

Article

Realization of People Density and Smoke Flow in Buildings during Fire Accidents Using Raspberry and OpenCV

Gajanand S. Birajdar ¹, Mohammed Baz ², Rajesh Singh ¹, Mamoon Rashid ^{3,*}, Anita Gehlot ¹, Shaik Vaseem Akram ¹, Sultan S. Alshamrani ⁴ and Ahmed Saeed AlGhamdi ²

¹ School of Electronics and Electrical Engineering, Lovely Professional University, Jalandhar 144411, India; birajdar.41800214@lpu.in (G.S.B.); rajesh.23402@lpu.co.in (R.S.); anita.23401@lpu.co.in (A.G.); vaseem.11814442@lpu.in (S.V.A.)

² Department of Computer Engineering, College of Computer and Information Technology, Taif University, P.O. Box 11099, Taif 21994, Saudi Arabia; mo.baz@tu.edu.sa (M.B.); asjannah@tu.edu.sa (A.S.A.)

³ Department of Computer Engineering, Faculty of Science and Technology, Vishwakarma University, Pune 411048, India

⁴ Department of Information Technology, College of Computer and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; susamash@tu.edu.sa

* Correspondence: mamoon.rashid@vupune.ac.in; Tel.: +91-781-434-6505

Abstract: Fire accidents in residential, commercial, and industrial environments are a major concern since they cause considerable infrastructure and human life damage. On other hand, the risk of fires is growing in conjunction with the growth of urban buildings. The existing techniques for detecting fire through smoke sensors are difficult in large regions. Furthermore, during fire accidents, the visibility of the evacuation path is occupied with smoke and, thus, causes challenges for people evacuating individuals from the building. To overcome this challenge, we have recommended a vision-based fire detection system. A vision-based fire detection system is implemented to identify fire events as well as to count the number people inside the building. In this study, deep neural network (DNN) models, i.e., MobileNet SSD and ResNet101, are embedded in the vision node along with the Kinect sensor in order to detect fire accidents and further count the number of people inside the building. A web application is developed and integrated with the vision node through a local server for visualizing the real-time events in the building related to the fire and people counting. Finally, a real-time experiment is performed to check the accuracy of the proposed system for smoke detection and people density.

Keywords: fire detection; Raspberry Pi; deep neural network (DNN); MobileNet SSD; ResNet101; vision node



Citation: Birajdar, G.S.; Baz, M.; Singh, R.; Rashid, M.; Gehlot, A.; Akram, S.V.; Alshamrani, S.S.; AlGhamdi, A.S. Realization of People Density and Smoke Flow in Buildings during Fire Accidents Using Raspberry and OpenCV. *Sustainability* **2021**, *13*, 11082. <https://doi.org/10.3390/su131911082>

Academic Editors: Cheuk Lun Chow, Koen Steemers and Asterios Bakolas

Received: 9 July 2021

Accepted: 30 September 2021

Published: 7 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, there have been several fire-related injuries causing widespread devastation and a significant loss of human life. According to the National Crime Records Bureau, over 18–20 people die in India every day as a result of fire [1], and these data indicate that the several fire incidents occur in residential buildings vs. several deaths. In India, almost 6296 people die each year because of fires and other similar causes. Many of these deaths could be prevented if fires were detected early and people were directed to a safe spot. In the building, a fire accident is an unplanned or accidental occurrence. Dense residential development, entangled wires, and a failure to implement fire safety standards are just a few causes of fire-related devastation in the country. Moreover, in India, the construction of high-rise buildings has increased dramatically and the fire management in these residential buildings is much more difficult to handle due to their built form and height. As per the national building code, the safety measures required in high-rise buildings are fire alarm systems, smoke detectors, sprinklers, PA system, safe fire escape routes, sufficient water supply, fire pumps, refuge area and authorized electrical system, etc.

Generally, there are two types of fire detection methods, namely, traditional fire detection systems and vision-based fire detection systems [2]. The traditional fire detection method utilizes sensors such as thermal, flame, and smoke sensors. However, these sensors lack the capability of detecting the dynamic and static features of fire flames such as motion and color, respectively [3]. The sensors and detectors are not feasible for obtaining multi-dimensional data regarding fire and smoke such as location, direction, severity, and growth, etc. [4]. Moreover, in this traditional method, there is a requirement of human intervention by triggering the alarm after receiving sensor data. The sensor requires a specific intensity of fire for detection, and it takes more time to confirm whether it is a fire or not. Hence, there is a requirement of authenticating fire events that were detected through the sensors. In order to authenticate fire events, the vision-based system is one of promising and evolving technology. A vision-based system is feasible for detecting the static and dynamic features of fire such as color and motion with the assistance of a camera [5]. The information regarding fire and smoke obtained through a vision-based system is processed through the emission and reflection of light. The camera module in a vision-based system permits visualizing and recording real-time visuals data. Furthermore, the camera module acts as a volume sensor for detecting smoke and fire. The vision-based fire detection methods are classified based on flame, fire location, visible range, infrared range, and multimodal [6]. Vision-based fire detection methods are also utilized for identifying the number of people stuck in the building during an accident. During this time, visibility is decreased due to smoke and flame, and it causes difficulty for people trying to find the right path for exiting the building [7]. Currently, traditional fire alarm systems have limitations, including surveillance coverage, reaction time, human interaction, and also detailed information of the fire, including its rate of spread and severity [2]. However, for a variety of reasons, complexity and false triggering continue to be a concern in fire detection.

Moreover, predicting and guiding the correct exit path during fire flow is crucial. Here, the dynamic exit path will play an appropriate role over the static path. Additionally, due to the dynamic and static characteristics of fire and smoke, developing a reliable fire detection system has gained attention, where machine/deep learning technologies are appropriate and feasible to integrate with the vision-based system for resolving the limitations [8].

In this study, we have implemented a Raspberry Pi-based vision system for detecting the flow of smoke and also people estimation with a deep neural network (DNN). Training a DNN model for fire detection and people counting is performed with a sample dataset and further trained models are used in the system for real-time fire detection and people count estimation. Moreover, we have designed a web application for visualizing real-time visual data obtained from the vision system. The contributions of this study are as follows:

- A real-time vision node-based fire detection system based on DNN is proposed with the web application.
- A vision node is developed by integrating the Raspberry Pi and Kinect sensors. Moreover, deep neural network (DNN) models, i.e., MobileNet SSD and ResNet101, are embedded in the vision node for real-time detection of the fire accident and then the number of people inside the building is counted.
- A web application is developed for visualizing and monitoring the visuals of the building from the vision node through a local server enabled through the internet.
- An experimental study is performed to analyze the accuracy of the proposed system for fire detection and people density.

The rest of the paper is structured in the following sections. Section 2 covers the related works of detecting fires through a vision-based system. Section 3 covers the proposed architecture. Section 4 covers hardware and software description. Section 5 covers methodology for fire detection and people counting. Section 6 presents the experimental results with comparative analysis and the article ends with the Conclusion.

2. Related Works

In this section, we present previous research studies that mainly focused on fire detection and counting the number of people during fire accidents. An automated video surveillance system is implemented for crowd counting and detecting normal or abnormal events [9]. Human detection with cognitive science approach is proposed for managing the crowd using Histogram of Oriented Gradients (HOG) features [10]. A system based on Raspberry Pi is proposed for counting multiple humans in a scene by counting their heads [11]. Mini standalone stations OpenCV and Raspberry Pi have been proposed for counting people [12]. The estimation of crowd density is performed by counting people using the background subtraction model and perspective correction [13]. A (HOG)-based people counting technique was presented in which it considers video as an input and predicts people count [14]. A fire alarm system using Raspberry Pi and Arduino is proposed in which it detects smoke in the air due to fire and captures image via a camera connected to it and then sends an alert message to the webpage [15].

