A Video Image Based Privacy-Preserving Fire Detection System Using IOT

T.A. Sunilraj, M.Ranjithkumar, Y.A.Shaludash, M.Sarathkumar
PSN Institute of Technology and Science, Anna University, Chennai, Tamil Nadu, India

Abstract: Fire detection is the main objective besides surveillances. The aim of the project is to early detection of fire apart from preventive measures to reduce the losses due to hazardous fire. The project mainly is based on image and video processing using IOT. One of the recent solutions developed to detect fire is to use Internetof-Things (IoT) devices equipped with cameras for surveillance. The captured videos of surroundings may be processed by the IoT devices themselves or at the cloud. The latter case is required if the detection algorithm is computationally demanding. However, the use of cloud has a flaw. In fact, using the cloud could pose the threat of having the privacy of a place violated, either through hacking or unauthorized access to the footage of the place where the cloud is installed. In this paper, a fire detection system that preserves the privacy of surroundings, while maintaining a high level of accuracy for fire detection is proposed. The proposed system makes use of the cloud for fire detection; and that is achieved by sending to the cloud features extracted from the video captured by the IoT device, instead of sending the actual footage. Binary video descriptors and Convolution Neural Network (CNN) have been used to develop the fire detection algorithm. The video descriptors are used to extract features, while the CNN is used for classification. Videos with real fire and non-fire scenes have been used in this development. Results show that the performance of proposed fire detection algorithm can achieve 97.5% classification accuracy that outperforms the state-of-the art algorithms which make direct use of raw videos. Therefore, the proposed fire detector is as reliable as other available systems, with the advantage of having a privacy-preserving capability. It is also demonstrated that the proposed video descriptors can be implemented for real-time processing using an IoT device, Raspberry Pi 4 platforms, with an average processing speed of 100ms per frame, which well satisfies practical needs.
Keywords: Fire detection, image descriptors, features extraction, pattern recognition, convolution neural network, IoT devices.

1. INTRODUCTION

Fire detection is an attractive topic for researchers, due its significant damage to lives and property within a very short time. Also, fire detection systems are of an utmost importance to alert occupants about the threat, because such systems have a crucial role for safe life and environment, and it has a great impact on human being, organisms, environment, and ecosystem. For that reason, many studies have been conducted to find the best strategies for fire detection using different techniques for indoor and outdoor areas. Generally, fire is detected based on two methods; the first is the traditional fire alarms, and the second relies on vision sensors for fire detection [1]. Traditional methods for fire detection rely on sensors that require close proximity for activation, such as infrared and optical sensors [2]. For example, in [3], The associate editor coordinating the review of this manuscript and approving it for publication was Yanli Xu . the authors proposed a prototype for fire monitoring in rural environments. It collects data from several long range (LoRa) nodes of sensors that measure temperature, relative humidity, wind speed and CO_2 of the environment. These data are collected using Arduino

