

## Research Article

# Hardware Module Design and Software Implementation of Multisensor Fire Detection and Notification System Using Fuzzy Logic and Convolutional Neural Networks (CNNs)

Robert A. Sowah <sup>1</sup>, Kwaku Apeadu,<sup>1</sup> Francis Gatsi,<sup>2</sup> Kwame O. Ampadu,<sup>1</sup> and Baffour S. Mensah<sup>1</sup>

<sup>1</sup>Department of Computer Engineering, University of Ghana, P.O. Box LG 77, Legon, Accra, Ghana

<sup>2</sup>Faculty of Engineering and Computer Science, Ashesi University, Berekuso, Eastern Region, Accra, Ghana

Correspondence should be addressed to Robert A. Sowah; [rasowah@ug.edu.gh](mailto:rasowah@ug.edu.gh)

Received 16 September 2019; Revised 20 November 2019; Accepted 8 January 2020; Published 1 February 2020

Academic Editor: Franca Giannini

Copyright © 2020 Robert A. Sowah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents the design and development of a fuzzy logic-based multisensor fire detection and a web-based notification system with trained convolutional neural networks for both proximity and wide-area fire detection. Until recently, most consumer-grade fire detection systems relied solely on smoke detectors. These offer limited protection due to the type of fire present and the detection technology at use. To solve this problem, we present a multisensor data fusion with convolutional neural network (CNN) fire detection and notification technology. Convolutional Neural Networks are mainstream methods of deep learning due to their ability to perform feature extraction and classification in the same architecture. The system is designed to enable early detection of fire in residential, commercial, and industrial environments by using multiple fire signatures such as flames, smoke, and heat. The incorporation of the convolutional neural networks enables broader coverage of the area of interest, using visuals from surveillance cameras. With access granted to the web-based system, the fire and rescue crew gets notified in real-time with location information. The efficiency of the fire detection and notification system employed by standard fire detectors and the multisensor remote-based notification approach adopted in this paper showed significant improvements with timely fire detection, alerting, and response time for firefighting. The final experimental and performance evaluation results showed that the accuracy rate of CNN was 94% and that of the fuzzy logic unit is 90%.

## 1. Introduction

Fires have a duality such that while they can be useful in cooking and manufacturing; they can also cause loss of lives and destruction of property. With fire and some of its by-products being employed in a lot of useful applications such as cooking, power generation, and manufacturing process, among others, it has the potential also to cause havoc.

According to the Center of Fire Statistics of the International Technical Committee for the Prevention and Extinction of Fire (CTIF), between 70,000 and 80,000 deaths occur annually due to fire outbreaks since the beginning of the 21<sup>st</sup> century [1]. In fact, GHC 1.74M (0.92M USD) was

lost due to fire outbreaks in the first quarter of 2012 in Ghana [2].

Fire safety experts say that when there is a fire outbreak in the home, workplace, or factory, the occupants usually have just about two minutes to escape before it is too late. Fire-related loss of lives and properties continue to increase despite vigorous fire safety campaigns being carried out by the public agencies. In developing countries, typical fire detection relies on humans. In cases of distress, poor town and country planning make it difficult for firefighters to arrive on time. The problem is further compounded by inadequate alert and notification mechanisms. Fires are, however, so destructive that they can annihilate a whole

fortune within hours. Therefore, this situation must be tackled. It is of utmost importance that fire outbreaks be detected and that occupants and the fire service be alerted within a very short time.

There are three main elements required for fire to exist; these are oxygen, heat, and fuel. These make up what is known as the fire triangle. The proportion of each of these elements determines the nature of the fire. Before the introduction of technology, fire detection was done mainly by visual inspection and confirmation. This, however, changed during the late 1930s when Walter Jaeger accidentally discovered a sensor that can detect smoke, thereby paving the way for research into modern smoke sensors. Until recently, most consumer-grade fire detection systems relied solely on smoke detectors. The protection provided by these fire detectors is limited by the type of fire present and the detection technology in use. The need to produce a more reliable fire detection system devoid of false alarms has led to the adoption of multisensor approaches. Unlike fire detection approaches relying solely on smoke, this approach relies on the detection of more than one fire signature. An emerging phenomenon in fire detection is the use of multiple parameters and sensors in the detection mechanism [3, 4]. Some multisensor fire detection techniques being used currently include data fusion algorithms [5, 6]. These techniques allow different fire signatures to be captured and processed together for ultimate fire detection.

Similarly, image processing techniques with neural networks are employed in [7–9] for fire detection. These algorithms use image processing techniques with neural networks to learn from video frames and detects fire based on fire signatures such as smoke and flames. Moreover, Bayesian networks are also employed for fire detection [10].

Deep learning has progressed significantly; neural networks can learn data with multiple levels of abstraction and automatically extract features from data, as evident in the paper published by Tao et al. in [11]. Motivated by the successes of deep learning in the area of computer vision and image processing, a convolutional neural network (CNN) was used to overcome the limitations of a sensor-based system. CNN was used to process video feeds from a surveillance camera and alert in the event of possible fire detection. To account for fire detection in wide areas, a camera surveillance unit was integrated into the system's architecture for ease of processing images of fires over wide areas of the installed units. Most sensor-based systems are generally limited in terms of their proximity to the fires they tend to detect, the closer they are to the fire, the better the detection results; thus vast areas can be quite daunting to detect fires using only sensed fire signatures like flame, smoke, and heat. Consequently, sensors only are not useful in large areas since they require proximity to the signatures of the fire, as stipulated by Frizzi et al. in [12].

In this paper, we present the design and development of a fuzzy logic-based multisensor fire detection system with both real-time web-based and mobile-based alert notification systems with the capability for image processing of fires using CNNs. We adopted the fuzzy logic approach to fire detection because it simplifies the fusion of data from the

multiple sensors, thereby making it easier to analyse the often-noisy sensor data. Fuzzy logic also best approximates the actual values of the various fire signatures observed for accurate and efficient detection.

This paper is organized into sections. Section 1 provides the introduction and necessary background for multisensor fire detection using fuzzy logic and CNNs. It presents the background and the motivation for the research work and the problem statement. Section 2 presents the literature review on fire detection systems. It highlights the various techniques for fire detection based on the use of sensors for sensing fire signatures and video sequence processing for fires. It identifies the strengths and weaknesses of such systems and proposes a novel method using fuzzy logic and CNNs. Section 3 provides the novelty of the system design architecture and methodology, including the concepts, algorithms, and flowcharts. It highlights the various modules with their design and procedures for implementation. Section 4 focuses on the actual implementations with experimental verification and corresponding integration of the developed modules. This section provides sufficient details on the testing and performance evaluation of the proposed system. Finally, Section 5 provides conclusions on the system design and methodology and makes relevant recommendations for future enhancements.

## 2. Literature Review

Sowah et al. [13–16] proposed the design and implementation of a fuzzy logic control system with feedback over an Arduino microcontroller system for fire detection for both homes and automobiles. The automatic system consisting of flame sensors, temperature sensors, smoke sensors, and a re-engineered mobile carbon dioxide air-conditioning unit was tested on a medium-sized physical car. Results show that the automobile fire detection and control system devoid of false alarms detects and extinguishes the fire under 20 seconds. An innovative, very promising solution module for hardware implementation in fire detection and control for automobiles has been developed by using new algorithms and fuzzy logic. The data from the sensors are fused to enable accurate detection of fire based on fire signatures.

There has been an increase in the number of publications dealing with video processing to detect fire, particularly with the usage of convolutional neural networks. Frizzi et al. [12] proposed a convolutional neural network for video fire and smoke detection. They implemented a neural network with nine layers. To detect fire and smoke in a video frame, a sliding window of size  $12 \times 12$  is applied to the last feature map generated from the network. The authors chose to use a small network to increase the classification time. The proposed system was, however, designed to detect red-fire and had some challenges with smoke detection.

Muhammad et al. [17, 18] proposed a cost-effective fire detection CNN architecture for surveillance videos with less computational time and memory footprints. Their model is inspired by GoogLeNet architecture, considering its reasonable computational complexity and suitability for the intended problem compared with other computationally

expensive networks such as AlexNet. To balance the efficiency and accuracy, the model is fine-tuned considering the nature of the target problem and fire data. Experimental results on benchmark fire datasets reveal the effectiveness of the proposed framework and validate its suitability for fire detection in closed circuit television (CCTV) surveillance systems compared with state-of-the-art methods.

In the paper by Wang et al. in [19], they proposed a novel algorithm based on a CNN and support vector machines (SVMs) for fire detection in infrared (IR) video surveillance. To improve the performance of IR fire detection, they develop a 9-layer convolutional neural network named IRCNN instead of traditional empirically handcrafted methods to extract IR image features. They utilized a linear support vector machine for training with extracted features to detect fire. Experimental results show that their approach achieved both high precision (98.82%) and high recall (98.58%) on infrared (IR) flame dataset and real-time detection.

Different fire signatures such a flame, smoke, and heat were used for fire and smoke detection using CNN by different researchers [20–26]. Some authors extended their work to include the detection of forest fires and enable fast response time for firefighting and the performance of rescue operations. Moreover, indoor fire detection using gas chemical sensing provides faster fire alarm responses than conventional smoke-based fire detectors. Gas-based fire detection could provide an additional level of safety to building occupants in the event of fire [27]. But, these detectors rely heavily on one or two of the three fire signatures, which are inadequate for effective fire detection and firefighting.