An effective method is proposed for obtaining the edges by identifying the variation in the image of fire [16]. A Canny edge detector and improved edge detector are implemented with a multi-stage algorithm with respect to images of simple and steady fires and flames [17]. A method based on wavelet algorithms and fast Fourier transformation (FFT) is implemented for assessing the forest fire patterns in video situations [18]. A technique based on the block technique is suggested for detecting smoke with k-temporal information of shape and color [19]. A smoke detection algorithm is proposed for a tunnel environment utilizing motion feature, invariant moment methods, and motion history image [20]. A color-dependent method is implemented for detecting the divergence in the sequential images [21].

A generic rule-enabled technique is implemented for differentiating the luminance from chrominance to detect distinct types of fire and smoke in images [22,23]. Moreover, the utilization of the YCbCr enhances the detection rate of fire from images over RGB due to its unique features of splitting luminance [24]. A fire detection technique is implemented with the assistance of Raspberry Pi and OpenCV for automating detection and alerting the authorities with an alarm [25]. A Raspberry Pi 3 based vision system is implemented for enhancing home security with OpenCV [26]. A fire detection framework is proposed based on the image with deep learning and the tiny-YOLO (You Only Look Once) v3 deep model is utilized [27]. An efficient computer vision-enabled fire detection method is implemented, where the red element of RGB distinguished the areas of the fire [28]. A novel intelligent fire detection approach is implemented for achieving high detection rates, low false alarm rates, and high speeds. To reduce CNN computations, the motion detection method in [29] is implemented in the study. In the study, scenario clusters are created throughout the building fire risk analysis process, and the fire risk indexes are chosen based on the number of deaths and direct property loss [30]. A framework for salient smoke detection and predicting smoke's existence may be employed in video smoke detection applications. As a result, the presence of smoke in a picture may be predicted by combining a deep feature map and a saliency map [31]. A deep domain adaptation-based method for video smoke detection is proposed to extract a powerful feature representation of smoke because the extraction of the feature of real smoke images degrades the performance of the trained model [32].

3. Proposed Architecture

The evolution of a vision-based system assists detecting fire and smoke in real-time through visuals. During the fire accident, the fire spread quickly in the building with respect to environmental parameters, and the fire spreading inside the building creates a amount of huge smoke where the visibility becomes difficult for people stuck in the building. Thus, a real-time vision-based system assists the authorities in monitoring and providing evacuation paths and identifying the number of people stuck in the building, which eventually minimizes the loss of life. With the advantage of the vision-based system,

we have proposed architecture and a system that is beneficial for implementing a real-time vision-based system by using Raspberry Pi and web application, as shown in Figure 1.

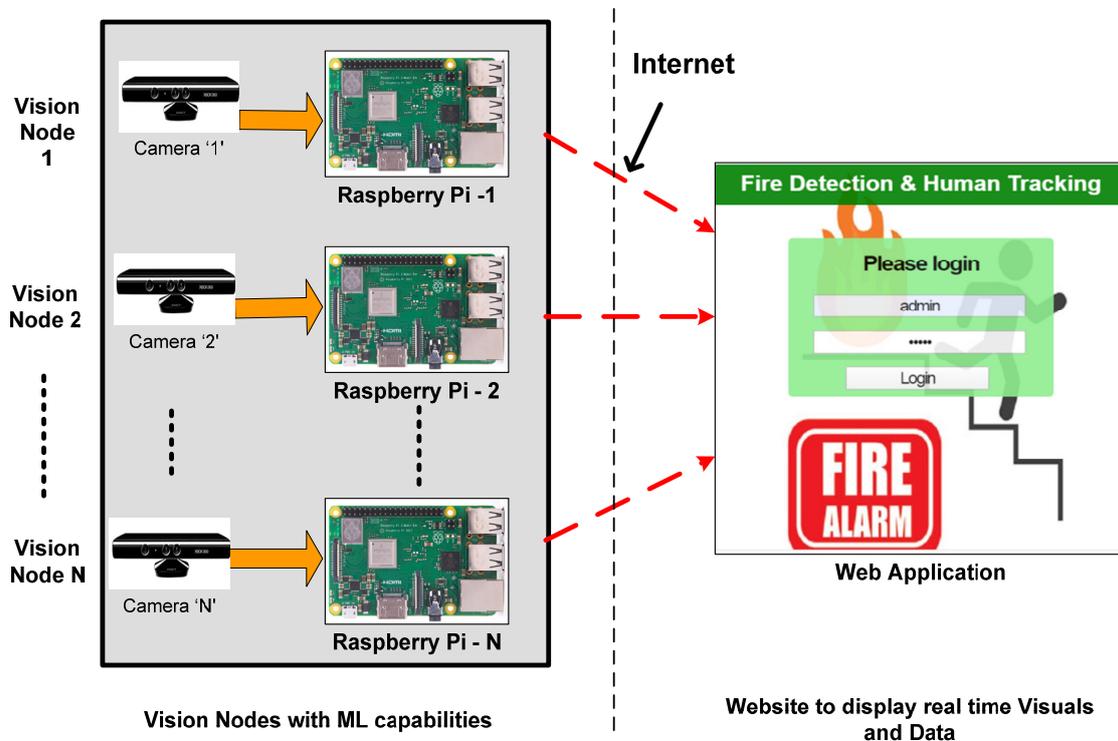


Figure 1. Detailed architecture of the proposed system.

The architecture comprises three components, namely, Raspberry Pi, Kinect sensor, and web application. Generally, Raspberry Pi used with a Kinect sensor empowers the implementation of a vision-based system for sensing visual data such as images and videos. A web application is developed for visualizing real-time data that is obtained from the vision node as Raspberry Pi is integrated with a web application through a local server enabled via the internet. The real-time detection of fire and people count estimation is achieved with the vision node because it is loaded with a machine learning (ML) model, i.e., a deep neural network. The real-time result regarding fire detection and people count is communicated with the web application through the local server.

4. Hardware and Software Description

For realizing fire detection and people count estimation, we have used a combination of hardware and software. In the following section, we discuss hardware components in use, their features, software tools, ide, and web development, etc.

4.1. Hardware Description

The proposed system is implemented using Raspberry Pi as it possesses low cost and is self-contained. It allows flexibility in the choice of programming languages and the installation of software that could be used. It can serve as a webpage by installing Apache HTTP Webserver on it. Moreover, it is possible to deploy a DNN model for object detection inside Raspberry Pi. The model used in this system is Raspberry Pi 3 Model B possessing a 1 GB class 10 micro-SD card, which is powered by a 700 MHz single-core ARM11 microprocessor and 512 MB RAM along with two USB ports and an HDMI connector. Raspberry Pi project development started in 2006. It is an inexpensive computer that uses a Linux-based operating system, equipped with a 700 MHz ARM-architecture CPU, having 512 MB RAM, and featuring two USB ports and an Ethernet controller. It

can handle full HD 1080 video playback by using the onboard Video core IV graphics processing unit (GPU). The detailed features of Raspberry Pi 3 are mentioned in Table 1.

Table 1. Features of Raspberry Pi Model B.

Parameters	Value/Feature
Model	Raspberry Pi 3 Model B
SoC	BCM2837A0/B0
Memory	1 GB
Form Factor	Standard
Ethernet	Yes
Wireless	Yes
GPIO	40 Pin

For capturing real-time visuals, we have integrated a Kinect sensor with a Raspberry Pi module. A Kinect sensor is used for the application because of its multi-output source capability and to which the function that requires easy interchangeability between normal vision camera output and IR feed output can be achieved. When interfaced with Raspberry Pi and after some optimizations such as overclocking, increasing GPU, and RAM allocation, we obtained an average FPS of 30. It is commonly utilized in image processing, machine learning, and surveillance projects. Table 2 provides detailed features of the Kinect sensor that is used in our system.

Table 2. Features of Kinect camera sensor.

