail.com (P.M.); naveenmishra.ece@gmail.com (N.M.); dilip.choudhary6@gmail.com (D.K.C.)
[2] Department of ECE, Vishnu Institute of Technology, Bhimavaram, A.P., India; prakash.p@vishnu.edu.in
[3] UTAD/IEETA, Engineering Department, 5001-801 Vila Real, Portugal
* Correspondence: mcabral@utad.pt

Abstract: This article presents an Internet of Things (IoT)-enabled system meticulously engineered for comprehensive environmental assessment, encompassing air quality, temperature, and humidity. Simultaneously, the devised system integrates cutting-edge MQ6 sensors, adept at detecting ambient gases, with a primary focus on monitoring Liquefied Petroleum Gas (LPG) and Butane gases concentrations in the atmosphere. This distinctive feature not only facilitates the identification of hazardous gases leakages, particularly common in household LPG usage, but also empowers preemptive measures. The environmental parameters, air quality, temperature, and humidity, are precisely gauged through the utilization of MQ135 and DHT11 sensors. The controlling and managing of the hardware component of the system (i.e., device) is assured by an Arduino UNO, that orchestrates the collection of sensor data, transmitting it seamlessly to both the ThingSpeak cloud and a Convolutional Neural Network (CNN) model. By interfacing with ThingSpeak, the device ensures the systematic storage and visualization of environmental data, streamlining monitoring and analysis processes. In the event of perilous gases detection, the system promptly triggers an alarm, concurrently dispatching a real-time notification to the device owner via the integrated If-This-Then-That (IFTTT) application programming interface_(API). To further enrich environmental awareness, an embedded microphone captures audio recordings, subsequently processed by a CNN-based deep learning model. This model employs advanced techniques such as spectrogram generation, Convolution2D, and MaxPooling2D within a Sequential Model architecture, to attain elevated levels of accuracy, furnishing insightful interpretations of ongoing activities within the environment.

Keywords: IoT; deep learning; CNN model; environmental sounds; gases detection; MQ6 sensor; MQ135 sensor; DHT11 sensor; IFTTT

1. Introduction

Air pollution, a pervasive and escalating global challenge, poses severe threats to public health, particularly in densely populated urban areas. The intricate web of causative factors encompasses vehicular emissions, industrial activities, and the multifaceted consequences of climate change. Governments worldwide are grappling with the complexity of this crisis, evident in initiatives like the commitment of European countries to transition to electric vehicles by 2030. Meanwhile, India has set an ambitious target of achieving a similar transition by 2025, reflecting the urgency to curb the environmental and health ramifications of air pollution. In the recent months, Delhi, India's capital, has witnessed a marginal improvement in air quality levels. However, persistent reports highlight concerning concentrations of PM 2.5 – fine particles capable of deeply penetrating the lungs [1] – in various pockets of the city and the broader National Capital Region (NCR). Delhi, ranking among the most polluted cities globally, becomes a focal point for intense public debates, particularly as winter sets in, drawing widespread international attention to its ongoing battle with air pollution.

A particularly alarming facet in the Indian context is the frequency of Liquefied Petroleum Gas (LPG) accidents, surpassing 1,500 incidents daily. These accidents result in a corresponding number of fatalities, disproportionately affecting the youth. The gravity of these incidents extends beyond individual households, underscoring the urgent need for technological interventions to prevent such tragedies. In this landscape, the Internet of Things (IoT) emerges as a dynamic and transformative technological frontier, particularly within the automotive sector, serving as a foundational element for the unfolding era of Industry 4.0. In the pursuit of addressing the escalating global challenge of air pollution, this research draws inspiration from various foundational works in the fields of environmental monitoring, machine learning, and the IoT. Notably, studies have significantly influenced the exploration of machine listening systems within Computational Auditory Scene classification (CASA) [2]. Moreover, the integration of IoT technology in environmental monitoring has the potential to comprehend and mitigate the impacts of air pollution [3]. Additionally, the research has highlighted the pivotal role of IoT applications in ensuring environmental safety [4,5]. The intersection of IoT technology and smart home management has put emphasis on the potential of IoT to create intelligent and interconnected living environments [6,7].