A real-time fire detection method combining AdaBoost, local binary pattern (LBP), and a convolutional neural network in video sequence was produced by Maksymiv et al. in [28]. The proposed framework for emergency detection consisted of two main parts; the first part generates regions or areas where flame or smoke may be present in the video stream. This operation is performed by the AdaBoost and LBP combined. Then a final classification is done by CNN. The generation of the possible regions reduces the computational time of CNN significantly but is prone to a high rate of false positives.

This research paper adopts the strengths of the real-time fire detection methods reviewed using convolutional neural networks on video sequence frames and integrates that with proximity detection of fires using sensors that fuse all the three fire signatures for early detection and alerts notification while providing navigational aid to the scene of fire for the fire rescue team to respond appropriately. This novelty in the choice of using both scalar and vision sensors overcomes the inherent problems highlighted in the review of fire detection systems.

### 3. System Design and Development

#### 3.1. System Overview

**3.1.1. Sensor-Based Fire Detection Unit.** The general overview of the hardware module design and software implementation of the multisensor fire detection system is shown

in Figure 1. The hardware components of the sensor-based fire detection unit, as shown in Figure 2, is a device meant to be installed in some premises similarly as a conventional smoke detector. At a high level, it comprises sensors and a global system for mobile communication (GSM) module connected to a microprocessor that runs the fuzzy logic fire detection program. The microprocessor polls the sensors at a regular interval and runs the inputs through the fuzzy logic application. If it concludes that a fire has been detected, a fire alert message is sent out through the web portal and the mobile app through the management information systems (MISs) to the occupants of the premises and the nearest fire station. If sending the message over the data connection is unsuccessful, then it sends the message out via short message service (SMS). The fire detection unit comprises the physical components, including the sensors and the microprocessor board, and the software that embodies the fuzzy logic fire detection algorithm and essentially drives the system.

The software subsystem is that nonphysical part of the fire detection unit, which is concerned with reading inputs from the sensors, determining whether the readings are indicative of a fire or not, and raising alerts in cases of fires. The sensors used, respectively, in [28–31] were chosen based on detection range, size, and cost. By adopting a multisensor approach, the need to fuse the data arises. Fuzzy logic provides an easy way of dealing with uncertain data from multiple sensors by aggregating these to make a decision. Fuzzy logic is a type of logic that tries to mimic the human brain by incorporating the imprecision with which humans make decisions. As a practical example, if a human is tasked to describe the temperature of a cup of beverage, he or she would typically just say that it is hot, cold, cool, warm, not too hot, etc. rather than stating that it is 25°C or 52°C as would a computer. Fuzzy logic is therefore based on imprecision, and it uses linguistic variables, defined as fuzzy sets, to codify common sense and hence approximate human reasoning. In this regard, each of the three fire signatures (variables) three states (sets) are defined and described by membership functions. The output is also defined as three possible states (sets): no fire, potential fire, and fire. Fuzzy rules describing combinations of the various sets which indicate a fire or no fire or a potential fire are coded into the system. Then as the system operates, it takes the raw readings from the sensors and fuzzifies and defuzzifies them to give an output. Thus, rather than saying there is  $x$  units of fire, it will say there is a fire, or there is a potential fire, or there is no fire. The basic configuration of the fuzzy logic unit of the fire detection system is shown in Figure 3.

The size of the rule base of a fuzzy logic system is affected by its inputs and sets used. A reduced number of fuzzy sets results in a reduced rule base and vice versa. For each sensor input (linguistic variable), three fuzzy sets were created using grade, reverse grade, and triangular membership functions. To determine these membership functions, data collected from each sensor during tests was used to tune the membership of the various fuzzy sets. We used the center of area (CoA) defuzzification method to obtain the system output, which gives the likelihood of a fire. Figure 3 shows a basic configuration of a typical fuzzy logic system. Figure 4 shows

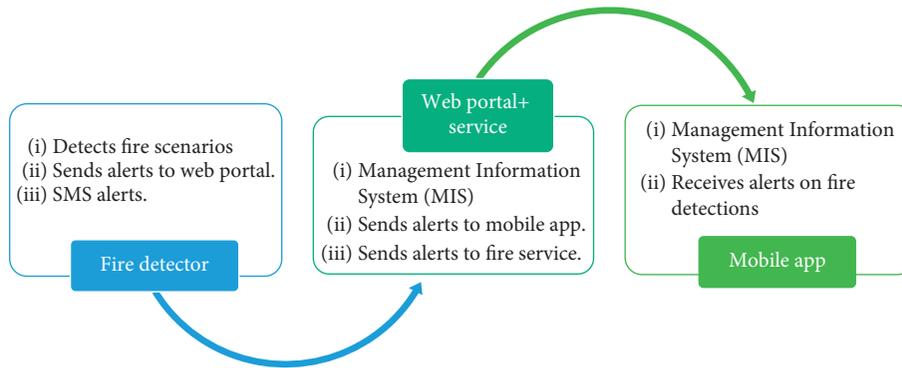


FIGURE 1: General overview of the system.

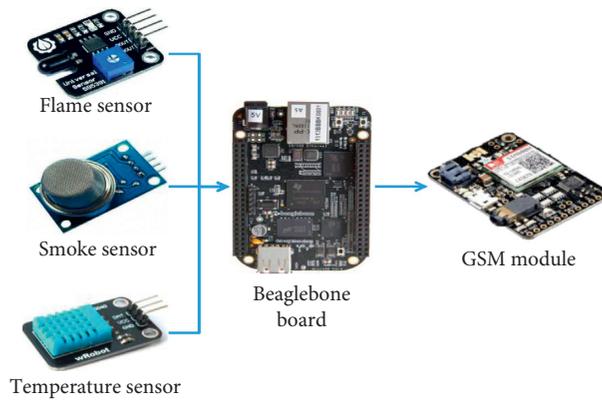


FIGURE 2: Sensor-based detection unit of the system.

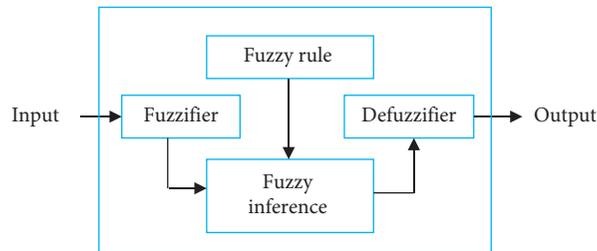


FIGURE 3: Basic configuration of a fuzzy logic system.

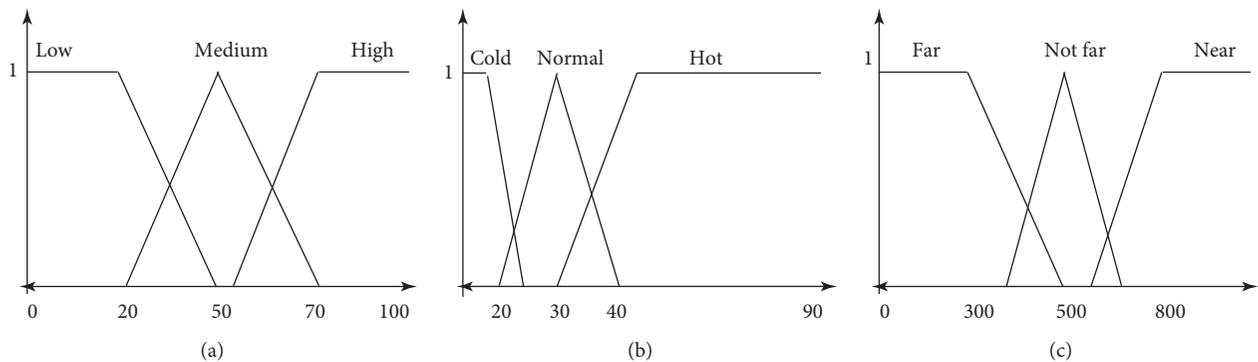


FIGURE 4: Fuzzy membership sets. (a) Smoke density. (b) Temperature. (c) Flame.

the membership functions for each of these sets. The ambient smoke density data from the smoke sensor were grouped into three fuzzy sets (low, medium, and high). The temperature sensor readings have been classified into cold, normal, and hot fuzzy sets. The fuzzy rules for fire status are shown in Tables 1–3 for different scenarios of smoke levels.

When the three sensor inputs are supplied to the system, the inputs are then fuzzified using the various membership sets. The output of this process gives the fire status where a fire alert may be activated.

Based on the output of the system, a fire event is either dispatched to the web-based notification system, and house owner(s) or no action is taken. The flow diagram for the fire detection system is shown in Figure 5.

**3.1.2. Convolutional Neural Networks.** Convolutional neural networks are multilayer artificial neural networks designed to handle two-dimensional input data like images [32]. It is inspired by the mode of operation found in the visual perception of living creatures [18].

Figure 6 shows the structure of CNN. The layers of the network are made up of multiple two-dimensional planes. Each 2-D planes consists of several neurons that make CNNs suitable for handling image data.

The presence of a convolutional layer and a pooling layer in the architecture enables easy extraction of local features and reduction of complexity as stipulated by Pei in [33].