board to be sent and analyzed by The Things Network (TTN) server, so that the result can be graphically presented in a website. In [4], a prototype for fire detection was built. Conventional sensors data, like temperature and gas measurement at the moment a fire starts, was collected to be processed using Arduino board. The main problem with the traditional methods is the high rate of false alarms. In addition, traditional methods might fail to detect small or distant fires, and do not give information about the size and precise location of the fire. Therefore, their costs are a factor of importance, where a large number of sensors are required to cover a large area. In case of an alarm, these methods require human intervention to confirm the occurrence of a fire. The second approach for fire detection is based on vision sensors. These sensory devices carry cameras; therefore, they can record videos. They have the advantages of being capable of covering larger surveillance areas, providing detailed information about the size, location, and degree of the fire. Vision-based sensors have also fast response with affordable costs and eliminate the need for human intervention to confirm the presence of a fire. Internet-of-Thing (IoT) has become an important research topic in academia and industry because of its potential wide range of business opportunities in different fields such as sensing, automation, telecommunication, and others [5]. This technology is simply a network of Internet-enabled devices that can sense, communicate, and react to changes in their environment [5], enabling a plethora of smart services to help improving our daily interactions and overall quality of life. Currently, billions of IoT devices are exchanging data between themselves and/or other devices using the internet. The use of IoT devices in fire detection has been considered recently [4], [6], [7]. These devices have the ability to cover a wider range than the traditional methods and respond at a relatively faster time [1], [2]. Further, they can convey information about the nature and characteristic of fire [8]. In fact, IoT devices are often equipped with video cameras for fire detection.look once algorithm (YOLOv3) [11] and single shot multibox detector (SSD) [12], for real-time fire detection. In [13], the problem of decreasing false positive value for fire and smoke detection is addressed. The authors proposed faster region-based convolutional network (R-CNN) for region-based fire detection. The R-CNN was further supported by local spatial and temporal information like frame difference, color, similarity, wavelet transform, coefficient of variation, and mean square error (MSE), so that the value of false positive rate is significantly decreased. In [14], multi expert system was used for fire detection goal. The multi expert system was based on different contributors including color dispersion of region of interest, and similarity between consecutive frames and centroid motion. In [15], the authors used support vector machine, fed with a combination of three static features and four dynamic features, to recognize flame images. They evaluated different kernels that are traditionally used in support vector machines. In [6], the authors designed a system for fire detection in forests and outdoor areas around smart cities. They used wireless sensor network combined with unmanned aerial vehicle (UAV) images for fire detection. They used environmental sensors to collect temperature, humidity, light intensity, and smoke through IoT devices. The presence of fire is confirmed using image processing techniques of images collected by the UAV. In [16], The authors used UAV supported by optical and thermal camera besides a temperature sensor to localize the fire area in forest regions. Once the UAV receives an alert message with the position of the center of the fire from telecommunication tower, it calculates a list of waypoints (x,y,z) of the fire area based on several trajectories, in a respect to take-off position. In [17], the authors concluded various techniques reported in the literature for intelligent detection systems of fires in forest and different environments. However, the reported researches dealt with raw images and data, without consideration to the privacy of environment under investigation. Further approaches include decision rule-assisted methods, temporal and spatial wavelet analysis, multi-resolution 2D wavelet analysis, YUV and YCbCr color spaces, spatio-temporal features based on texture analysis, and deep learning architectures; see [1], [2], [8] and the references therein.The detection of fire from videos is not any easy task; it depends on scene complexity, quality of video frames, regularity of the light, etc. This work is concerned with the problem of providing some privacy with respect to fire detection using IOT devices. In particular, the main contributions of this paper are as follows:It proposes a cloud-based fire detection system that preserves the privacy of surroundings captured by IoT vision sensors; and that is by introducing the concept of sending video features to the cloud instead of sending the actual footage It evaluates the performance of binary video descriptors to detect fire in frames of databases recently utilized in reputable literature [1], [2] It proposes the use of machine learning to fuse the results of two binary video descriptors for the purpose of achieving better performance in classifying fire/non-fire video frames.
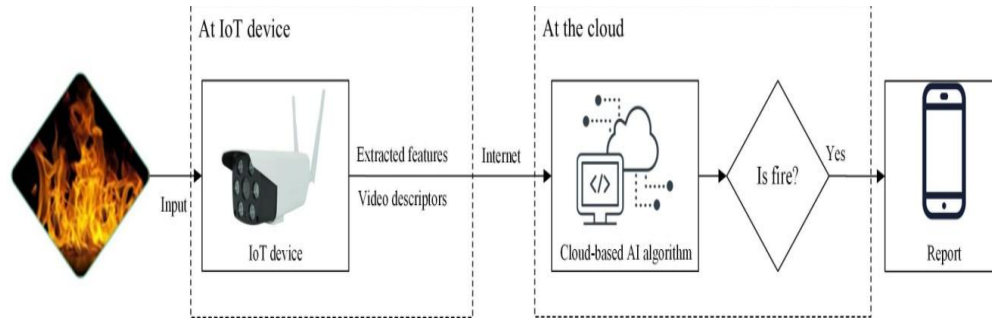
FIGURE 1. System model.

- ❖ It considers, for the first time, implementation of binary video descriptors on an IoT device (Raspberry Pi 4 platform) and provides measurements about the average processing speed
- ❖ .It shows that the use of binary video descriptors for fire detection can achieve performance very close to that of state-of-the art algorithms which make direct use of raw videos. Therefore, the proposed IoT-based fire detector is as reliable as other systems available in literature, with the advantage of having a privacy-preserving capability.
- ❖ The proposed systems will give the combine result at the output whether smoke and fire is present are not.
- ❖ The system performance can be improved with the use of optimal algorithms for detecting motion and area and extracting features of fires. The enhance system will performed well than the existing system in terms of detection rate.