In this research we propose and present an IoT-enabled system for comprehensive environmental assessment, encompassing air quality, temperature, and humidity. This system integrates MQ6 sensors, with the aim of detecting ambient gases, with a primary focus on monitoring LPG and Butane gases concentrations in the atmosphere. This distinctive feature not only facilitates the identification of hazardous gases leakages, particularly common in household LPG usage, but also empowers preemptive measures. The environmental parameters, air quality, temperature, and humidity, are also gauged through the utilization of MQ135 and DHT11 sensors. With the main aim of controlling the costs of the hardware component of the system an Arduino UNO is used to control the collection of sensor data, transmitting it seamlessly to both the ThingSpeak cloud and a Convolutional Neural Network (CNN) model. Through its connection to ThingSpeak, this device guarantees the methodical preservation and presentation of environmental data, simplifying the monitoring and analysis procedures. If hazardous gases are detected, the system immediately activates an alarm and simultaneously sends a real-time alert to the device owner through the integrated If-This-Then-That (IFTTT) application programming interface (API). To enhance environmental awareness, a built-in microphone records audio from the environment, which is then analyzed by a CNN-based deep learning model. This model utilizes methods like spectrogram creation, Convolution2D, and MaxPooling2D in a Sequential Model design to achieve high levels of precision, offering valuable insights into ongoing environmental activities.

As this research explores the potential of IoT technology in tandem with deep learning for environmental monitoring, it acknowledges the comprehensive body of work in the cited papers. This convergence of insights from diverse disciplines forms the basis for an innovative approach to address the multifaceted challenges posed by air pollution, with the intention of fostering a healthier and more sustainable future. Additionally, findings from research on recurrence quantification analysis features for auditory scene classification [8], and principles, algorithms, and applications of computational auditory scene analysis [9] enhance the comprehension of auditory scene classification principles and algorithms, thereby strengthening the groundwork for the proposed research. In the realm of machine learning, particularly deep learning, this research leverages insights from a study focused on understanding the effective receptive field in deep convolutional neural networks [10]. The work offers valuable perspectives on the architectural considerations for CNNs, a crucial component in the proposed Acoustic Scene Classification (ASC) deep learning model. The significance of these foundational studies is further underscored by the inclusion of datasets, such as one curated by K. J. Piczak [11], serving as a benchmark for environmental sound classification studies.

2. Device Design and Implementation

2.1. Proteus Schematic for IoT Device

The schematic circuit for the entire device was crafted using Proteus 8 software, offering a comprehensive visualization of the device's architecture and functionality. Data acquisition from the sensors is orchestrated by the Arduino UNO microcontroller, serving as the central processing unit of the device. Through integration, sensor data is transmitted to the ThingSpeak cloud, enabling remote monitoring and analysis of environmental parameters. In its physical representation, the virtual device retains full functionality. It swiftly transfers sensor data to the cloud, ensuring real-time visualization within a remarkably short timeframe of 2-3 minutes. This expedited data transfer process is complemented by the system's capability to trigger remote notifications to the user via Wi-Fi, enhancing user awareness and responsiveness to environmental dynamics. Central to its functionality are the MQ135 and MQ6 gas sensors, adept at detecting a range of gases including NH3, NOx, Alcohol, Benzene, CO2, and LPG. While the primary focus lies in air-quality monitoring [12–14], the inclusion of the MQ6 LPG detection sensor underscores the virtual device's versatility in identifying potentially hazardous gases, thereby safeguarding households from the perils of LPG cylinder blasts.

Furthermore, the virtual device's customization capabilities empower users to tailor its functionality to their specific needs and industry requirements. The integration of the DHT11 sensor provides users with real-time information on humidity and temperature, enriching their understanding of environmental conditions. The user interface is facilitated through an LCD screen, offering clear visualizations of air quality measured in parts per million (PPM), alongside humidity and temperature readings. Additionally, a buzzer linked to the Arduino serves as an alert mechanism, promptly notifying users in the event of LPG detection. To facilitate seamless data transfer from the Arduino to the ThingSpeak cloud, Virtual Serial Port Emulator (VSPE) and a Python Script are employed as efficient alternatives to NodeMCU. This ensures reliable and uninterrupted communication between the device and the cloud infrastructure, enhancing the overall efficacy and responsiveness of the system. Figure 1 presents a block diagram elucidating the components and interactions of the developed device, providing a comprehensive overview of its architecture and functionality.

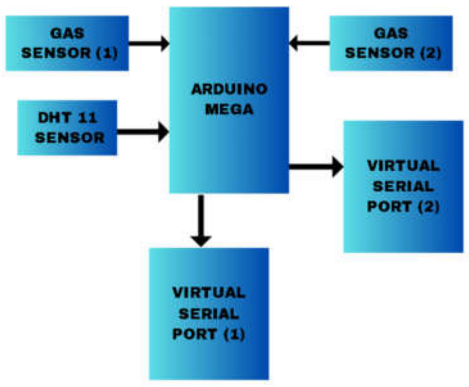


Figure 1. Main block diagram of the proposed device.