The Inception structure in GoogLeNet [34] can extract more relevant features with less computational effort by establishing and exploiting the sparse structure, thus improving the accuracy of the network. GoogLeNet has 22 layers, deeper than AlexNet’s 12 layers, but the parameters are only one-twelfth of AlexNet’s, and the accuracy is higher. The basic structure of GoogLeNet is the Inception module, which is shown in Figure 7. The Inception module has two main features, namely, (1) reunion of convolution cores with multiple sizes and (2) dimension conversion using  $1 \times 1$  convolution [34]. Experiments show that, although dimensionality reduction is first carried out, as long as the output dimension remains unchanged, the process of dimensionality reduction in the middle will not affect the accuracy of the network. Inception V3 further decomposes a larger two-dimensional convolution into two smaller one-dimensional convolutions. That means  $1 \times N$  convolution followed by  $N \times 1$  convolution can replace  $N \times N$  convolution. The set  $n = 7$  is used to handle feature maps of  $17 \times 17$  image size [34].

In the convolution operation, feature maps are generated by applying kernels of different sizes to the input data. A pooling operation is performed on the feature maps with maximum activations from small neighbourhood in the feature maps, as published by Muhammed et al. [18, 36]. The pooling operation handles the computational overhead of passing the features extracted directly to the classifier, as proposed by Aloysius and Geetha in [32]. This approach using pooling operation is essential for images with high resolution and large sizes to be handled in a computationally

TABLE 1: Rules for fire status when the smoke is low.

		Flame intensity		
		Far	Not far	Near
Ambient temperature	Cold	No fire	No fire	Potential fire
	Normal	No fire	No fire	Potential fire
	Hot	Potential fire	Fire	Fire

TABLE 2: Rules for fire status when smoke is medium.

		Flame intensity		
		Far	Not far	Near
Ambient temperature	Cold	No fire	Potential fire	Fire
	Normal	Potential fire	Fire	Fire
	Hot	Potential fire	Fire	Fire

TABLE 3: Rules for fire status when smoke is high.

		Flame intensity		
		Far	Not far	Near
Ambient temperature	Cold	Potential fire	Potential fire	Fire
	Normal	Potential fire	Potential fire	Fire
	Hot	Fire	Fire	Fire

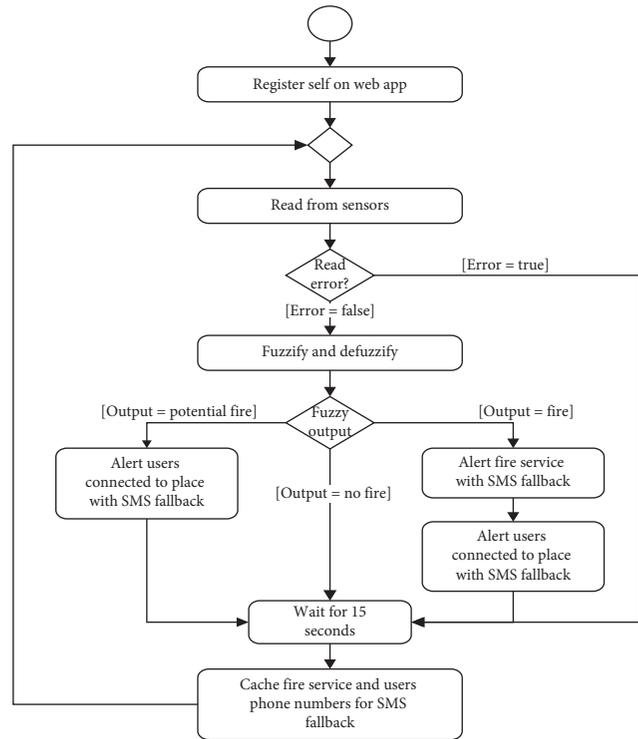


FIGURE 5: Flow diagram for fire detection software.

efficient manner. For the intended classification task, the Inception-V3 module was used. Inception-V3 is the 3<sup>rd</sup> version in a series of Google’s deep learning convolutional architectures. It was trained using a dataset of 1,000 classes from the original ImageNet dataset, which was trained with over 1 million training images.

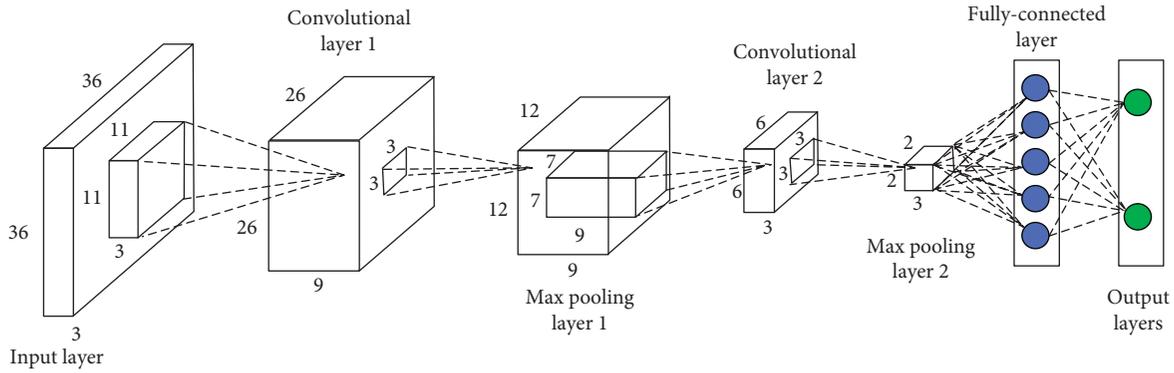


FIGURE 6: Architecture of CNN.

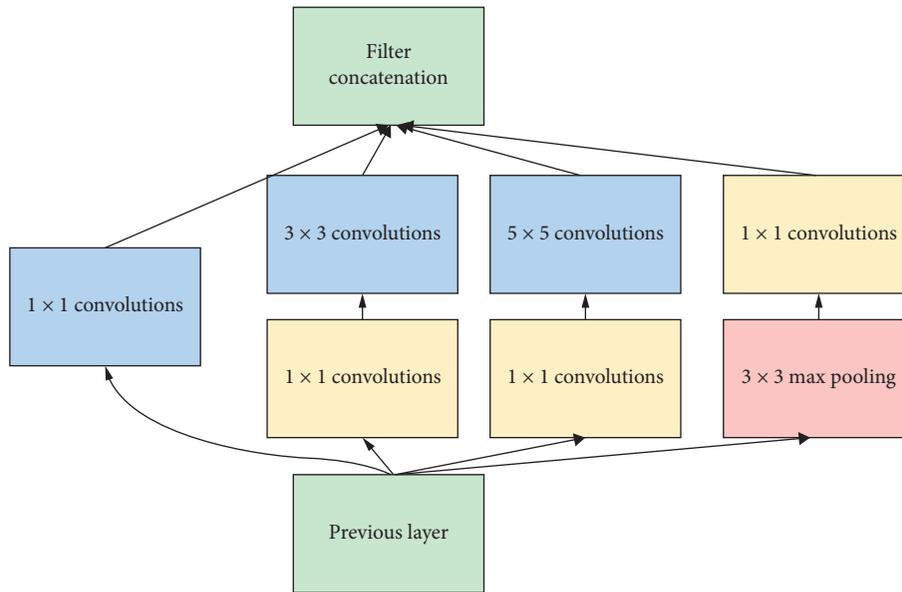


FIGURE 7: Inception module with dimension reduction. Source: adapted from [34, 35].

In traditional classification learning, to ensure the accuracy and reliability of the training classification model, there are two fundamental assumptions, namely, (1) the training samples for learning and the new test samples must satisfy the same distribution independently and (2) sufficient training samples must be available. However, as time goes on, previously available labelled sample data may become unavailable. Also, labelled sample data are often scarce and difficult to obtain.

Transfer learning is a new machine learning method that uses existing knowledge to solve problems in different but related fields [37]. It relaxes two underlying assumptions in traditional machine learning, aimed at migrating existing knowledge to solve learning problems in target areas where only a small amount of labelled sample data are available. The more the factors shared by two different fields, the easier it is to transfer learning. Transfer learning can be divided into three main categories: firstly, case-based transfer learning; secondly, feature-based transfer learning in isomorphic space; thirdly, transfer learning in heterogeneous space [37]. Therefore, through transfer learning, relevant

instances, or features in massive data sets can be migrated to micro data sets, thus improving the generalization ability and recognition accuracy of the model.

With the transfer learning technique which allows an existing model to be retrained, resulting in a significant reduction in the training time, we retrained the Inception modules with over 1000 images of both instances fires and nonfire situations. The architecture of the Inception model is shown in Figure 7.

The Inception module performs multiple convolutions on given input and decides which one is best for the image. This was necessary to handle the issue of varying resolutions of images and enable efficient classifier computations.

**3.1.3. Web Application.** The web portal provides an interface for general users, manufacturers, and the fire service, while the web service serves as a link between the fire detector and the mobile app. It is the backing store for the entire system. It performs the same functions as the mobile application and informs firefighters nearby the detected fires and enables the

manufacturers to track production units. The use case diagram for the web application is shown in Figure 8.