The paper is organized as follows. Section 2 presents description for the proposed detection system. Section 3 provides information about the dataset used in the development of fire detection algorithm. Sections 4 and 5 introduce the binary video descriptors and convolutional neural network employed for features extraction and classification, respectively. Section 6 provides discussions on classification results obtained using actual fire/non-fire scenes. Section 7 is concerned with the estimation of processing time when the proposed video descriptors are executed on a commercially available IoT hardware platform. Section 8 provides concluding remarks.

2. SYSTEM DESCRIPTION

Development of an IoT-based fire detection system requires understanding different components pertaining to the system's software and hardware. In this section, we start first by presenting the main concept of proposed fire detection system. In fact, there are different ways to detect fire, but our objective here is to extract simple features from the video stream so that these features are sent to the cloud for fire detection and possibly alarms initiation. Among the possible choices to achieve such a task is to use video descriptors, which are computationally inexpensive. Video descriptors extract feature points in an image by finding corners, edges, blobs or any unique point in the image [20]. Using video descriptors will make it easier to extract fire features. There are various video descriptors available in the literature, but we have focused on seven well-known types. The selected descriptors are the scale-invariant feature transform (SIFT), oriented fast and rotated brief (ORB), binary robust invariant scalable key points (BRISK), speeded up robust features (SURF), Canny edge detector, Harris corner detector, and finally Shi-Tomasi corner detector. Once the outputs of video descriptors are sent to the cloud through the internet, machine and deep learning techniques can be used to perform the task of fire detection. Here, we choose to use the convolution neural network (CNN) for fire detection, due to its excellent performance in many closely related applications [1], [21]. Equipped with the video descriptors and machine/deep learning techniques, the proposed fire detection systems can be synthesized along the lines of concept presented above, as shown in Figure 1. The captured images first go through Area detection. If the mentioned condition is satisfied then Alert alarm is turned ON, if it not satisfied then the system keeps on extracting images.

3. DATASET

For the development of our system, we have used real videos to design our detection algorithm and evaluate its performance before implementation on hardware. In this work, we have used the dataset utilized in [8], which is provided, with thanks, from the authors. This dataset has an advantage of containing videos for many non-fire situations,

TABLE 1. Details of the dataset.

Video Name	Resolution	Number of Frames	Fire(F)/Non-Fire(N)	Notes
Fire 1	320*240	705	F	fire generated into a bucket and a person walking near it.
Fire 2	320*240	116	F	A fire very far form the camera generated into a bucket.
Fire 3	400*256	255	F	A big fire in a forest.
Fire 4	400*256	240	F	A big fire in a forest.
Fire5	400*256	195	F	A fire generated in a red ground.
Fire6	320*240	1200	F	A big fire in a forest.
Fire7	400*256	240	F	A big fire in a forest.
Fire8	400*256	240	F	A big fire in a forest.
Fire9	400*256	210	F	A big fire in a forest.
Fire10	400*256	210	F	A big fire in a forest.
Fire11	400*256	210	F	A big fire in a forest.
Fire12	400*256	1650	F	A big fire in a forest.
Fire13	320*240	1535	F	A fire in a bucket in indoor environment .
Fire14	320*240	240	F	Fire generated by a paper box. The video has been acquired by the authors near a street.
Fire15	320*240	900	N	Some smoke seen from a closed window. A red reflection of the sun appears on the glass.
Fire16	320*240	1725	N	Some smoke pot near a red dust bin.
Fire17	320*240	600	N	Some smoke on the ground near a moving vechicle and moving trees.
Fire18	352*288	630	N	Some far smoke on a hill.
Fire19	320*240	5958	N	Some smoke on a red ground.
Fire20	320*240	80	N	Some smoke on a hill with red buildings.
Fire21	720*480	22500	N	Some smoke far from the camera behind some moving trees.
Fire22	480*272	6097	N	Some smoke behind the mountain.
Fire23	720*576	342	N	Some smoke above a mountain.