2.2. Integration of ThingSpeak Cloud with Arduino and Alerting User with the Help of IFTTT

Within ThingSpeak, there are two applications available for customizing triggers: ThingHTTP and React. React comes into play when the Arduino detects LPG, initiating the trigger ticket and interfacing with ThingHTTP. This, in turn, activates two If-This-Then-That (IFTTT) applets.

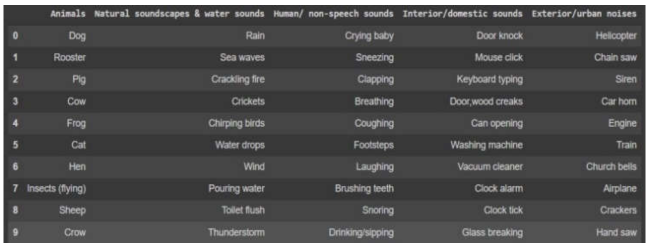
The first applet is designed to send an email to the user upon LPG detection, while the second applet delivers a notification directly to the user's mobile device through the IFTTT application. Once

React initiates the ThingHTTP process, ThingHTTP calls the API for the IFTTT applet, prompting IFTTT to send the notification to the user. Remarkably, this entire process is completed within a swift 3-minute timeframe from the Arduino detecting the gas. This showcases the robust and efficient nature of the product in promptly securing the environment from potential harm.

2.3. CNN based ASC Deep Learning Model

The linkage formed between the IoT device and the web application permits users to periodically send audio captures of their surroundings from the IoT device to the web application. This smooth integration greatly enhances the capacity for environmental surveillance, empowering users to remotely supervise their surroundings and proactively ensure safety and well-being.

The goal is to recognize numerous sound environments or circumstances in the surroundings. For instance, if a dog barks, the current scheme is designed to classify the sound pattern specifically associated with a dog's bark. Acoustic scene classification aims to assign a test recording to a predefined category that characterizes the recording environment. The dataset utilized comprises a total of 2000 environmental audio recordings spread across various classes and subclasses. It encompasses five primary classes representing different categories of sounds commonly heard in everyday environments, with each class further divided into ten subclasses, each representing a specific kind of sound resulting from its respective class. Notably, each subclass consists of 40 distinct sound recordings, resulting in a total of 2000 sound recordings used for training the model. CNNs have emerged as the predominant choice for tasks like acoustic scene classification. The most effective CNNs primarily utilize audio spectrograms as input, drawing architectural inspiration mainly from computer vision principles [13]. Consequently, all audio files are transformed into spectrograms using the 'librosa' Python library. This proposal examines the capacity of convolutional neural networks to classify concise audio recordings of environmental sounds. Figure 2 presents a screen-shot exemplifying the environmental audio files of the used dataset.



	Animals	Natural soundscapes & water sounds	Human/ non-speech sounds	Interior/domestic sounds	Exterior/urban noises
0	Dog	Rain	Crying baby	Door knock	Helicopter
1	Rooster	Sea waves	Sneezing	Mouse click	Chain saw
2	Pig	Crackling fire	Clapping	Keyboard typing	Siren
3	Cow	Crickets	Breathing	Door,wood creaks	Car horn
4	Frog	Chirping birds	Coughing	Can opening	Engine
5	Cat	Water drops	Footsteps	Washing machine	Train
6	Hen	Wind	Laughing	Vacuum cleaner	Church bells
7	Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
8	Sheep	Toilet flush	Snoring	Clock tick	Crackers
9	Crow	Thunderstorm	Drinking/sipping	Glass breaking	Hand saw

Figure 2. Dataset of the environmental audio files, where columns are classes and rows are subclasses.

Upon training the model, it was observed that the existing dataset lacked adequacy for achieving satisfactory accuracy. To address this, Data Augmentation was employed, incorporating white noise into copies of the original dataset, resulting in a total of 4000 training examples. Subsequently, 10% of the data is reserved for testing, and another 10% for model evaluation.

Following augmentation, four Convolution2D layers are introduced to enhance accuracy, complemented by the addition of MaxPooling2D to mitigate overfitting. Prior to model compilation, the data is flattened from 2D to 1D using 'flatten'. The 'dense' layer is utilized to connect all layers, and 'dropout' is applied to randomly deactivate selected neurons after each epoch, ensuring ongoing improvement in model accuracy. Finally, the Sequential model is compiled using the 'adam' optimizer, initiating the model training phase with 80% of the dataset and setting the epochs value to 30. Figures 3 and 4 present a diagram and a summary of the ASC model.