**3.1.4. Mobile Application.** As our mobile phones are handy devices, it is expedient to have a mobile app to complement conventional siren. This way, in the cases of a fire incident, users (occupants and owners of the place on fire) can be notified wherever they are. After a person acquires and installs the fire detection unit in an area of his/her choice, he/she can mark that place within the app by providing a name and marking its location on a map. The user can then add the fire detection units installed to that place hence creating an association between the fire detection units. Then that user may add other users to that place by entering their e-mail addresses. The use case diagram for mobile application development is shown in Figure 9.

**3.2. System Architecture.** The system architecture consists of both hardware and software components. The hardware components comprise of the multisensor fire detection system, which will have the installation of the software components when the user purchases the unit and installs it at residential, commercial, or industrial premises. The user will configure the system based on the number of units purchased. For enhanced fire detection, a surveillance camera unit is incorporated as part of the hardware system implementation, which will continuously monitor the premises and send the video feed to a centralized server for fire incident detection and alert notification. In the case of fire detection, the owner of the property will be alerted and also an alert will be sent to the fire service for the fire and rescue crew with locational mapping and navigational instructions using Google Map API. The overall system architecture is depicted in Figure 10. Based on the development and testing of the multisensor fire detection, the hardware components were configured according to the diagram shown in Figure 11, and the actual hardware implementation for fire detection is also given in Figure 12. In order to enhance fire detection over a wider area that has a monitoring camera installation, we employ convolutional neural networks for efficient video processing during the testing phase of the development. This architecture, with the incorporated video processing capability, is depicted in Figures 10 and 12.

In the development of the video processing unit of the fire detection system, we considered the processing speed of the entire system. That is because, in the event of a fire outbreak, an alert of the fire must be delivered on time for necessary measures to be taken. Since the data stream being fed to the convolutional neural network is primarily a video stream, there was the need for a high processing power to run the recognition model on the data feed. Due to this need, the option of running the recognition module on the Beaglebone microprocessor unit was not an efficient option to consider due to its memory capacity limitation. The prospect of incorporating a dedicated server to run all video processing was therefore employed. Hence, the video stream from the surveillance camera is sent to the server where the

CNN model is run on the data, and if a fire is detected, the alerting system is thereby triggered.

## 4. System Implementation and Testing

The convolutional neural network was trained on over 1000 images of instances of fire and nonfire situations. The data were collected from online image data sources. Sample images used for the convolutional neural networks are shown in Figure 13. The model was trained through 800 steps with a batch size of 100 and the 0.01 default initial learning rate. As a principle for the training of neural networks, the dataset was split into training and testing sets. The module was trained on Google Colab, Google's free cloud service for Artificial Intelligence (AI) developers. Colab helps develop deep learning applications on a graphics processing unit (GPU) for free. The GPU specifications are 12 GB GDDR5 VRAM having 2496 CUDA cores [38]. To get an inference from the model, test video streams were used as input and passed through to the system. The classifier outputs probabilities for the two classes: "fire" and "no fire." The class with the maximum probability score is considered as the result of the classifier. The classifier module was implemented with Google's TensorFlow. TensorFlow is an open-source software library provided by Google for numerical computation using data flow graphs. [35, 39, 40]. After the training the network on the data, a classifier with an accuracy of 94% was achieved. The classification process is as follows; the video feed is preprocessed, and frames are extracted from the video. The extracted frames/images are classified to determine the condition of the area.

The system hardware specification for the different sensors and system components are shown in Table 4.

**4.1. System Testing and Results.** Given the requirements of the various divisions of the system, they were tested to verify that they met those requirements. For the sensor-based fire detection unit, tests were run in the lab using strong fumes from the methylated spirit to trigger the smoke sensor, a flame from the matchstick triggered the flame sensor and the temperature sensor. The readings that were taken from the sensors were logged to files and inspected to see if they reflected the presence and absence of the various fire signatures. Logs were also kept for all requests made to the web service so that we could ascertain whether the fire detection unit was able to send the alert when a fire was detected. Occasionally, we also disabled the Internet connection on the fire detector to verify if it sent SMS alerts under such circumstances. Field tests were conducted in the lobby outside the lab and in the corridor at the School of Engineering Sciences, University of Ghana, Legon, with fire from burning paper. The results of the mobile application development and test runs, as well as web application interface, are shown in Figures 14(a)–14(e). The Google API enabled navigation to the location of the detected fire. It gives fire service personnel quick response time as it gives them the time of day navigation to avoid heavy traffic jams along the path to the location.

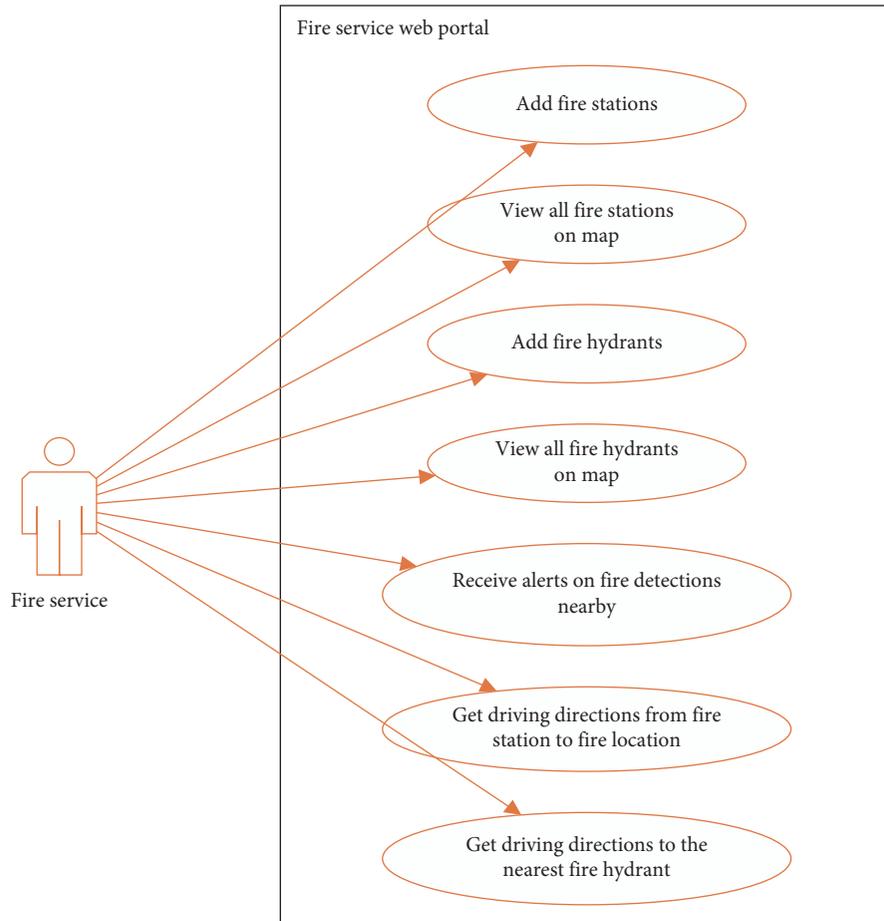


FIGURE 8: Web application use case diagrams.

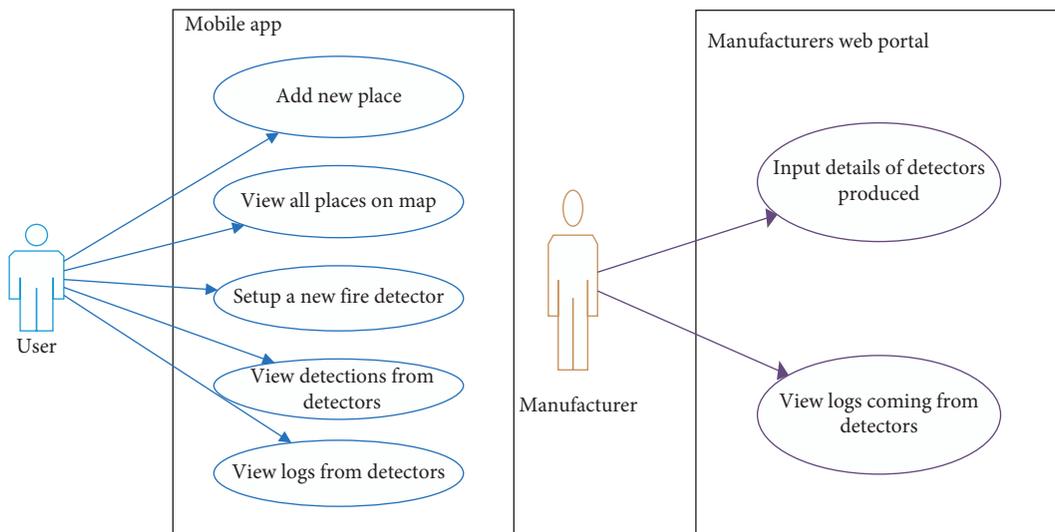


FIGURE 9: Mobile application use case diagrams.

The field tests performed for the different components and scenarios, and the corresponding results are given in Table 5.