Parameters	Value/Feature
Camera	640 × 480 pixels @ 30 Hz (RGB camera) 640 × 480 pixels @ 30 Hz (IR depth-finding camera)
Connectivity	USB 3.0 (type-A for original model; proprietary for Xbox 360 S)
Frame per second (FPS)	30 FPS
Sensor technology	Time of Flight
Maximum distance	4–5 m
Microphone	4 microphones assisting single speaker voice recognition

4.2. Software Description

In this section, we discuss software tools, compatibility of hardware, web development, methods, and algorithms, etc. Once we obtain real-time visuals from the vision node, they will be processed through the algorithms that are written specifically for this study. We have developed the algorithms through Python and OpenCV as the combinations of Open CV and Deep Neural Network can detect objects in real-time input frames captured from vision nodes. There are two primary issues to avoid when utilizing Raspberry Pi for deep learning: memory limitations (only 1 GB on the Raspberry Pi 3) and processor speed limitations. This makes using larger, more complex neural networks nearly difficult. Instead, we need to deploy networks such as MobileNet and SqueezeNet, which are more computationally efficient and possess smaller memory/processing footprint. In our system, we have implemented a MobileNet network. These networks are better suited for Raspberry Pi with sufficient speed. Figure 2 illustrates steps followed to optimize OpenCV for the Raspberry Pi module. However, in our study, we have trained the required models inside a laptop and then deployed these pretrained models in Raspberry Pi for object detection and counting. DNN provides many modules for detecting the person in the image. OpenCV DNN runs faster inference than the TensorFlow object detection API with higher speed and low computational power. Those modules have different accuracy and speed of response.

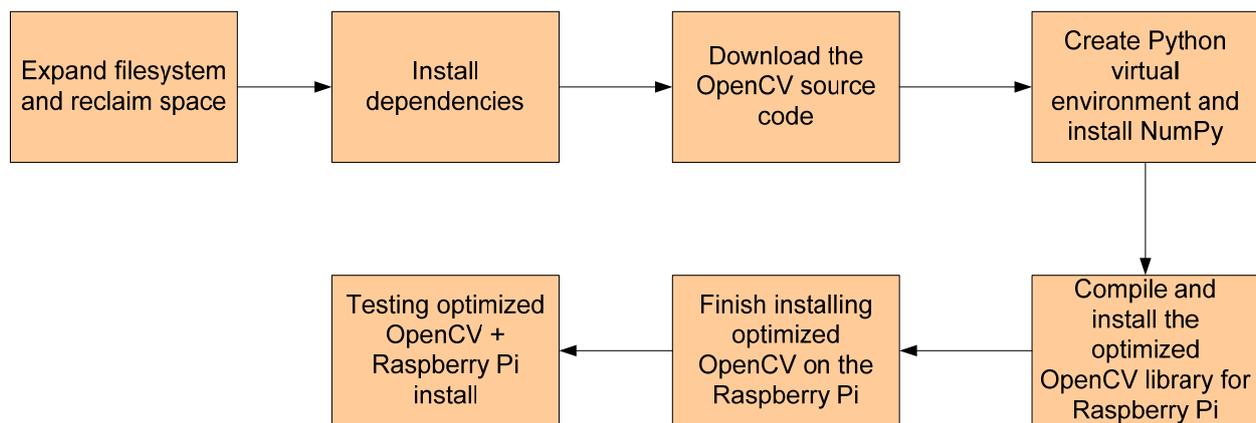


Figure 2. Steps for OpenCV optimization for Raspberry Pi.

5. Methodology for Fire Detection and People Counting

In this section, we present the methodology that is applied for detecting fire and people with a vision-based system. Additionally, in this section, we also present the training dataset detailed with a flow diagram.

5.1. Fire Detection and People Counting

The functional block diagram is discussed to realize fire detection and people counting, as shown in Figure 3. For the detection of human presence and their counting inside a room, MobileNet SSD is utilized, which is a single shot detector that has been designed by using transfer learning procedures. The existing MobileNet SSD model's knowledge was kept in its first 24 layers, and the last four layers towards the output are trained with the images collected in our datasets that are more relevant to the test application area's conditions and scenarios. After training was completed, we obtain a model that has cumulative knowledge of the existing MobileNet SSD model and also of the images from our datasets, which thereby makes it more reliable. Similarly, for the fire detection objective, ResNet101 was used, which is considered one of the most efficient neural network models that can be used as the backbone for fire detection application. Furthermore, out of its total 101 layers, the last 15 layers are trained with the dataset that we collected from various sources using transfer learning. After training was completed, the model file was then transferred and deployed in Raspberry Pi.

Once the system was activated and powered up, the system imports both the model into play and initially starts taking the input from the Kinect normal vision camera and feeds those into the input layers of the trained ResNet101 fire detection model after pre-processing, i.e., resizing, formatting, etc. The model then processes it and outputs a prediction about the detection of fire along with the confidence of the prediction. Our system compares the confidence received from the model with a preset threshold value. In the case where confidence is above the preset threshold value, the system considers the case as having a fire detected, and then it moves to the human detection and counting module. The system shifts the input source as the IR feed from the Kinect and then feeds it into the MobileNet SSD model after preprocessing according to compatibility in the human detection and counting module. The model then processes the input frames and detects the number of human presence in its vicinity or FOV. It then streams the real-time output and the live feed to a network, which is then parsed by an application listening to the stream port. This feed can then be used by the concerned authorities to plan out further steps and evacuation procedures.

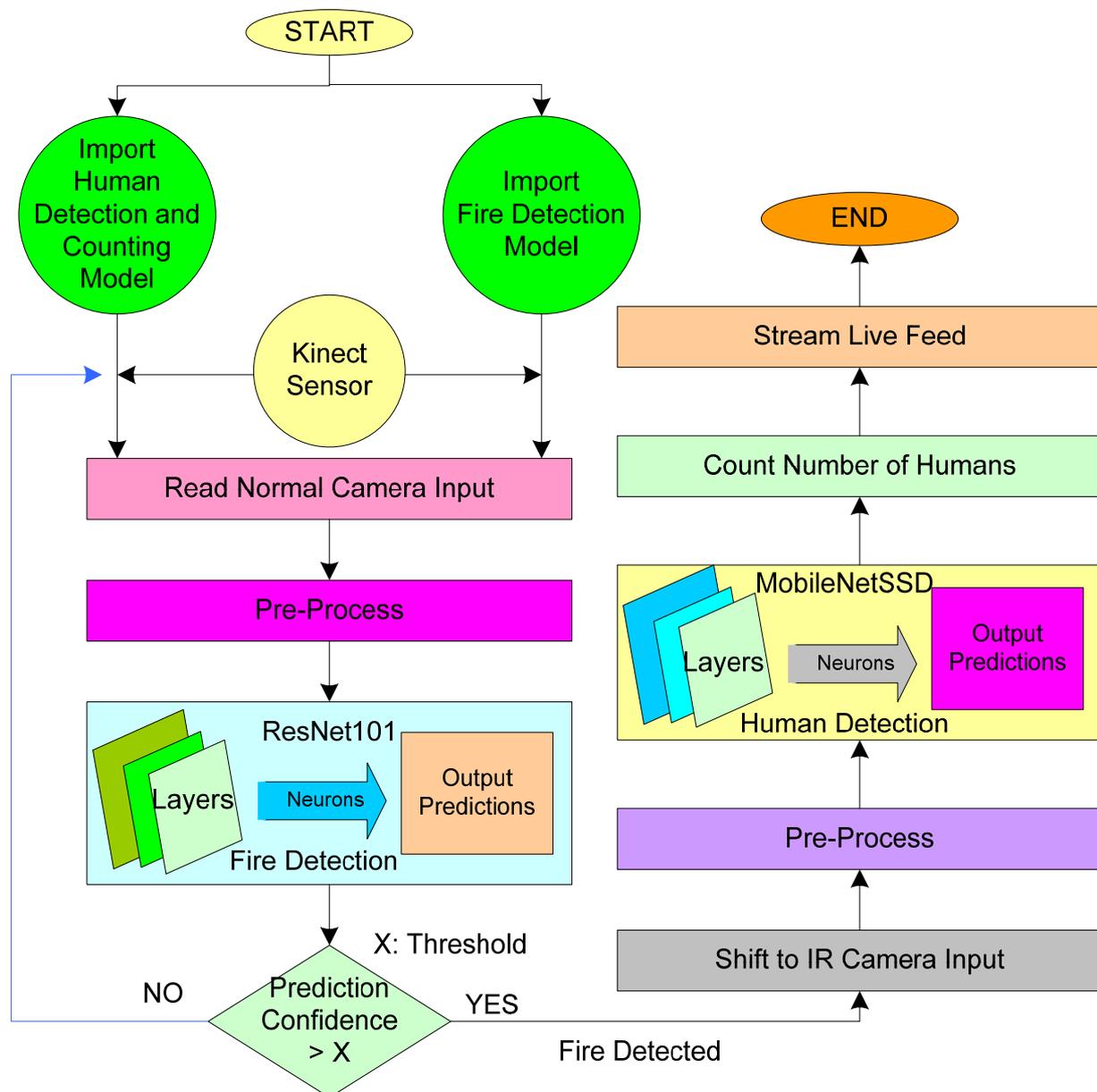


Figure 3. Methodology of detecting fire and counting people.