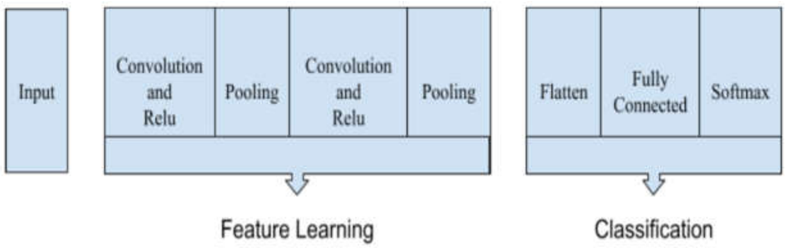
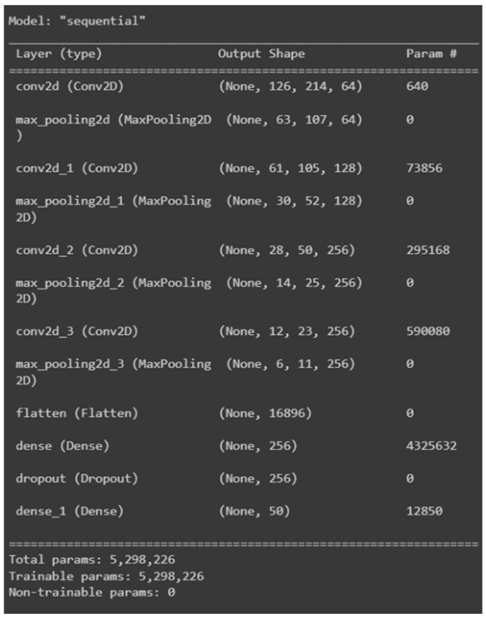


Figure 3. Flowchart of the ASC model.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 214, 64)      640

 max_pooling2d (MaxPooling2D  (None, 63, 107, 64)      0
 )

 conv2d_1 (Conv2D)           (None, 61, 105, 128)      73856

 max_pooling2d_1 (MaxPooling  (None, 30, 52, 128)      0
 2D)

 conv2d_2 (Conv2D)           (None, 28, 50, 256)       295168

 max_pooling2d_2 (MaxPooling  (None, 14, 25, 256)      0
 2D)

 conv2d_3 (Conv2D)           (None, 12, 23, 256)       590080

 max_pooling2d_3 (MaxPooling  (None, 6, 11, 256)       0
 2D)

 flatten (Flatten)           (None, 16896)             0

 dense (Dense)               (None, 256)               4325632

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 50)                12850

=================================================================
Total params: 5,298,226
Trainable params: 5,298,226
Non-trainable params: 0
```

Figure 4. Summary of the ASC Model.

2.4. Collaborating IoT Device with ASC Model

In the previous sections, we discussed the development of two core components: the IoT device responsible for environmental monitoring and the Audio Scene Classification (ASC) model designed for sound classification. These components operated as distinct entities, each serving a specific purpose within the broader context of the system. However, to realize the full potential of the system and provide users with comprehensive insights into their environment, integration of these components is essential. In the final phase, a web application serves as the nexus, seamlessly bringing together the functionalities of the IoT device and the ASC model. This web application acts as a centralized platform where users can access and interact with the data collected by the IoT device and processed by the ASC model. Figure 5 presents a global view of the integration of the main components of the developed system.

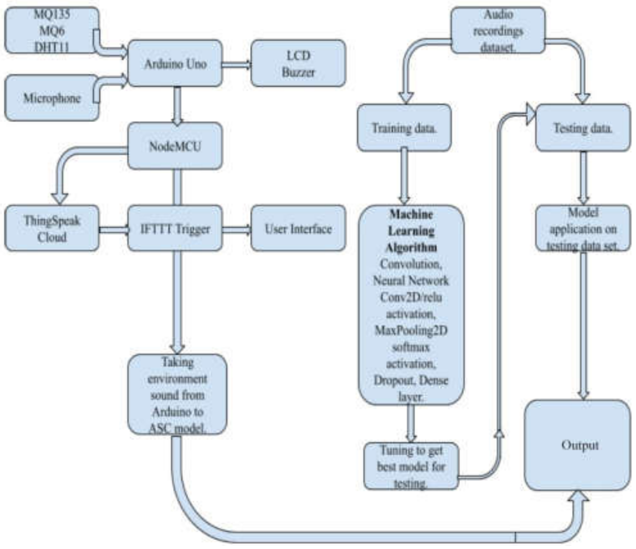

MQ135
MQ6
DHT11
Microphone
Arduino Uno
LCD
Buzzer
NodeMCU
ThingSpeak Cloud
IFTTT Trigger
User Interface
Taking environment sound from Arduino to ASC model.
Audio recordings dataset.
Training data.
Testing data.
Machine Learning Algorithm
Convolution, Neural Network Conv2D/relu activation, MaxPooling2D softmax activation, Dropout, Dense layer.
Model application on testing data set.
Tuning to get best model for testing.
Output

Figure 5. Project demonstration after collaborating IoT device with the ASC Model.

The integration between the IoT device and the web application enables users to send audio recordings of their environment from the IoT device to the web application at regular intervals. This seamless communication is facilitated by a Python script scheduled to run every 5 minutes, ensuring continuous monitoring of the environment (Figure 6). Through this application, users can receive notifications about ongoing activities in the environment using the ASC model. Additionally, the web application provides information on the air quality, temperature, and humidity obtained through the Arduino sensors. Figure 7 presents an example visualization of these data, offering users insights into the current environmental conditions. Moreover, the system has the capability to trigger alarms and notifications in the event of detecting any hazardous gas, such as LPG in the context of this project. This comprehensive integration enhances environmental monitoring capabilities, empowering users to remotely monitor their surroundings and take proactive measures to ensure safety and well-being.

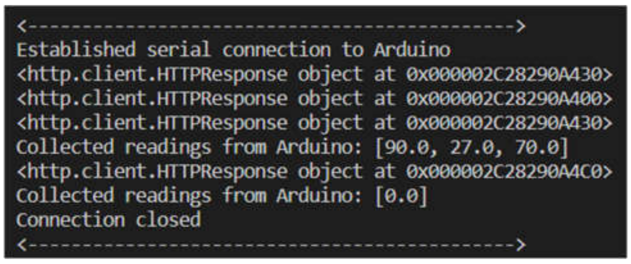