The video classification task for the system's use case is a basic binary classification task. The system indicates whether

there is an incidence of a fire or otherwise; the classes for classification are "fire" and "no fire." Figures 15(a) and 15(b) show the results obtained from test instances of fire and no fire situations, respectively. The test cases used considered the occurrence of fire both indoors and outdoors from the

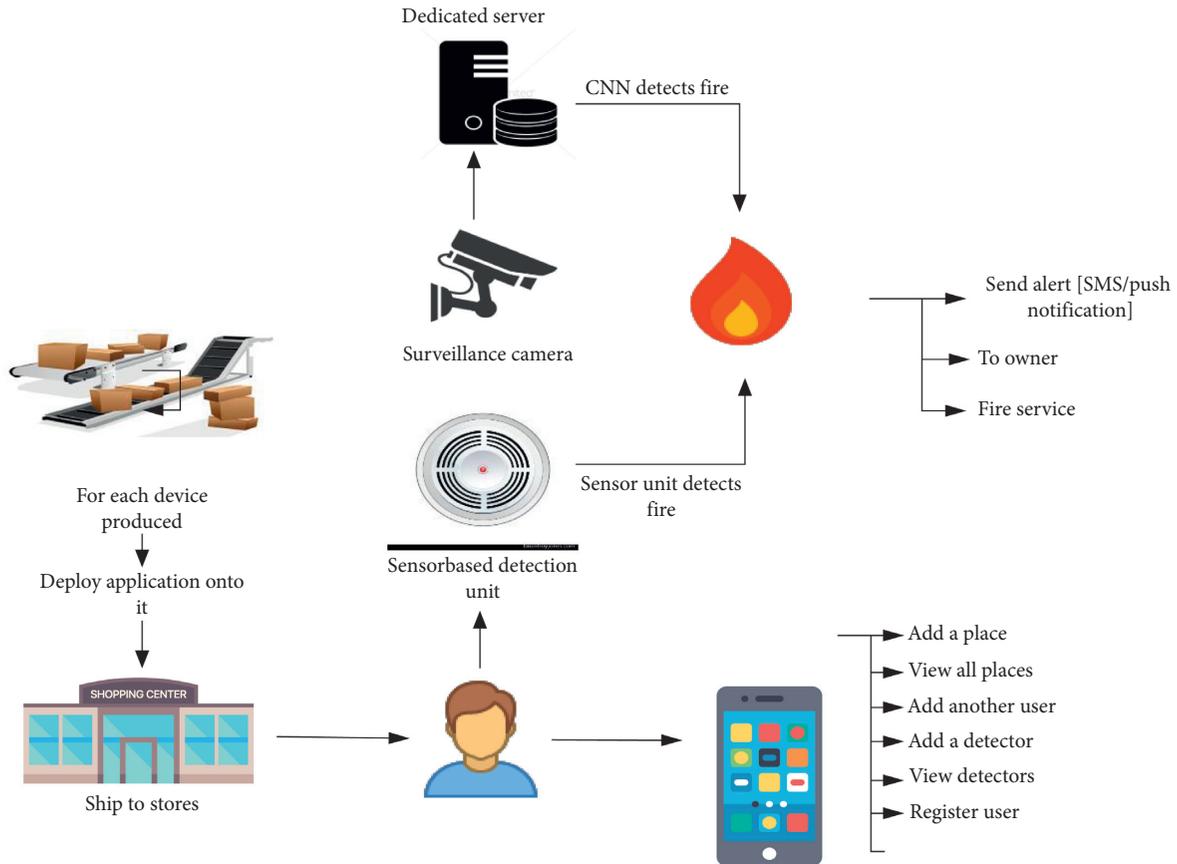


FIGURE 10: System architecture.

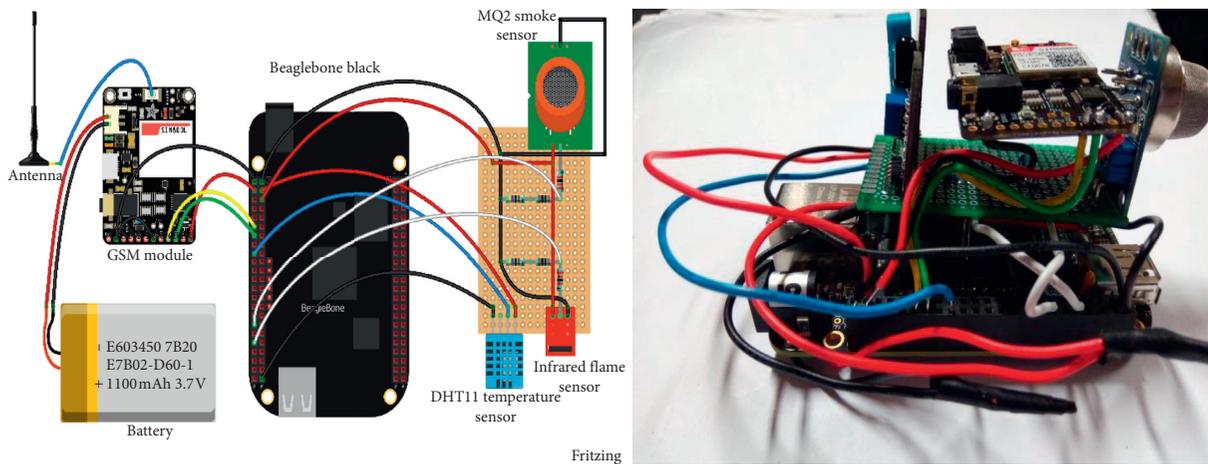


FIGURE 11: Sensor-based detection hardware unit.

surveillance camera. Table 6 shows the results of the test cases with corresponding accuracy values obtained.

**4.2. Results and Discussion.** The Beaglebone microprocessor was central to the system since it performs all the processing functions. The smoke and flame sensors, being analog devices, rely on the ADC on-board the microprocessor to perform the conversion. The ADC on-board produces

outputs from 0 to 4096. The sensor units' tests conducted were intended to ensure the accuracy of each sensor. From the result obtained from the temperature sensor, we observed an error margin of  $\pm 2^{\circ}\text{C}$  for each reading. For the smoke sensor, a higher value indicates high smoke obscuration and vice versa. With three output states of the fuzzy logic algorithm, fire alerts only get activated when either the potential fire state or fire state pertinence is high. Unlike a single signature-based detection system which operates by

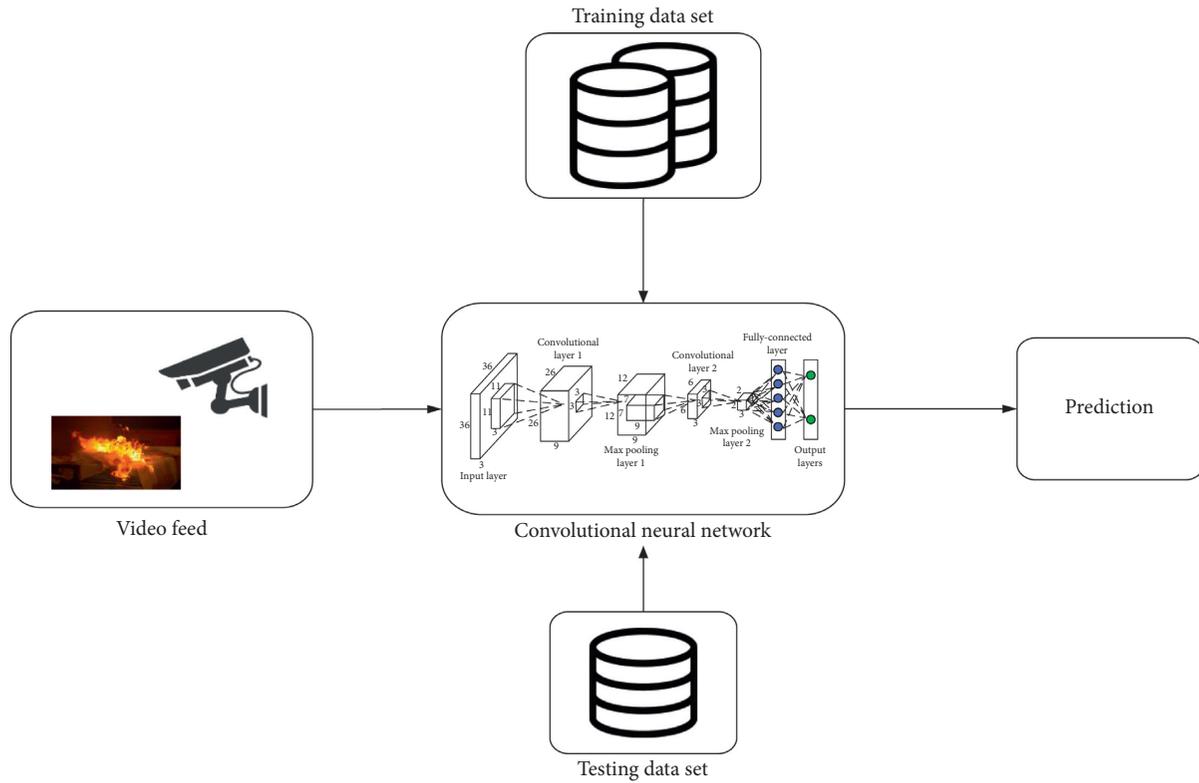


FIGURE 12: Video processing unit on a dedicated server.