5.2. Training Dataset

People are recognized by using DNN once the training model is ready, as shown in the architectural diagram. Furthermore, DNN has attracted the attention of researchers for fire detection and predicting fire flow. Deep learning methodology differs from traditional computer vision-based fire detection in major ways. The first is that the characteristics are automatically collected in the network after training with a vast quantity of different training data rather than being studied by an expert. The lack of strong data for assessing and assessing the recommended approach is one of the key drawbacks of vision-related activities. To identify an appropriate dataset, we investigated datasets that have previously been employed in research. However, the diversity of data coming from the video is insufficient for training, and we cannot expect it to perform well in realistic fire detection circumstances. By extracting frames from fire and smoke films and gathering photographs from the internet, we aimed to construct a diversified dataset. Fire photos from Foggia's dataset and photographs from the internet make up our training set. Our dataset was varied by using smoke images from several online sites. In order to create our final fire-smoke

dataset for use in this investigation, we extracted frames from videos and randomly picked a few photos from each one. Figure 4 illustrates a sample training dataset for fire detection. This dataset is taken from different resources such as the internet, live images, etc. This training procedure has been carried out in another system that has a better configuration as the training model inside Raspberry Pi is still challenging. Figure 5 illustrates a sample training dataset for people detection. For training, we have taken standard parameters set from previous studies [33], as shown in Table 3.

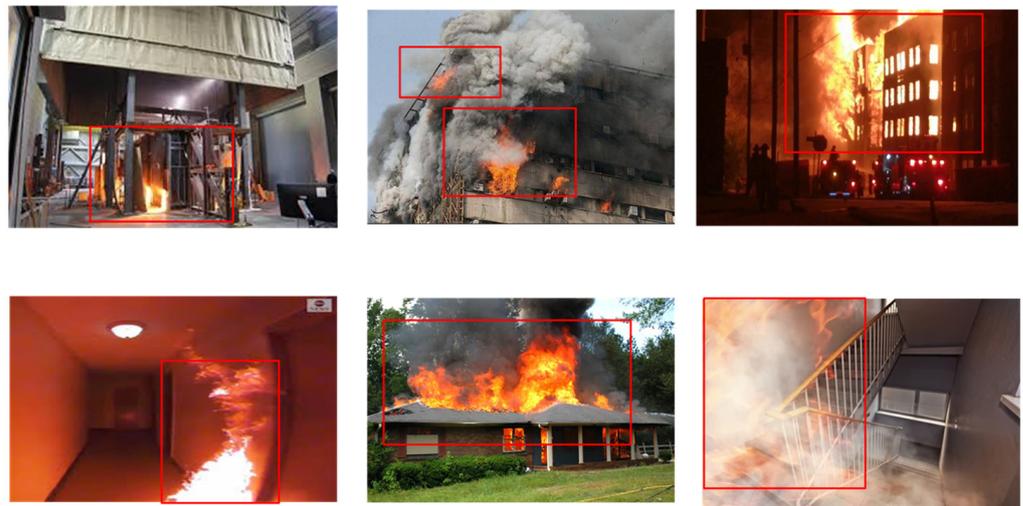


Figure 4. Training Dataset for fire detection.

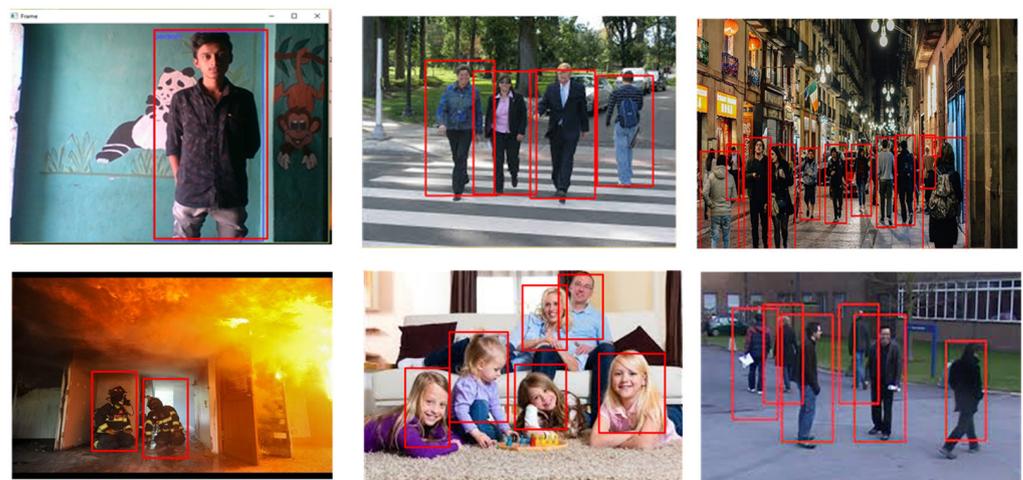


Figure 5. Training Dataset for people detection.

Table 3. Training parameter for people counting.

Parameter	Value
Test iteration	673
Test interval	10,000 s
Base Learning Rate	0.0005
Learning Rate Policy	multistep
Gamma ratio	0.5
Step interval	20,000
Maximum iterations	120,000
Weight decay	0.00005

During training for people counting, the hyperparameters considered and presented in Table 3 and hyperparameters for training for fire prediction in Table 4 are as follows: test iteration, test interval, base learning rate, learning rate policy, gamma ratio, step interval, maximum iterations weight decay, step size, batch size, and pre-train weight. The number of times a batch of data has been processed by the algorithm in a neural network is called a test iteration. The test iteration is performed many times to avoid overfitting and errors during the training of a model. Test interval describes the time interval between two iterations. The base learning rate is a configurable hyperparameter employed in the training of neural networks that has a small positive value in the range between 0.0 and 1.0. The learning rate (lr) policy is utilized for adjusting lr during training by minimizing the lr concerning a pre-defined schedule. Learning rate decay is the term used to describe how the learning rate varies over time. Weight decay is a parameter that prevents growing weights too large during the training of neural networks. The amount of training cases in a single forward/reverse pass is called the “batch size”, and if we increase the batch size, it requires extra memory. During pretraining of the model, pre-trained weights are preset for minimizing training time.

Table 4. Training parameter for fire detection.

Parameter	Value
Test Iteration	150,000
Step size	100,000
Weight decay	0.00004
Learning rate	0.01
Batch size	256
Learning rate decay	0.00001 (iteration equal step size)
Pre-train weight	ResNet-101

(a) People detection and Counting:

The procedure of fire detecting and people count estimation are illustrated in Figures 6 and 7. People count estimation is implemented with DNN after the training model according to Figure 6. People recognition and counting are achieved through the following sequential steps: Initially, the pre-trained object detection model with labels is loaded. When the camera captures the image as an input, it is fed into the DNN network after pre-processing. Here, the network decides whether the object is known or not. In the case where the object is unknown, it starts taking a picture. In the case where the object is known, it assigns a label and draws the bounding boxes for all detected objects and counts the labels. Finally, it displays the count and repeats the process.