```
<--------------------------------------------->
Established serial connection to Arduino
<http.client.HTTPResponse object at 0x000002C28290A430>
<http.client.HTTPResponse object at 0x000002C28290A400>
<http.client.HTTPResponse object at 0x000002C28290A430>
Collected readings from Arduino: [90.0, 27.0, 70.0]
<http.client.HTTPResponse object at 0x000002C28290A4C0>
Collected readings from Arduino: [0.0]
Connection closed
<--------------------------------------------->
```

Figure 6. Python Script sending outputs from the Arduino to ThingSpeak channel.

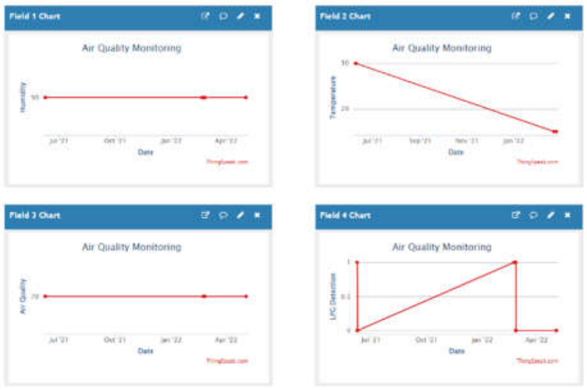


Figure 7. ThingSpeak visualizing the environment statistics.

In the Proteus schematic, all the necessary library files were incorporated, along with the inclusion of the Arduino Code. The output screen utilizes the Virtual Terminal to display the air quality, temperature, and humidity of the environment using data from MQ135 and DHT11 sensors. Meanwhile, the other terminal indicates the detection of LPG, represented in binary where 0 signifies 'not detected' and 1 denotes 'detected' (please refer to Figure 10). These virtual terminals are exclusively linked to the Virtual Serial Port Emulator (VSPE) ports. The next step involves transmitting the Arduino outputs to a Python Script through these virtual ports.

The Python Script plays a crucial role in ensuring seamless communication between the Arduino-based IoT device and the ThingSpeak cloud platform. Integrated with the Write API keys of the ThingSpeak channel, the script acts as a bridge, facilitating the transmission of real-time sensor data from the device to the cloud. To enable continuous monitoring and timely updates, the Python Script is configured to run at regular intervals, typically every 20 seconds. This scheduling ensures that sensor readings from the Arduino are consistently captured and transmitted to the ThingSpeak channel without interruption. Once executed, the Python Script initiates the data transmission process, packaging the sensor readings into structured data packets compatible with the ThingSpeak platform. These packets, containing vital information such as air quality measurements, temperature, humidity, and gas detection status, are then sent over the internet to the designated ThingSpeak channel. Upon receiving the data, ThingSpeak processes and stores it in real-time, allowing users to access up-to-date visualizations and analytics of their environment at any given moment. Through the ThingSpeak channel interface, users can view comprehensive graphs, charts, and statistics depicting environmental parameters, enabling them to gain valuable insights into their surroundings (Figure 7).

Crucially, the real-time nature of this data transmission ensures that users are promptly alerted to any significant changes or anomalies detected by the IoT device. In the event of gas detection, for example, the system triggers an alert within a remarkably short timeframe of 2-3 minutes, notifying the user via the integrated notification system. Overall, the integration of the Python Script with the ThingSpeak platform ensures that the IoT device operates seamlessly, continuously updating its cloud-based repository with real-time sensor data. This enables users to monitor their environment remotely, stay informed about critical events, and take timely actions to ensure safety and well-being.