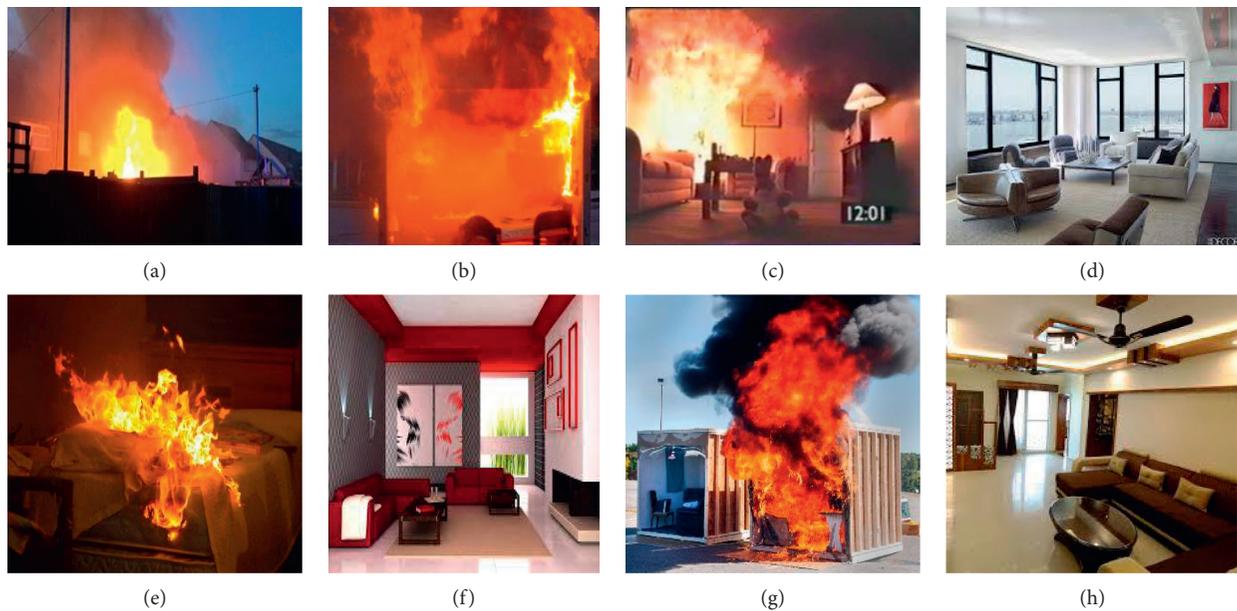


FIGURE 13: Sample images for classes.

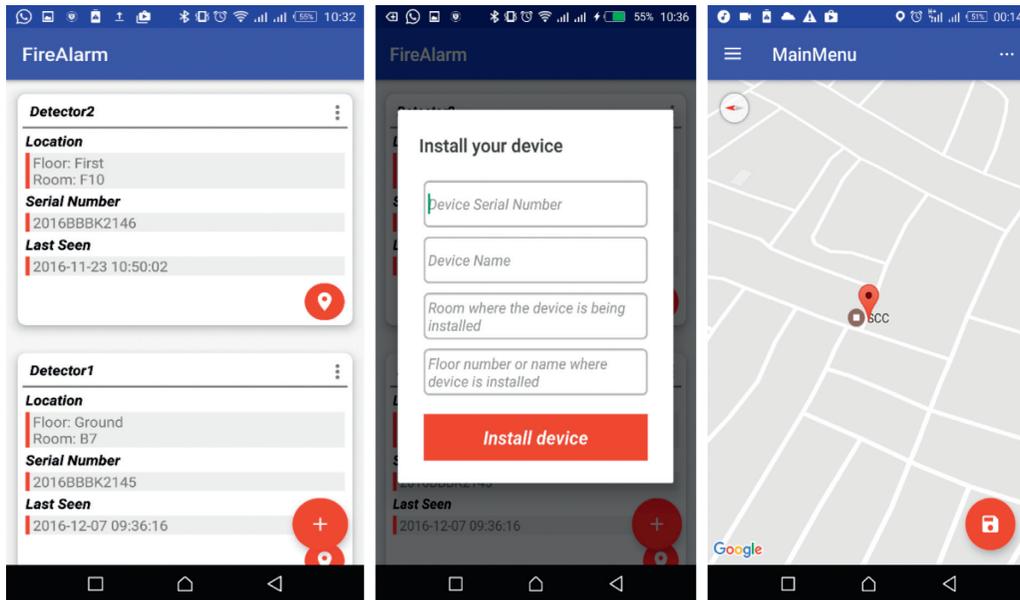
comparing the sensor output to a set threshold, our multi-sensor approach decides based on the weight of each input. This implies that the system can operate without necessarily relying on all signatures. This means the system can detect a flaming fire that produces flame and heat with little or no smoke. This property of the system allows for potential fire

warning to be issued. Sample simulation results from the fuzzy logic algorithm are shown in Table 5. The temperature values are given in degrees.

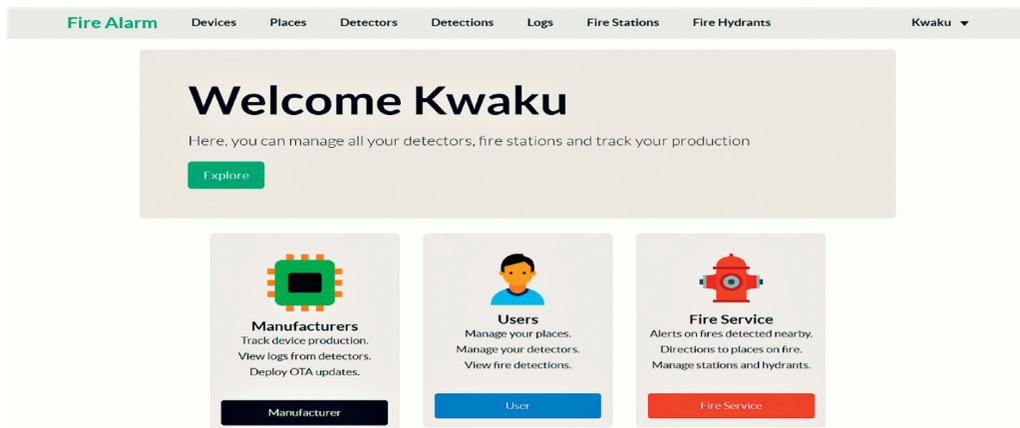
The video classifier performed very well on the tests run on the classifier module. To avoid the instances of false alarms being triggered, a threshold for the classifier

TABLE 4: Hardware specification.

Component	Specifications
Temperature sensor	DHT11 temperature sensor (digital)
Flame sensor	Infrared flame sensor (analog)
Smoke sensor	MQ2 gas sensor (analog)
Microprocessor	Beaglebone black rev. C
Power	5 V 2 A 5.5 mm * 2.1 mm barrel power adapter
GSM module	SIMCOMM SIM800H-based Adafruit FONA 2 G GSM breakout board
uFL antenna	Sticker type 2 dBi uFL antenna
Li-poly battery	1200 mAH 3.7 V rechargeable battery
Perforated board	4 cm x 6 cm perforated board
Resistors	22 kΩ, 10 kΩ and 4.7 kΩ 1% resistors
Surveillance camera	5 megapixel camera with accessories



(a)



(b)

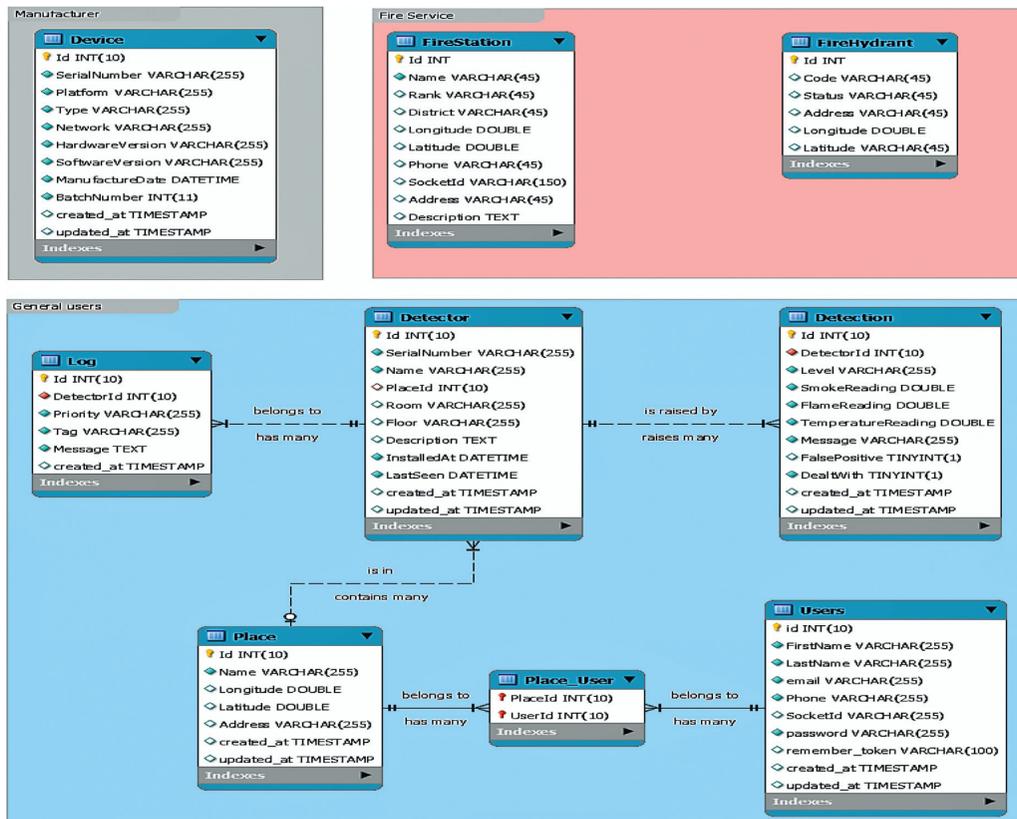
FIGURE 14: Continued.