(b) Fire Detection:

Similarly, detection of fire is implemented with DNN after the training model according to Figure 7, and sequential steps are explained as follows: In the initial stage, the camera captures the image as input, and pre-processing is applied to the image. Here, the color image is converted into grey color, and the pre-trained model for fire detection is loaded. The image is fed to the DNN network, where it decides whether the object in the image is a fire or not.

If the result is a fire detected, then it immediately sends information on the web application. Table 5 presents the parameters that are utilized for the proposed system. Eight parameters of the proposed system in Table 5 are presented.

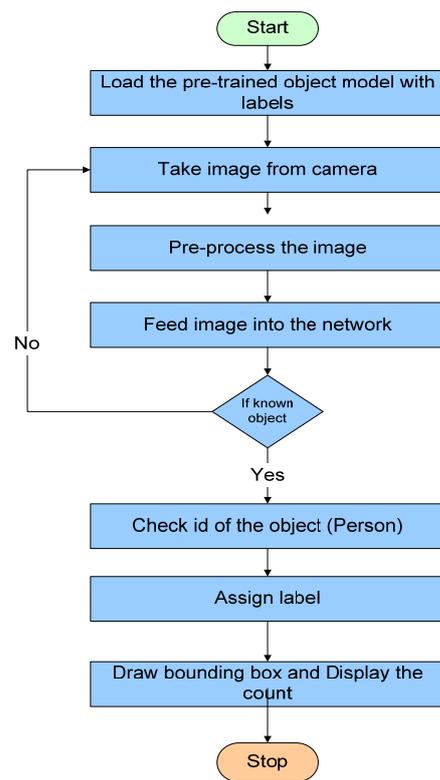


Figure 6. Flowchart of people counts estimation.

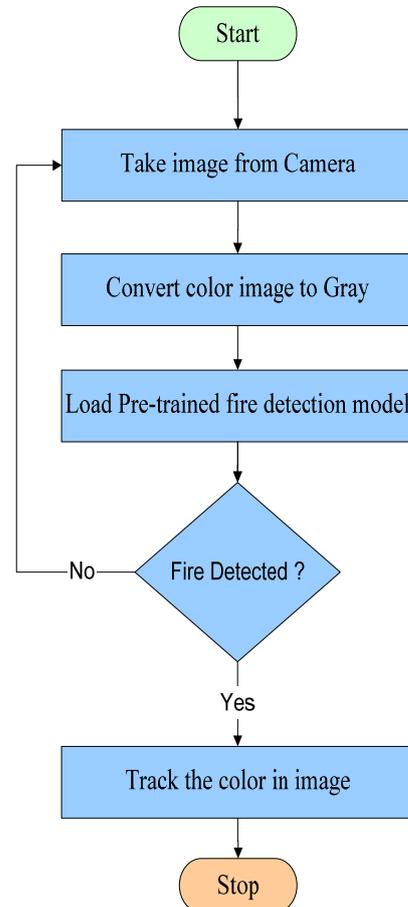


Figure 7. Flowchart of the fire detection system.

Table 5. Description of parameters of the proposed system.

C_{IC} → Capturing Image	D_{OD} → Deciding Object
P_{PP} → Pre-processing Image	B_{DB} → Draw Bonding Box
I_{LD} → Loading pre-trained model	D_M → Display Message
S_S → Status of detection	C_{CO} → Count object

Table 6 explains the algorithm that is implemented for real-time fire detection. The parameters for the algorithm are presented in the above table. The parameters that are utilized for fire detection are C_{IC} , P_{PP} , I_{LD} , D_{OD} , and D_M . Moreover, five steps are involved in the proposed algorithm.

Table 6. Algorithm for real-time fire detection.

Steps Followed	Function Used in Programming
Input: Real time video of object C_{IC} → Capture the image from input	camera cap = cv2.VideoCapture(0)
P_{PP} → Apply pre-processing on input image, i.e., convert a color image to gray.	gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
I_{LD} → Load pre-trained model for fire detection to DNN Network	fire_cascade = cv2.CascadeClassifier('fire_detection.xml')
D_{OD} → It will decide whether image having known object (Fire) or not.	for (x,y,w,h) in fire: cv2.rectangle(img, (x,y), (x + w,y + h), (0, 0, 255), 2) roi_gray = gray[y:y + h, x:x + w] roi_color = img[y:y + h, x:x + w]
D_M → Display Message Once it finds then track the colour in the image.	cv2.putText(img, "Fire Detected", (40, 40), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2) cv2.imshow('img',img)

Table 7 explains the algorithm that is implemented for people detection and counting. The parameters for the algorithm are presented in the above table. I_{LD} , C_{IC} , P_{PP} , D_{OD} , B_{DB} , and D_M are the parameters that are involved in the algorithm for people detection and counting. Moreover, five steps are involved in the proposed algorithm.

Table 7. Algorithm for real-time people detection and counting.

Steps Followed	Function Used in Programming
Input: Real-time video of the object I_{LD} → Load pre-trained model for fire detection to DNN Network	net = v2.dnn.readNetFromCaffe("MobileNetSSD_deploy.prototxt.txt", "MobileNetSSD_deploy.caffemodel")
C_{IC} → Capture the image from input	camera cap = cv2.VideoCapture(0)
P_{PP} → Apply pre-processing on input image	blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)), 0.007843, (300, 300), 127.5)
D_{OD} → Deciding Object (Person) and label it	label = CLASSES[idx]
B_{DB} → Draw Bonding Box over the Object detected (Person)	cv2.rectangle(frame, (startX, startY), (endX, endY),COLORS
D_M → Display the people count and repeat the process	if(label=='person'): count+=1 print("number of persons ",count)

6. Results

The main objective of the proposed system is to work effectively in fire accident situations. We intend to deploy the proposed system in real-time and to test it rigorously under different scenarios. To evaluate the performance of the proposed system, we have deployed the vision node in real time. In the current scenario, a video surveillance system based on CCTV is generic, and this can be further utilized for fire detection [34]. We have deployed our vision node inside the room in the place of the existing CCTV to capture the

real-time scene. In this system, in order to obtain real-time data from vision nodes, we have built a web application that will be hosted on the local server.

The login window of the application is shown in Figure 8. Only authorized personnel with proper credentials can obtain access to the entire system. To trigger the vision node, we need access the login web portal. As we login to the web portal successfully, all the vision nodes deployed inside the building connected through the internet will be deployed. Once they connect successfully, vision nodes start streaming live visuals along with estimated data such as fire detection status and people count, as shown in Figure 9. The detailed stages involved in the end-end workings are illustrated in Figure 10.



Figure 8. Admin panel to login the portal.

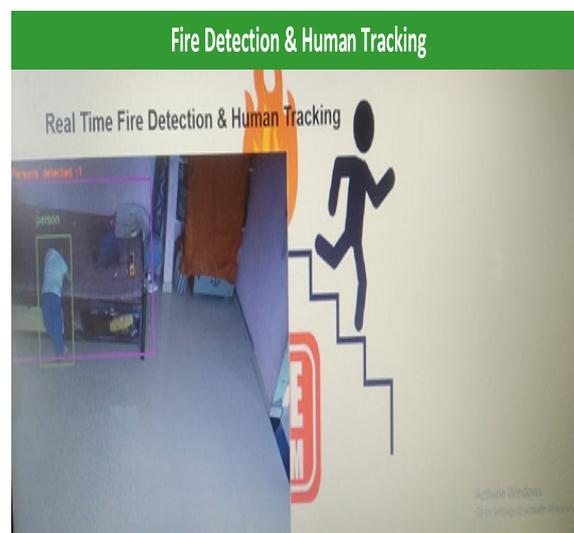


Figure 9. Real-time visualization on website-based application.

In our experiment, we have tested our system in various scenarios. To obtain accuracy and correctness of the people density counting mechanism, we have tested the system based on the distance of people from the camera and light intensity inside the room. Table 8 shows the accuracy of the system as per the distance of persons from the camera. The results are obtained by classifying the distance into four different sets. In each set, the distance from the camera increases to 5 feet.