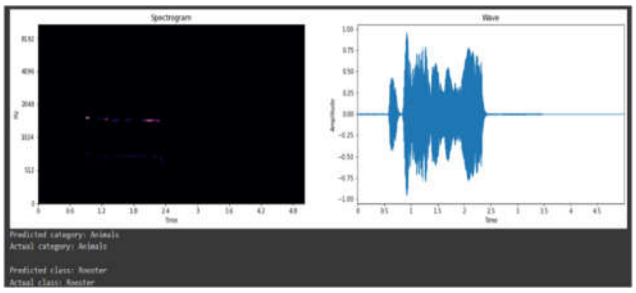
3. Results and Discussion

The detection and measurement module of the system underwent extensive testing to assess its ability to accurately detect and measure environmental factors such as temperature, humidity, and various gases, including LPG. Testing primarily involved simulations using the Proteus software, facilitating controlled experimentation and evaluation of the system's performance. Within the Proteus environment, tests were conducted to simulate diverse environmental conditions and scenarios, enabling precise control over factors like temperature, humidity, and gas concentrations. This approach allowed for thorough assessment of the system's response across different conditions.

The simulation-based testing spanned over a period of two months, during which extensive tests were performed to assess the system's accuracy, reliability, and response time. The duration of the testing period ensured comprehensive evaluation of the system's performance under diverse conditions and scenarios. The ASC model underwent extensive testing to evaluate its ability to accurately classify environmental sounds. The testing process involved training the model on a diverse dataset and evaluating its performance using standard metrics such as accuracy, precision, recall, and F1-score.

The dataset used for training and testing the ASC model comprised environmental sounds commonly encountered in everyday situations. The dataset was divided into five primary classes, each representing a different category of environmental sound. Within each class, there were ten subclasses, each representing a specific type of sound within that category as shown in Figure 2.

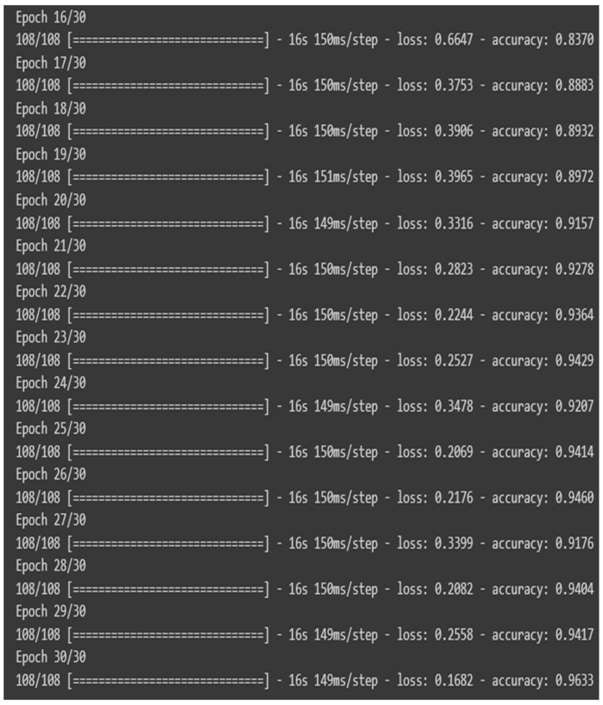
To evaluate the performance of the ASC model, a separate validation dataset was used, consisting of audio recordings not included in the training dataset. The model was trained using standard machine learning techniques, and its performance was evaluated using the validation dataset. Figure 8 presents an example of the output prediction of the model from a random input from the testing set. The accuracy of the ASC model was measured by comparing the model's predictions with the ground truth labels for each audio recording in the validation dataset. The accuracy metric represents the percentage of correctly classified audio recordings out of the total number of recordings in the validation dataset. After successfully training and testing the ASC model, an impressive accuracy of 96% was achieved. This accuracy metric indicates that the model correctly classified 96% of the audio recordings in the validation dataset as shown in Figure 9. Additionally, precision (0.95), recall (0.94), and F1-score (0.945) metrics were calculated to provide a more comprehensive evaluation of the model's performance. In addition to testing the detection and measurement component of the system in simulated environments, simulations were conducted specifically to evaluate the system's ability to detect hazardous gases such as LPG leaks. These simulations aimed to assess whether the system could accurately detect real hazardous gas leaks and promptly alert the user to mitigate potential risks. Several simulation scenarios were created to represent different levels of LPG gas leakage, ranging from minor leaks to significant concentrations of gas.