Fire24	320*240	140	N	Some smoke in a room.
Fire25	352*288	847	N	Some smoke far from the camera in a city.
Fire26	720*576	1400	N	Some smoke in a room.
Fire27	320*240	6025	N	Some smoke on a red ground.
Fire28	720*576	600	N	Some far smoke on a hill.
Fire29	800*600	1920	N	A person moving in a lab holding a red ball.
Fire30	800*600	1485	N	A person moving in a lab with a red note book.

which resemble fire. Specifically, the dataset contains 31 videos, 14 videos contains fire and 17 are classified as non-fire videos. The 17 non-fire videos contain red objects, fog, smoke and clouds, which might be classified as a fire because of their color or motion. Using this dataset will help to design and test our detection algorithm since it has situations faced in real environments. Table 1 contains details about each video in this dataset. Figure 2 shows samples of the dataset.

4. VIDEO DESCRIPTORS

In this study, we consider binary descriptors for features extraction. These descriptors have attracted the attention of researchers because of their advantages.

FIGURE 2.Sample of the fire and non-fire frames.

of easy matching with those of other images. In particular, a Hamming distance in terms of bit count or XOR methods can be used to determine the degree of matching between two images, hence eliminating the need for the computationally more expensive Euclidean Distance or L2 norm. This, in turn, will greatlyincrease the matching speed. In what follows, we briefly introduce the proposed video descriptors, which will be used in a different signal processing setting. For more details about the binary video descriptors, the reader is referred to [18], [19] and the references therein. Harris corner detector is one of the earliest local descriptors. It defines a measure of corner quality η, which is the difference between the determinant and trace of a matrix defined in terms of the gradients of an image in the spatial dimensions, x and y. If η is positive, then this indicates the presence ofifies corners, Canny detector looks for edges. Shi-Tomasi corner detector, on the other hand, builds upon Harries detector by utilizing a slightly modified version of η. SIFT is considered a benchmark in the field, but computationally expensive as compared to other descriptors. It is based on the Difference of Gaussians (DoG) obtained by first convolving an image with a Gaussian kernel. The output of this step is then convolved with the transpose of the same Gaussian kernel, after which the difference between the outputs of convolution operations is obtained. This is to compensate for noise and non-significant variations in image appearance. In-plane rotation and illumination variations are considered in SIFT by taking the orientation and gradient magnitude of a given pixel. SIFT also builds upon the concept of image pyramids to take scaling into account. Specifically, it generates several different versions of the image at different scales, where each pixel of a given scale is a combination of all its neighbors at the higher scale. This is performed via bilinear interpolation. SURF is a closely related descriptor to SIFT. It is based on the same principles, but with details that are different for each step of the algorithm. For example, square-shaped filters are used to provide an approximation of Gaussian smoothing. It has an advantage in that it is several times faster than SIFT and claimed to be more robust than SIFT against different image transformations. ORB, on the other hand, makes use of two algorithms, Features from Accelerated Segment Test (FAST) and Binary Robust Independent Elementary Features (BRIEF) algorithms to create binary string for image description. The first algorithm is a computationally efficient method for locating corners. It performs a series of comparisons between a pixel and the pixels around it, which are located on a circle. If all surrounding pixels are of values higher or lower than that of the center pixel, then the center pixel is declared a corner. All corners declared by FAST are further filtered using Harris measure, and only the strongest corners or key-points are retained. FAST is also used to determine the orientation component of each key-point by finding the centroid of the image patch under consideration. This information is further utilized by the BRIEF algorithm, which performs a series of threshold tests pertaining to pixels' intensities of a given patch image. The tests are performed following a sampling pattern, and the result of each test is considered a single bit which is concatenated with the bits of other results to form a string or descriptor element.

As of BRISK algorithm, it relies on an optimized version of FAST, known as Adaptive and Generic Accelerated Segment Test (AGAST), to determine key points. The optimization is performed with respect to speed. Note that BRISK makes use of its own way to identify key-points. Specifically, it keeps the points that have the highest FAST score out of all their 8 neighbors. FAST score is the highest threshold that a point can have and still be considered a corner. BRISK implements a pyramid-based method to maintain invariance against scale variation [20], [22].In addition to the above binary descriptors, we consider Histograms of Oriented Gradient (HOG) [23], which is a gradient based descriptor that consists of local histograms of gradient orientations of image's grids. It works by dissecting the image to number of grids, then calculating the local 1-D histogram of gradient directions or edge orientations over the pixels of the given grid. The collected histograms of all grids are then normalized to form the HOG descriptor. In the above analysis, the flame is usually dynamic, and the shape is always changing. However, from the image have a certain degree of similarity, and the flame changes have a certain ranges.