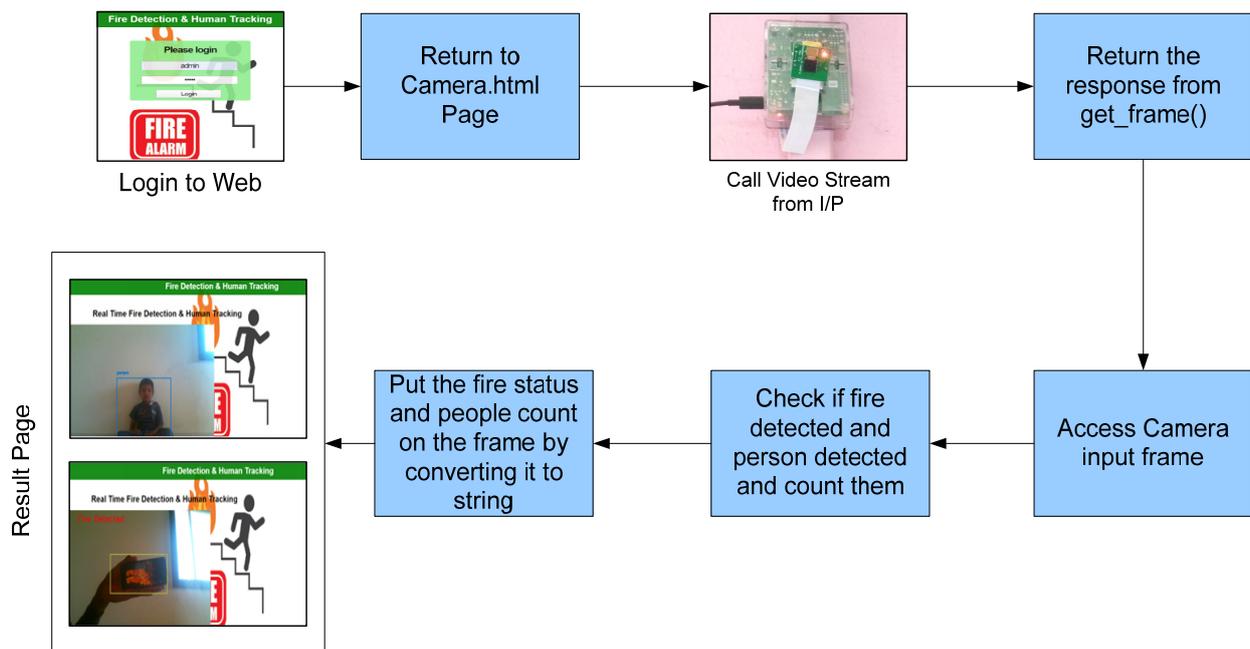


Figure 10. Stages involved in end-end working.

Table 8. Result based on the distance of persons from the system.

SET	Distance from Camera (In Feet)	Result from Captured Image	Accuracy
1	5	Detected 4 out of 4 persons	100%
2	10	Detected 4 out of 4 persons	100%
3	15	Detected 4 out of 4 persons	100%
4	20	Detected 3 out of 4 persons	75%

In SET 1, the distance of a person from the camera is 5 feet, and then the accuracy of the system is 100%. In SET 2, when the distance of a person from the camera changed to 10 feet, the accuracy of the system is the same, i.e., 100%. Similarly in SET 3, the distance of the person from the camera increased to 15 feet, where the accuracy of the system remained the same. However, in SET 4, the distance of a person from the camera increased to 20 feet, and then the accuracy reduced to 75%.

The following images are considered for evaluating the accuracy of the vision-based system. Here, we have considered images with different counts of people in them. The system can detect the number of people in the images and it is represented in Figure 11 and Table 9.

Table 9. Number of people detection using images.

Image	Actual Count of People	Count of People by Vision Node	Accuracy
a	36	28	77.77%
b	5	5	100%
c	8	6	75%
d	14	4	30%

Table 9 presents the number of people that are detected by the vision node. In the case of Image 'a', the actual number of people present is 36, and the vision node can detect the 28 people with 77.77% accuracy. The accuracy for the image 'a' has been affected due to low image resolution. In the case of image 'b', the actual number of people present is five, and the vision node can detect the five people with 100% accuracy. In the case of image 'c',

the actual number of people present is eight, and the vision node can detect the six people with 75% accuracy. In the case of image 'd', the actual number of people present is 14, and the vision node can detect the four people with 30% accuracy. Here, the accuracy has been affected due to blurry images.



Figure 11. People detection by vision node. (a) Actual count of people, 36 (b) Actual count of people, 5 (c) Actual count of people, 8 (d) Actual count of people, 14.

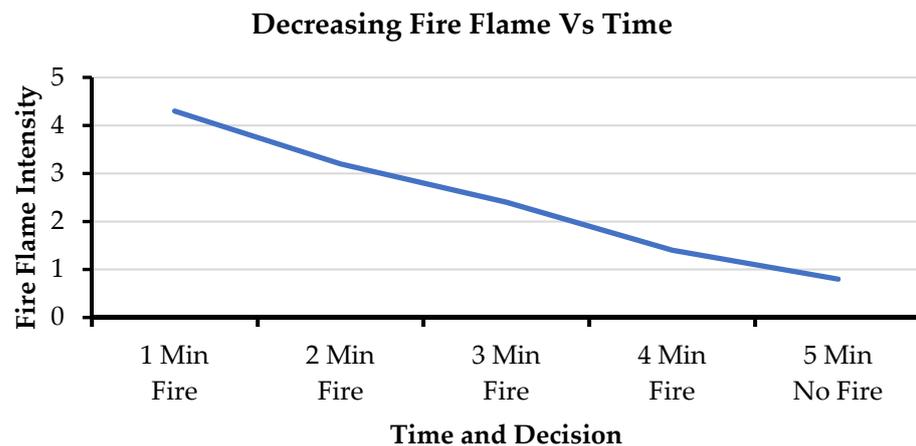
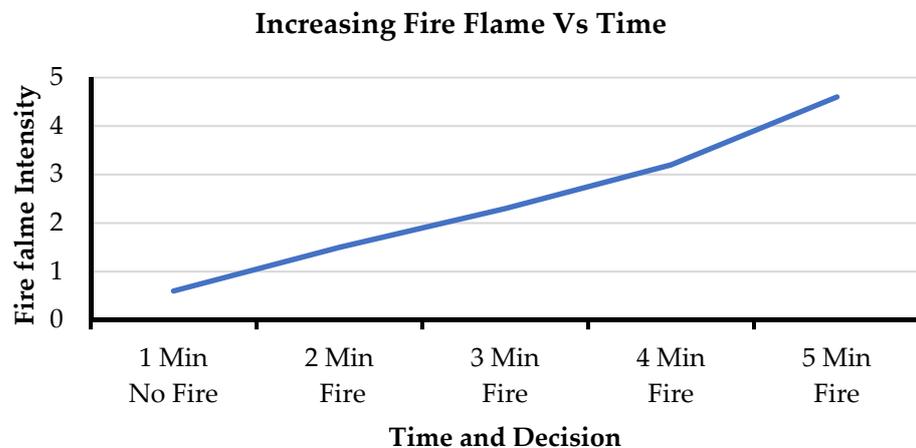
Based on the above result, it is concluded that the accuracy is increased when there is better frame visibility, as in the case of image 'a'. In the case of blurry images, we obtain less accuracy than in the case of image 'd'. Better resolution images will enhance the accuracy of the vision node fire detection system.

Table 10 shows the accuracy of a system based on light intensity inside the room. The results are obtained by considering light from four different sources. In SET 1, the intensity of light inside the room corresponds to sunlight, and the accuracy of the system is 100%. In SET 2, the intensity of light inside the room corresponds to tube light, and the accuracy of the system is the same, i.e., 100%. Similarly in SET 3, the intensity of light inside the room is a partially dark room, and the accuracy of the system is 50%. However, in SET 4, the intensity of light inside the room is absent since there is no sunlight and tube light, and the accuracy reduced to 50%. To improve the result and accuracy in the case of SET 3 and SET 4, we can integrate a night vision camera or thermal imaging camera along with the proposed vision node, which can detect human beings even in low visibilities.