```
Predicted category: Animals
Actual category: Animals

Predicted class: Rooster
Actual class: Rooster
```

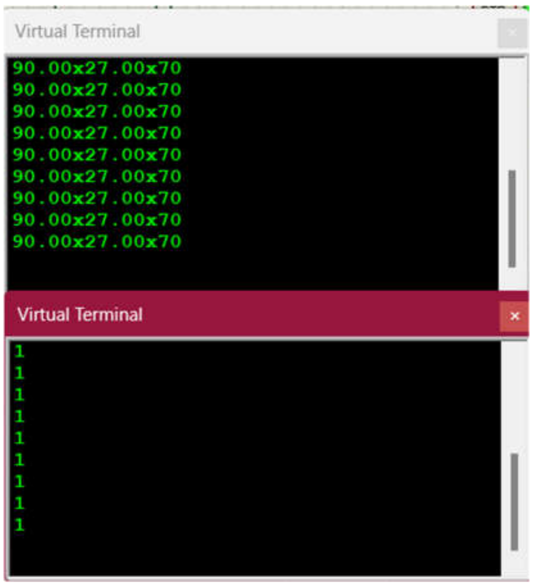
Figure 8. Output Prediction of the model from a random input from the testing set.



```
Epoch 16/30
108/108 [==============================] - 16s 150ms/step - loss: 0.6647 - accuracy: 0.8370
Epoch 17/30
108/108 [==============================] - 16s 150ms/step - loss: 0.3753 - accuracy: 0.8883
Epoch 18/30
108/108 [==============================] - 16s 150ms/step - loss: 0.3906 - accuracy: 0.8932
Epoch 19/30
108/108 [==============================] - 16s 151ms/step - loss: 0.3965 - accuracy: 0.8972
Epoch 20/30
108/108 [==============================] - 16s 149ms/step - loss: 0.3316 - accuracy: 0.9157
Epoch 21/30
108/108 [==============================] - 16s 150ms/step - loss: 0.2823 - accuracy: 0.9278
Epoch 22/30
108/108 [==============================] - 16s 150ms/step - loss: 0.2244 - accuracy: 0.9364
Epoch 23/30
108/108 [==============================] - 16s 150ms/step - loss: 0.2527 - accuracy: 0.9429
Epoch 24/30
108/108 [==============================] - 16s 149ms/step - loss: 0.3478 - accuracy: 0.9207
Epoch 25/30
108/108 [==============================] - 16s 150ms/step - loss: 0.2069 - accuracy: 0.9414
Epoch 26/30
108/108 [==============================] - 16s 150ms/step - loss: 0.2176 - accuracy: 0.9460
Epoch 27/30
108/108 [==============================] - 16s 150ms/step - loss: 0.3399 - accuracy: 0.9176
Epoch 28/30
108/108 [==============================] - 16s 150ms/step - loss: 0.2082 - accuracy: 0.9404
Epoch 29/30
108/108 [==============================] - 16s 149ms/step - loss: 0.2558 - accuracy: 0.9417
Epoch 30/30
108/108 [==============================] - 16s 149ms/step - loss: 0.1682 - accuracy: 0.9633
```

Figure 9. Accuracy observation at the end of 30 epochs.

The system's performance was assessed under each scenario to determine its ability to detect and respond to hazardous gas leaks effectively. The simulations yielded very good results, demonstrating the system's capability to accurately detect hazardous gas leaks and promptly alert the user as shown in Figure 10 where a virtual screen is giving away output of detection of gases.



```
90.00x27.00x70
90.00x27.00x70
90.00x27.00x70
90.00x27.00x70
90.00x27.00x70
90.00x27.00x70
90.00x27.00x70
90.00x27.00x70
90.00x27.00x70
```