## Detectors

Name	Serial Number	Location	Last Seen
Detector2	2016BBBK2146	F10 First floor	2016-11-23
Demo Detector	1516BBBK0E33	Kitchen Ground floor	2017-01-12
Detector2	2016BBBK2147	Kitchen Ground floor	2016-11-23
New Detector	2016BBBK2149	B9 Ground floor	2016-12-13
Detector1	2016BBBK2145	Huawei Lab Ground floor	2016-12-07
ORID Test Detector	2016BBBK2528	Huawei Lab Ground floor	2016-12-30

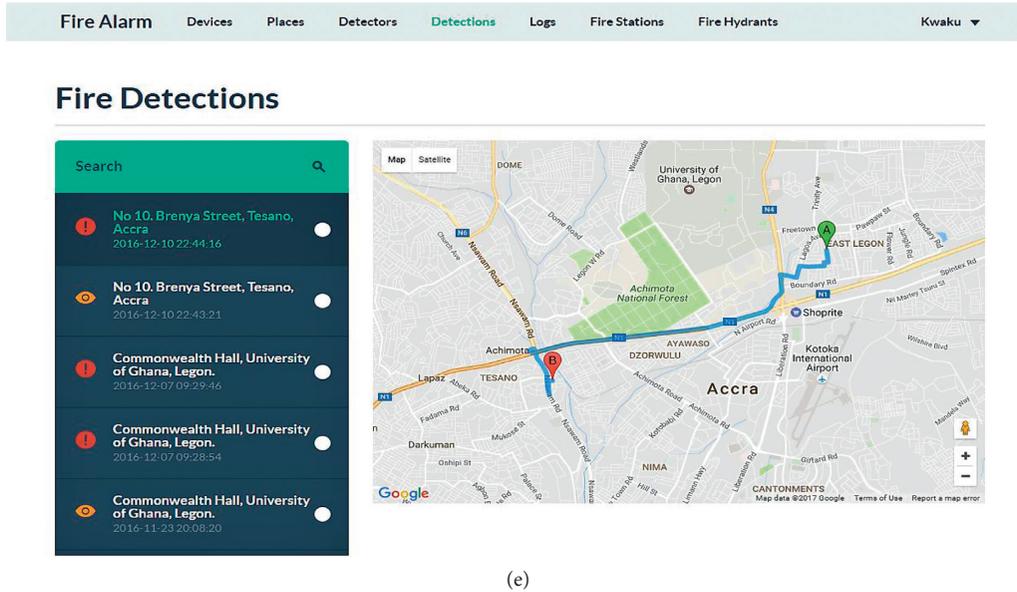


(c)



(d)

FIGURE 14: Continued.



(e)

FIGURE 14: (a) Mobile application screenshots. (b) Web application home page. (c) Detector inventory. (d) Entity-relationship diagram. (e) Web application map navigation.

TABLE 5: Field test results.

Test	Expected result	Remarks
Power on fire detection unit	The fire detection unit boots automatically sets its date and time, tells the web service that it is online, and starts running the fuzzy logic detection algorithm	It was successful most of the time Occasionally it failed to automatically update the date and time; a fix for that is soon to be tested
Leave the fire detection unit running under normal environmental conditions for three days	The fire detection unit keeps running the main detection program and keeps logs of readings from sensors	The system kept running; one of the units, however, overheated; this was attributed to the lack of a proper cooling system on the microprocessor board
Expose the fire detection unit to fire	The readings from the various sensors vary with the intensity of the flames, temperature, and smoke; the fuzzy logic algorithm classifies the inputs as a definite fire and sends an alert	For all the tests conducted, the system was able to detect the fire; beyond a 1-meter radius; however, the detected fire level had to be significant before the system could detect it
Expose the fire detection unit to very high levels of only one or two fire signatures	The fuzzy logic algorithm classifies the input as a potential fire	Successful
Send alert to web service on the detection of fire	The fire detection unit makes a request to the web service to alert all concerned users and the fire service about the detected fire	Successful
Send alert via SMS on detection of fire and absence of Internet connection	The fire detection unit sends the fire detection alert via SMS to all the concerned users and the fire service	Successful

Video output for classification tasks.

confidence was set. Hence, the alarm is only triggered when the confidence is greater or equal to the threshold. The aim is to detect a fire from the video stream with very high accuracy and trigger an alert as quickly as possible. To boost the speed of the classifier, TensorFlow’s “optimize\_for\_inference” script was used to remove all unnecessary nodes in the module. The script also does a few other optimization processes like normalizing operations

into the convolutional weights that help speed up the model. Figure 16 shows a plot of the cross entropy against the number of steps.

The cross entropy is a statistical measure of the error between computed outputs and the desired target outputs of the training data [41]. From the graph, the cross entropy reduces as the step size increases, an indication of a functional and suitable classification module. The video

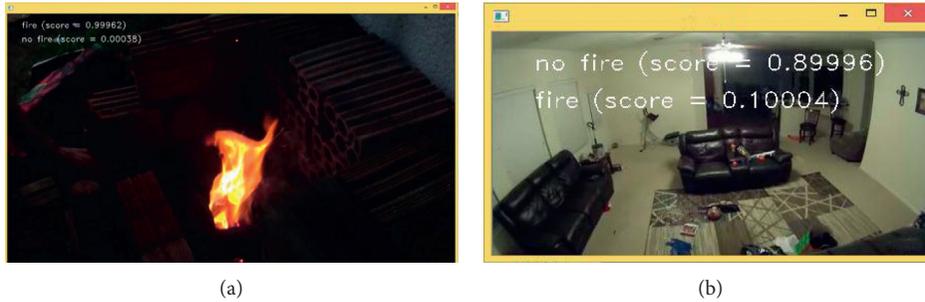


FIGURE 15: (a, b) Screenshots from the video classifier test.

TABLE 6: Video output results for the fire detection and classification.

Scenario	Class detected	Accuracy
Indoor (no fire)	No fire	0.89
Indoor (fire)	Fire	0.91
Outdoor (no fire)	No fire	0.90
Outdoor (fire)	Fire	0.99

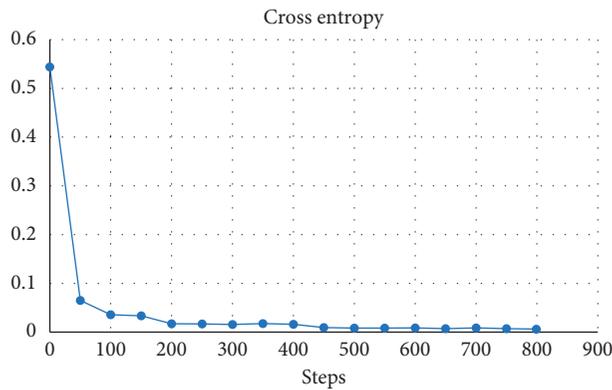


FIGURE 16: Graph of cross entropy against step size.

TABLE 7: Performance evaluation of subsystem modules.

Subsystem	Performance remarks
Microcontroller unit	100% The microcontroller works all the time
GSM module unit	80% GSM mobile is not always reliable due to network problems and power requirement issues
Sensor unit	80% Sensor data spikes sometimes lead to unreliable data from the sensors
Convolutional neural network	94% The CNN unit works as expected to provide high accuracy levels for detected fires from video feeds
Fuzzy logic system	90% Due to test setup constraints, the extensive fire test could not be conducted, hence the incorporation of the surveillance camera hardware system for broader area coverage
Fire notification subsystem	100% The system shows fire notifications in real-time

TABLE 8: Performance evaluation of fire detection.

Smoke	Temperature	Flame	Status
647.1	83.31	0.00	Fire (0.708)
338.92	25.00	844.30	Fire (0.602)
548.5	28.93	918.49	Potential fire (0.459)

classifier also supported different video resolutions since the Inception module performs multiple convolutions on input and decides which kernel size suits the image best.

Based on the performance evaluation and testing, all the subsystem modules of the developed hardware and software system were tested, and the results are recorded and given in Tables 7 and 8. The magnitude of the sensed parameters (such as smoke, temperature, and flame) received from the sensors used, enabling easy fire detection as they offer greater flexibility in handling imprecision and nonlinearities.

## 5. Conclusion and Recommendation

A multisensor fire detection system using fuzzy logic has been designed, developed, and deployed successfully. The fuzzy logic detection unit was augmented by a convolutional neural network to classify video stream to detect and alert homeowners of the occurrence of a fire. Unlike the existing fire detectors, this system employs a multisensor approach whereby the outputs of three sensors sensing three different fire signature parameters (smoke, flame, and temperature) contribute to the fire alert decision and hence produce a more reliable fire detection system devoid of false alarms. Experimental results obtained by introducing strong fumes from methylated spirit to trigger the smoke sensor, a flame from the matchstick triggered the flame sensor, and the temperature sensor to the device shows that the system can effectively enhance the reliability of a fire detection system. When a fire alert is detected, the system automatically reports them to the fire and rescue service and also notifies the owner(s) on their cell phones. The system also provides the fire and rescue service with a map-assisted navigation system to help locate the scene of a fire outbreak which is very helpful when there are poor house and street addressing system in a neighbourhood. There is, however, a need to design a good packaging for the fire detection unit. The package must be able to accommodate the entire unit while allowing light, air, and smoke to reach the sensor. A possible improvement to the developed system is the incorporation of neuro-fuzzy to substantially improve the detection rate and more images to train the network for higher accuracy.