Furthermore, as part of our experiment, we carried various testing on the fire detection side. Different frames captured by vision nodes under different scenarios have been taken as test cases and analyzed over some time. When intensity proceeds above the threshold value, around 0.6 of the system will predict it as fire detected. In the first case, frames with decreasing fire flames were given as an input to the system, and we received a response as shown in Figure 12. In the second case, we have given frames with increasing fire flame intensity as input and received the results shown in Figure 13.

Table 10. Result based on intensity of light inside room.

SET	Intensity Inside Room	Captured Image	Accuracy
1	With Natural Light	Detected 4 out of 4 persons	100%
2	With tube light	Detected 4 out of 4 persons	100%
3	Partial dark room	Detected 2 out of 4 persons	50%
4	Without Natural Light and tube light	Detected 2 out of 4 persons	50%

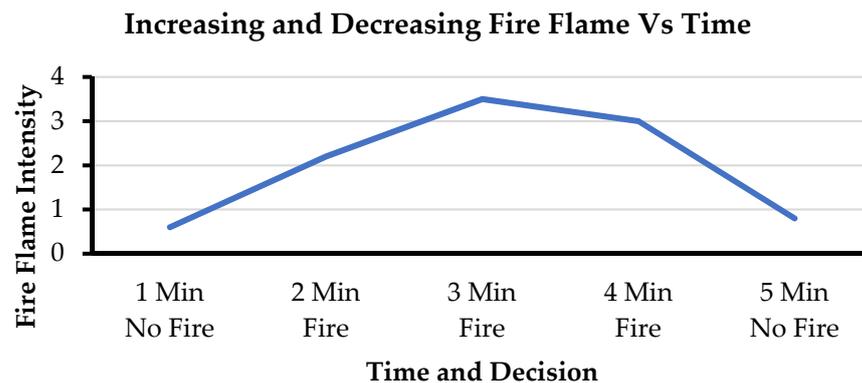
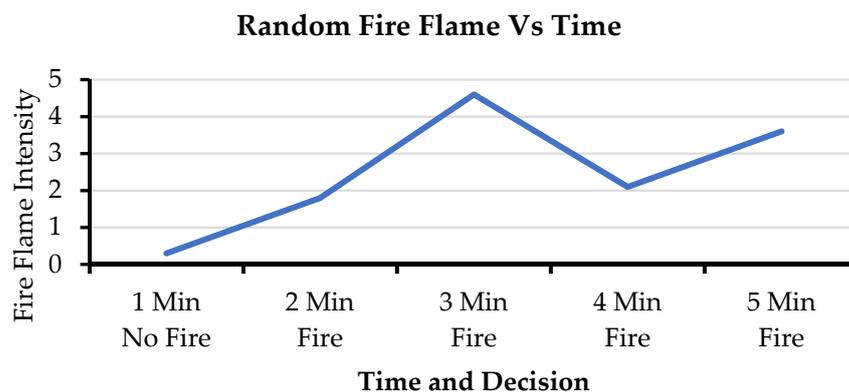
**Figure 12.** Decreasing fire flame vs. time.**Figure 13.** Increasing fire flame vs. time.

In the third scenario, we have tested our system with frames with increasing and decreasing fire flames and received the results shown in Figure 14. In the last case, we have tested the system with frames with random fire flame intensities, and we obtained the results shown in Figure 15. In Figures 12–15, the y-axis represents scaled fire flame intensity, and the x-axis represents time in minutes and decision.

Table 11 illustrates the comparative analysis of the proposed system with previous studies implemented for fire detection. The comparative analysis is performed based on certain parameters: Raspberry Pi, object detection method, real-time implementation, web interface, and retrofitting with CCTV. From the comparative analysis, it was concluded that the proposed system is superior to the previous studies in terms of implementing fire detection with Raspberry Pi and web interface. In this system, a web interface is developed for visualizing real-time information obtained by Raspberry Pi through the Kinect camera. Moreover, retrofitting with CCTV is also possible with the proposed system.

Table 11. Comparative analysis with previous studies.

Research	Raspberry	Object Detection Method	Real-Time Implementation	Web Interface	Retrofitting with CCTV
[2]	No	CNN	Yes	No	Not implemented
[34]	No	Image processing	Yes	No	Not implemented
[33]	No	DNN, CNN	Yes	No	Not implemented
Proposed Study	Yes	DNN	Yes	Yes	Implemented

**Figure 14.** Increasing and decreasing fire flame vs. time.**Figure 15.** Random fire flame vs. time.

7. Discussion

The main objective of the proposed work is the realization of the vision node for the fire accident. In this section, we have discussed some important parameters such as accuracy, restriction, applicability of proposed solution, and further research, etc.

Accuracy is the correctness of measured results with respect to the actual one. The accuracy of the proposed system is validated by multiple methods. In the case of people counting, variation in visible light intensity and distance of the target object is applied. Based on the result, it is observed that accuracy is good in the case of natural visibility and up to 15 feet. Furthermore, lower visibility and increased distance result in low accuracy. There is mostly low visibility present during fire events as space inside the building will be occupied with smoke and fire. In such conditions, in order to obtain good accuracy, it is necessary to incorporate a vision node with night vision capability or thermal imaging capability.

Furthermore, the experiments are carried out to analyze fire flow during fire event detection. A set of experiments was undertaken to elaborate the decision of fire event by the system based on the intensity of fire flame under different scenarios such as increasing flame, decreasing flame, random fame, etc. It is observed that the system has identified

the fire event and fire flow with good accuracy. The proposed solution can be deployed at high-rise buildings, commercial malls, schools, and hospitals, etc., as it is facilitated by a web application and assists the authorities in monitoring events during a fire. It can help to further expedite rescue operations and assist in obtaining control over the fire.

Nowadays, buildings are becoming smart with the use of intelligent devices such as CCTV, smoke detectors, smart lighting, etc. In future research investigations, the proposed vision node will be integrated into CCTV with night vision and thermal imaging capabilities. Furthermore, an intelligent dynamic display will be designed and developed for predicting and displaying the safest exit path.

8. Conclusions

Fire detection and people estimation are very crucial during a fire accident. Early fire detection enables suggesting a safe exit path from the building for the people inside the building. Moreover, it is important to find the number of people stuck inside the building during a fire accident. Thus, in order to detect fires and to estimate the number of people in real time, we require a real-time visual capturing system. In this study, we have implemented a vision node based on Raspberry Pi with machine learning capabilities for detecting fire and estimating people count accurately in real time. Along with the vision node, we have designed a web interface that is capable of triggering all connected vision nodes by authorized signing in to the portal. Here, the vision node interfaces with the web portal through a local server hosted on Raspberry Pi. As the vision node is switched on, it starts capturing the real-time frame. Moreover, a machine learning-based pre-trained model processes real-time frames and sends the estimated result on web portals such as the status of fire detection and count of people. In this study, we also tested the proposed system to analyze the accuracy by considering two different scenarios such as distance of the vision node from people and intensity inside the room. The evaluation of these scenarios is performed by considering four sets of the distance of vision node from people and intensity inside the room. In the case of the distance of vision node from people scenario, we attained 100% accuracy for a distance 15 feet and accuracy reduced to 75 for a distance of 20 feet. In the second scenario with respect to the intensity inside a room, we attained 100% accuracy when there was normal sunlight or when the tube light is on, and accuracy was reduced to 50% for the other two sets. Finally, comparative analyses with previous studies related to fire detection and estimation are presented. In the future, we will integrate a dynamic exit path display system for assisting individuals in safely exiting from the building during a fire event. In general, nowadays, buildings may have installed CCTV cameras with respect to the point of view of security. We can integrate our smart device with the existing CCTV camera, which will further act as a vision node for our system. In future research, the proposed vision node will be integrated into CCTV with night vision and thermal imaging capabilities. Furthermore, an intelligent dynamic display will be designed and developed for predicting and displaying the safe exit path