5. CONVOLUTIONAL NEURAL NETWORK

The convolutional neural network (CNN) is considered in this work to perform the task of classification between two classes: Fire/Non-fire. It is a specialized artificial neural network used to process multidimensional input data [24]. In our development, the CNN splits frames (images) of a video descriptor for training and

testing. Training is accomplished by presenting images at the input and the corresponding labels at the output; e.g., fire or non-fire. The CNN learns the patterns of fire in video descriptors after training using large dataset of images. CNN consists of multiple layers. The first layer is a convolutional layer that extracts features from
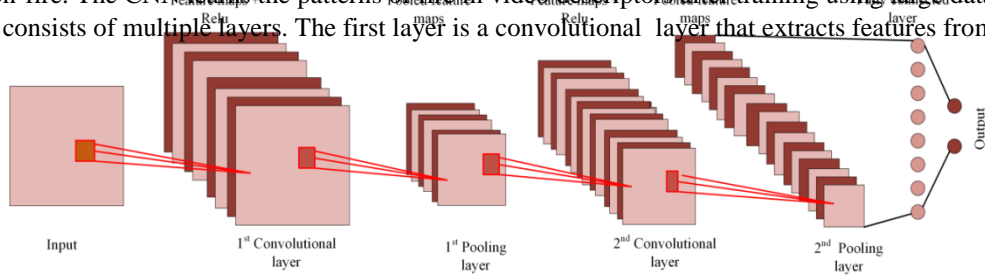

FIGURE 3. Layers of CNN

input images, using a set of trainable weights, to produce features maps [24]. The second layer is a pooling layer that reduces the number of parameters of features maps by selecting the maximum values of each small neighborhood [1]. The third and final layer is a fully connected layer, which provides high-level representation of the input data. In our work, the output of this layer gives the probability of the classes: fire and non-fire [1]. Rectified Linear Unit (ReLU) is an activation function used to retain only the positive part of the activation by setting the negative part to zero. The layers of CNN are shown in Figure 3.

6. EXPERIMENTAL RESULTS

6.1 Performance of Cnn Based Detection: In this section, we present the performance of our proposed fire detection algorithm, where raw videos and the CNN are used to act as a baseline model of fire classification. We have used all videos from the dataset. We selected 23216 frames. We used 70% of the data for training and 30% for testing. In the beginning, we converted all frames to gray scale images, and changed and a pooling layer with size 2×2 pixels. The output of the first pooling layer is applied to similar convolution and pooling layers. Another convolution layer is used with 64 filters with size 3×3 pixels and a pooling layer with size 2×2 pixels. A flatten layer is then used to convert featuresmaps into a vector. Finally, we used a fully connected layer with 64 neurons followed by another fully connected layer with 2 neurons. This last layer gives the probabilities of fire and non-fire cases.

TABLE 2. Ideas information.

Video type	Number of videos	Number of used frames	Total frames
Fire	14	10969	23216
No fire	17	12247	

TABLE 3. Comparisons of various fire detection methods.

Descriptor	Average accuracy yover 10 runs	Average loss over 10 runs
Baseline CNN model	94.84	17.4
BRISK	81.7	48.0
Canny	76.8	58.8
Harris	89.0	38.7
HOG	96.9	15.2
ORB	83.8	44.8

Shi-Tomasi	89.7	40.7
SIFT	73.0	60.6
SURF	75.2	60.8
Proposed model	97.5	

Table 2 provides information about the utilized videos. Table 3, on the other hand, gives the classification accuracy and loss, defined as follows. The accuracy is the number of correct classifications divided by the total number of classifications. The loss, on the other hand, is defined as the difference between the predicted label and the true label computed using the Categorical Cross Entropy (CCE) function, given by [25].