## Data Availability

The data used for training the convolutional neural network are available online.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] World fire statistics CTIF—international association of fire services for safer citizens through skilled firefighters, 2019, <https://www.ctif.org/world-fire-statistics>.
- [2] Fire outbreaks cost Ghana over GHC1 million in 2012 general news 2012 <https://www.ghanaweb.com/GhanaHomePage/NewsArchive/Fire-outbreaks-cost-Ghana-over-GHC1-million-in-2012-253460#>.
- [3] S.-J. Chen, D. C. Hovde, K. A. Peterson, and A. W. Marshall, "Fire detection using smoke and gas sensors," *Fire Safety Journal*, vol. 42, no. 8, pp. 507–515, 2007.
- [4] Z. Tang, W. Shuai, and L. Jun, "Remote alarm monitor system based on GSM and ARM," *Procedia Engineering*, vol. 15, pp. 65–69, 2011.
- [5] R. C. Luo and K. L. Su, "Autonomous fire-detection system using adaptive sensory fusion for intelligent security robot," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 3, pp. 274–281, 2007.
- [6] K. L. Su, "Automatic fire detection system using adaptive fusion algorithm for fire fighting robot," in *Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics*, pp. 966–971, Taipei, Taiwan, October 2006.
- [7] C. Cheng, F. Sun, and X. Zhou, "One fire detection method using neural networks," *Tsinghua Science and Technology*, vol. 16, no. 1, pp. 31–35, 2011.
- [8] C.-y. Lian, "Design of intelligent fire alarm system based on GSM network," in *Proceedings of 2011 International Conference on Electronics and Optoelectronics*, vol. 1, no. 1, pp. 393–396, Dalian, China, July 2011.
- [9] T. Ono, H. Ishii, K. Kawamura et al., "Application of neural network to analyses of CCD colour TV-camera image for the detection of car fires in expressway tunnels," *Fire Safety Journal*, vol. 41, no. 4, pp. 279–284, 2006.
- [10] C. Jing and F. Jingqi, "Fire alarm system based on multi-sensor bayes network," *Procedia Engineering*, vol. 29, pp. 2551–2555, 2012.
- [11] C. Tao, J. Zhang, and P. Wang, "Smoke detection based on deep convolutional neural networks," in *Proceedings of the 2016 International Conference On Industrial Informatics—Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICI)*, pp. 150–153, Wuhan, China, December 2016.
- [12] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," in *Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 877–882, Florence, Italy, October 2016.
- [13] R. Sowah, K. O. Ampadu, A. Ofoli, K. Koumadi, G. A. Mills, and J. Nortey, "Design and implementation of a fire detection and control system for automobiles using fuzzy logic," in *Proceedings of the IEEE Industry Applications Society Annual Meeting*, vol. 2, pp. 3–5, October 2016, Portland, OR, USA.
- [14] R. Sowah, A. R. Ofoli, S. Krakani, and S. Fiawoo, "A web-based communication module design of a real-time multi-sensor fire detection and notification system," in *Proceedings of the 2014 IEEE Industry Application Society Annual Meeting*, pp. 1–6, Vancouver, Canada, October 2014.
- [15] R. A. Sowah, A. R. Ofoli, S. N. Krakani, and S. Y. Fiawoo, "Hardware design and web-based communication modules of a real-time multisensor fire detection and notification system using fuzzy logic," *IEEE Transactions on Industry Applications*, vol. 53, no. 1, pp. 559–566, 2017.
- [16] R. Sowah, A. R. Ofoli, S. Krakani, and S. Fiawoo, "Hardware module design of a real-time multi-sensor fire detection and notification system using fuzzy logic," in *Proceedings of the 2014 IEEE Industry Application Society Annual Meeting*, pp. 4–9, Vancouver, Canada, October 2014.
- [17] K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed, and S. W. Baik, "Efficient fire detection for uncertain surveillance environment," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3113–3122, 2019.

- [18] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18174–18183, 2018.
- [19] K. Wang, Y. Zhang, J. Wang, Q. Zhang, B. Chen, and D. Liu, "Fire detection in infrared video surveillance based on convolutional neural network and SVM," in *Proceedings of the 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, pp. 162–167, Shenzhen, China, July 2018.
- [20] S. Khan, K. Muhammad, S. Mumtaz, S. W. Baik, and V. H. C. de Albuquerque, "Energy-efficient deep CNN for smoke detection in foggy IoT environment," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9237–9245, 2019.
- [21] C. Hu, P. Tang, W. Jin, Z. He, and W. Li, "Real-time fire detection based on deep convolutional long-recurrent networks and optical flow method," in *Proceedings of the 2018 37th Chinese Control Conference (CCC)*, pp. 9061–9066, Wuhan, China, July 2018.
- [22] J. Zeng, Z. Lin, C. Qi, X. Zhao, and F. Wang, "An improved object detection method based on deep convolution neural network for smoke detection," in *Proceedings of the 2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 184–189, Chengdu, China, July 2018.
- [23] A. Filonenko, L. Kurnianggoro, and K. H. Jo, "Comparative study of modern convolutional neural networks for smoke detection on image data," in *Proceedings of the 2017 10th International Conference on Human System Interactions, HSI*, Ulsan, South Korea, July 2017.
- [24] Z. Yin, B. Wan, F. Yuan, X. Xia, and J. Shi, "A deep normalization and convolutional neural network for image smoke detection," *IEEE Access*, vol. 5, pp. 18429–18438, 2017.
- [25] V. Sedlák, V. Stopjaková, and J. Brenkuš, "A real-time method for smoke detection in monitored forest areas," in *Proceedings of the International Conference On Applied Electronics*, Pilsen, Czech Republic, September 2017.
- [26] Z. Zhong, M. Wang, Y. Shi, and W. Gao, "A convolutional neural network-based flame detection method in video sequence," *Signal, Image Video Process*, vol. 12, no. 8, pp. 1619–1627, 2018.
- [27] J. Fonollosa, A. Solórzano, and S. Marco, "Chemical sensor systems and associated algorithms for fire detection: a review," *Sensors (Basel)*, vol. 18, no. 2, p. 553, 2018.
- [28] O. Maksymiv, T. Rak, and D. Peleshko, "Real-time fire detection method combining AdaBoost, LBP and convolutional neural network in video sequence," in *Proceedings of the 2017 14th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, pp. 351–353, Lviv, Ukraine, February 2017.
- [29] DHT11 temperature and humidity sensor—temp/humid robot R Us, 2019, <https://www.robot-r-us.com/vmchk/sensor-temp/humid/dht11-temperature-and-humidity-sensor.html>.
- [30] In-depth: How MQ2 gas/smoke sensor works? and interface it with Arduino, 2019, <https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/>.
- [31] IR infrared 4 wire flame detection sensor module IR flame sensor module detector smartsense for Arduino-in sensors from electronic components, 2019, <https://www.aliexpress.com/item/32849642745.html>.
- [32] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," in *Proceedings of the 2017 International Conference on Communications, Signal Processing, and Systems (ICCSP 2017)*, pp. 588–592, Harbin, China, July 2017.
- [33] X. Pei, "Emphysema classification using convolutional neural networks," in *Intelligent Robotics and Applications*, vol. 9244, pp. 455–461, Springer, Berlin, Germany, 2015.
- [34] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," 2014, <http://arxiv.org/abs/1404.3978>.
- [35] TensorFlow how to retrain inception's final layer for new categories tutorial—Youtube, 2018, <https://www.youtube.com/watch?v=m2D02eZTB4s>.
- [36] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1419–1434, 2019.
- [37] Y. Li, A. Wu, N. Dong, J. Han, and Z. Lu, "Smoke recognition based on deep transfer learning and lightweight network," in *Proceedings of the 2019 Chinese Control Conference (CCC)*, pp. 8617–8621, Guangzhou, China, July 2019.
- [38] Welcome to colab—colaboratory, 2019, <https://colab.research.google.com/notebooks/welcome.ipynb>.
- [39] R. Phadnis, J. Mishra, and S. Bendale, "Objects talk—object detection and pattern tracking using tensor flow," in *Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 1216–1219, Coimbatore, India, April 2018.
- [40] D. Demirovic, E. Skejic, and A. Serifovic-Trbalic, "Performance of some image processing algorithms in tensorflow," in *Proceedings of the 2018 25th International Conference On Systems, Signals and Image Processing (IWSSIP)*, pp. 1–4, Maribor, Slovenia, June 2018.
- [41] G. Nasr, E. Badr, and C. Joun, "Cross-entropy error function in neural networks: forecasting gasoline demand," in *Proceedings of the International FLAIRS Conference*, pp. 381–384, Sarasota, FL, USA, August 2014.