Author Contributions: R.S., A.G. and G.S.B. made contributions to conception and manuscript writing; M.B. and S.S.A. examined and supervised this research study and its outcomes; M.R., S.V.A. and A.S.A. revised and polished the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Taif University Research Supporting Project number (TURSP-2020/311), Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- National Crime Records Bureau. Available online: <https://ncrb.gov.in/> (accessed on 27 July 2021).
- Valikhujaev, Y.; Abdusalomov, A.; Im Cho, Y. Automatic fire and smoke detection method for surveillance systems based on dilated cnns. *Atmosphere* **2020**, *11*, 1241. [CrossRef]
- Celik, T. Fast and efficient method for fire detection using image processing. *ETRI J.* **2010**, *32*, 881–890. [CrossRef]
- Wong, K.K.A. A Study of Video Fire Detection and Its Application. 2017. Available online: <https://theses.lib.polyu.edu.hk/handle/200/9761> (accessed on 27 July 2021).
- Gaur, A.; Singh, A.; Kumar, A.; Kumar, A.; Kapoor, K. Video flame and smoke based fire detection algorithms: A literature review. *Fire Technol.* **2020**, *56*, 1943–1980. [CrossRef]
- Verstockt, S. *Multi-Modal Video Analysis For Early Fire Detection*; Faculty of Engineering and Architecture: Ghent, Belgium, 2011.
- Li, M.; Xu, W.; Xu, K.; Fan, J.; Hou, D. Review of fire detection technologies based on video image. *J. Theor. Appl. Inf. Technol.* **2013**, *49*, 700–707.
- Pan, H.; Badawi, D.; Cetin, A.E. Computationally efficient wildfire detection method using a deep convolutional network pruned via Fourier analysis. *Sensors* **2020**, *20*, 2891. [CrossRef] [PubMed]
- Shehzad, A.; Jalal, A.; Kim, K. Multi-person tracking in smart surveillance system for crowd counting and normal/abnormal events detection. In Proceedings of the 2019 International Conference on Applied and Engineering Mathematics (ICAEM), Taxila, Pakistan, 27–29 August 2019; pp. 163–168. [CrossRef]
- Gajjar, V.; Khandhediya, Y.; Gurnani, A. Human detection and tracking for video surveillance: A cognitive science approach. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2805–2809. [CrossRef]
- Syed Ameer Abbas, S.; Anitha, M.; Vinitha Jaini, X. Realization of multiple human head detection and direction movement using Raspberry Pi. In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22–24 March 2017; pp. 1160–1164. [CrossRef]
- Parthornratt, T.; Burapanonte, N.; Gunjarueg, W. People identification and counting system using raspberry Pi (AU-PiCC: Raspberry Pi customer counter). In Proceedings of the 2016 International Conference on Electronics, Information, and Communications (ICEIC), Danang, Vietnam, 27–30 January 2016; pp. 1–5.
- Singh, C.; Sohani, M. Estimation of Crowd Density by Counting Objects. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Tirunelveli, India, 11–12 May 2017; pp. 235–238. [CrossRef]
- Surasak, T.; Takahiro, I.; Cheng, C.H.; Wang, C.E.; Sheng, P.Y. Histogram of oriented gradients for human detection in video. In Proceedings of the 2018 5th International Conference on Business and Industrial Research: Smart Technology for Next Generation of Information, Engineering, Business and Social Science (ICBIR), Bangkok, Thailand, 17–18 May 2018.
- Bin Bahrudin, M.S.; Kassim, R.A.; Buniyamin, N. Development of Fire alarm system using Raspberry Pi and Arduino Uno2013. In Proceedings of the International Conference on Electrical, Electronics and System Engineering (ICEESE), Kuala Lumpur, Malaysia, 4–5 December 2013; pp. 43–48. [CrossRef]
- Bheemul, H.C.; Lu, G.; Yan, Y. Three-dimensional visualization and quantitative characterization of gaseous flames. *Meas. Sci. Technol.* **2002**, *13*, 1643. [CrossRef]
- Jiang, Q.; Wang, Q. Large space fire image processing of improving canny edge detector based on adaptive smoothing. In Proceedings of the 2010 International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering, Macao, China, 30–31 January 2010; pp. 264–267.
- Zhang, Z.; Zhao, J.; Zhang, D.; Qu, C.; Ke, Y.; Cai, B. Contour based forest fire detection using FFT and wavelet. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; Volume 1, pp. 760–763.
- Kim, D.; Wang, Y.-F. Smoke detection in video. In Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA, 31 March–2 April 2009; Volume 5, pp. 759–763.
- Han, D.; Lee, B. Flame and smoke detection method for early real-time detection of a tunnel fire. *Fire Saf. J.* **2009**, *44*, 951–961. [CrossRef]
- Ma, S.; Zhang, Y.; Xin, J.; Yi, Y.; Liu, D.; Liu, H. An early forest fire detection method based on unmanned aerial vehicle vision. In Proceedings of the 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 6344–6349.
- Premal, C.E.; Vinsley, S.S. Image processing based forest fire detection using YCbCr colour model. In Proceedings of the 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], Nagercoil, India, 20–21 March 2014; pp. 1229–1237.
- Vijaylaxmi, V.K.; Sajjan, S.C. Fire Detection using YCbCr Color Model. *IJRSET* **2016**, *2*, 698–701.
- Liu, Z.-G.; Yang, Y.; Ji, X.-H. Flame detection algorithm based on a saliency detection technique and the uniform local binary pattern in the YCbCr color space. *Signal Image Video Process.* **2016**, *10*, 277–284. [CrossRef]
- Sathyakala, G.; Kirthika, V.; Aishwarya, B. Computer Vision Based Fire Detection with a Video Alert System. In Proceedings of the 2018 International Conference on Communication and Signal Processing (ICCS), Chennai, India, 3–5 April 2018; pp. 725–727.
- Prathaban, T.; Thean, W.; Sazali, M.I.S.M. A vision-based home security system using OpenCV on Raspberry Pi 3. In Proceedings of the AIP Conference, Putrajaya, Malaysia, 24–25 June 2019; AIP Publishing LLC: Melville, NY, USA, 2019; Volume 2173, p. 020013. [CrossRef]

27. Kang, L.-W.; Wang, I.-S.; Chou, K.-L.; Chen, S.-Y.; Chang, C.-Y. Image-based real-time fire detection using deep learning with data augmentation for vision-based surveillance applications. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–4.
28. Khan, R.A.; Uddin, J.; Corraya, S.; Kim, J. Machine vision based indoor fire detection using static and dynamic features. *Int. J. Control Autom.* **2018**, *11*, 87–98.
29. Wu, H.; Wu, D.; Zhao, J. An intelligent fire detection approach through cameras based on computer vision methods. *Process Saf. Environ. Prot.* **2019**, *127*, 245–256. [[CrossRef](#)]
30. Xin, J.; Huang, C. Fire risk analysis of residential buildings based on scenario clusters and its application in fire risk management. *Fire Saf. J.* **2013**, *62*, 72–78. [[CrossRef](#)]
31. Xu, G.; Zhang, Y.; Zhang, Q.; Lin, G.; Wang, Z.; Jia, Y.; Wang, J. Video smoke detection based on deep saliency network. *Fire Saf. J.* **2019**, *105*, 277–285. [[CrossRef](#)]
32. Xu, G.; Zhang, Y.; Zhang, Q.; Lin, G.; Wang, J. Deep domain adaptation based video smoke detection using synthetic smoke images. *Fire Saf. J.* **2017**, *93*, 53–59. [[CrossRef](#)]
33. Kim, B.; Lee, J. A video-based fire detection using deep learning models. *Appl. Sci.* **2019**, *9*, 2862. [[CrossRef](#)]
34. Foggia, P.; Saggese, A.; Vento, M. Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1545–1556. [[CrossRef](#)]