```
1
1
1
1
1
1
1
1
1
```

Figure 10. Circuit simulation showing air-quality, temperature, humidity and hazardous gas detection (0 denotes 'not detected' and 1 denotes 'detected').

The system successfully detected simulated gas leaks of varying concentrations, ranging from low to high levels, within milliseconds of their occurrence. The response time of the system was found to be highly efficient, with alerts being generated and transmitted to the user within seconds of gas detection. This rapid response time ensures timely notification to the user, enabling them to take immediate action to mitigate the potential risks associated with gas leaks. Furthermore, the system exhibited high accuracy of about 96% in distinguishing between normal environmental fluctuations and hazardous gas leaks, minimizing false alarms and ensuring reliable detection of real safety hazards.

4. Conclusions

The system presented here utilizes an Arduino microcontroller and IoT technology to identify air pollution in the environment, aiming to improve overall air quality. A gas leakage detection system is integrated to provide timely alerts via buzzer and user interface in the form of web application as mentioned earlier. The incorporation of IoT technology enhances the monitoring of various environmental aspects, with a focus on air quality. The basis of the hardware component of the system relies on MQ135 and MQ6 gas sensors for assessing air quality, with Arduino serving as the central component. A Wi-Fi module establishes the connection to the internet, and an LCD screen offers visual output. ThingSpeak is employed for monitoring, analyzing, and displaying data, while the IFTTT application is used for user notifications and alerts. The study also delves into exploring an alternative deep learning convolutional neural network architecture configuration, specifically tailored for distinct maximum receptive fields across audio spectrograms. This approach aims to enhance the design of deep CNNs for acoustic classification tasks and adapt successful CNNs from other domains, particularly image recognition, to acoustic scene classification.

To address the initial data insufficiency, white noise was added to the existing 2000 datasets. Augmentation was then applied, converting audio files to spectrogram files to obtain spectrogram outputs. The model implemented in this research employs a two-dimensional CNN with max-pooling 2D, utilizing a total dataset of 4000 samples. Training was conducted using TensorFlow, utilizing all available data with 30 epochs, resulting in an impressive accuracy of 96%. This technology holds the potential to extend beyond individual devices, enabling the installation of air quality sensors throughout a city. This broader application could facilitate the mapping of air quality and the establishment of a website where individuals can track pollution levels in their respective areas.

The system effectively identified simulated gas leaks across a range of concentrations, from minimal to substantial, within milliseconds of their onset. Its responsiveness proved remarkably swift, issuing alerts to users, mere seconds after gas detection. This rapid reaction guarantees prompt notification to users, empowering them to promptly address potential risks linked to gas leaks. Additionally, the system displayed remarkable precision in discerning between typical environmental variations and dangerous gas leaks, reducing false alarms and ensuring dependable detection of genuine safety threats.

References

1. Y. Xing, Y. Xu, M. Shi, and Y. Lian, "The impact of PM2.5 on the human respiratory system," J. Thorac. Dis., vol. 8, no. 1, pp. E69–E74, Jan. 2016.
2. K. Koutini, H. E. Zadeh, G. Widmer, “Receptive-field-regularized CNN variants for acoustic scene classification,” cited as asXiv:1909.02859 [eess.AS] accepted at DCASE Workshop 2019.
3. H. N. Shah, Z. Khan, A. A. Merchant, M. Moghal, A. Shaikh, P. Rane, “IOT Based Air Pollution Monitoring System,” International Journal of Scientific & Engineering Research Volume 9, Issue 2, February-2018 ISSN 2229-5518.
4. Varma, Prabhakar S., K. Jayavel, “Gas Leakage Detection and Smart Alerting and prediction using IoT,” IEEE 2017 2nd International Conference on Computing and Communications Technologies (ICCCT) 327-333.
5. K. Kumar, Hemanth, Sabbani, “Smart Gas Level Monitoring, Booking & Gas Leakage Detector over IoT,” IEEE 7th International Advance Computing Conference (IACC) 330–332.
6. Y. Lee, W. Hsiao, C. Huang, S. T. Chou, “An integrated cloud-based smart home management system with community hierarchy,” IEEE Transactions on Consumer Electronics, 62(1), 1–9.
7. J. Joshi,V. Rajapriya, S. R. Rahul, P. Kumar, S. Polepally, R. Samineni, D. G. K. Tej, “Performance enhancement and IoT based monitoring for smart home,” IEEE 2017 International Conference on Information Networking (ICOIN) 468–473.
8. G. Roma, W. Nogueira, P. Herrera, “Recurrence quantification analysis features for auditory scene classification,” IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events, 2013.
9. D. Wang, G. Brown, “Computational Auditory Scene Analysis: Principles, Algorithms, and Applications,” Wiley, 2006.
10. Y. Sakashita, M. Aono, “Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divi.” DCASE2018 Challenge, 2018.
11. K. J. Piczak, “ESC: dataset for environmental sound classification,” in Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM ’15, Brisbane, Australia, October 26 - 30, 2015.
12. W. Luo, Y. Li, R. Urtasun, R. Zemel, “Understanding the Effective Receptive Field in Deep Convolutional Neural Networks,” in Advances in Neural Information Processing Systems 29, 2016, pp. 4898–4906.
13. M. Bishop, “Pattern Recognition and Machine Learning,” (Springer, Berlin, 2006).
14. Martins, H., Gupta, N., Reis, M.J.C.S., Ferreira, P.J.S.G. (2022). Low-cost Real-time IoT-Based Air Quality Monitoring and Forecasting. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 442. Springer, Cham. DOI: 10.1007/978-3-031-06371-8_20