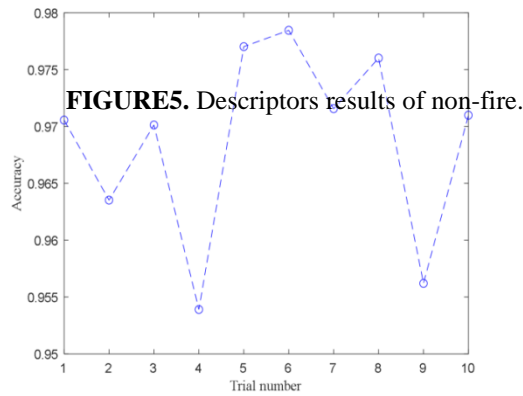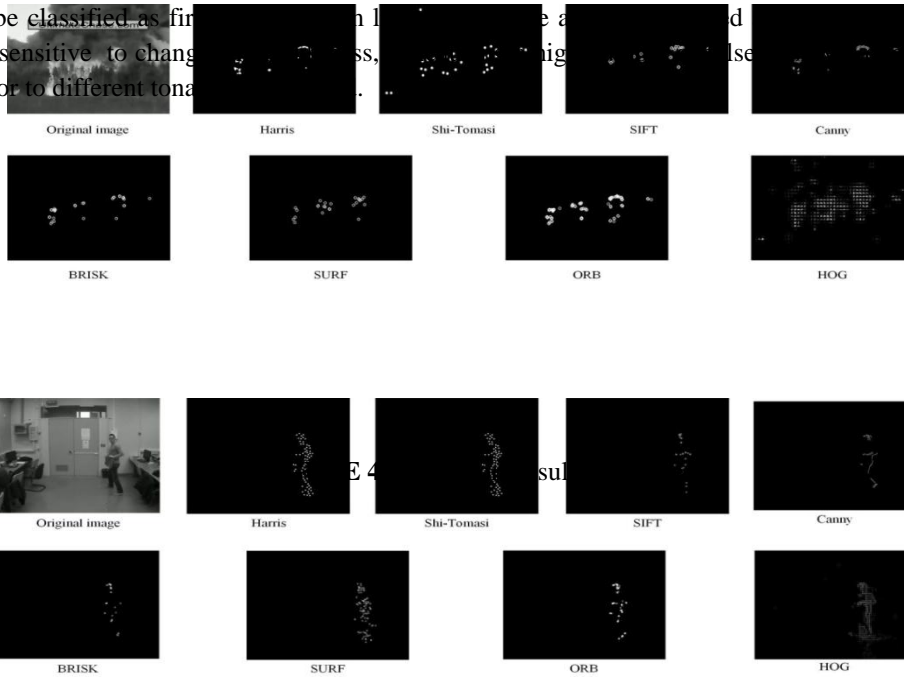$$CCE = -\frac{1}{N} \sum_{j=1}^{M} \sum_{i=1}^{N} (y_{ij} \log(y0_{ij})) \quad (1)$$

where y_{ij} is the true label of j th class of the i th sample, $y0_{ij}$ is the probability of predicted label of j th class of the i th sample, M is the number of classes and N is the number of samples. The proposed CNN achieved 94.84% and 17.4% in terms of accuracy and loss, respectively. This result is used to act as a benchmark to facilitate the evaluation of the system performance when video descriptors are being used.

6.2 Performance of Video Descriptors: In Section 4, we have introduced eight video descriptors. The objective here is to consider the performance of these descriptors and select the most suitable ones for our application. Towards that objective, we applied these descriptors tosome videos from the dataset by first subtracting the current frame of video from the average of the previous three frames. The reason is to eliminate any motionless objects such as a house. After that, the eight descriptors are applied to all frames and produced binary images, where '1s' denote the presence of feature points and '0s' denote their absence. Figure 4, and Figure 5 show samples of our results, where the original frames and corresponding video descriptors are provided. The output of each descriptor is fed to independent CNN model. We used 70% of the data for training and 30% for testing. The performance of each descriptor in testing phase is shown in Table 3, where we first report the results of baseline model. It is clear that the performance of HOG and Shi-Tomasi descriptors have shown superior results over other descriptors. Therefore, we selected HOG and Shi-Tomasi algorithms in the implementation of our proposed system for fire detection. Figures 6a, 6b, 6c and 6d show the performance of HOG and Shi-Tomasi descriptors in terms of accuracy and loss computed for 10 independent runs. It is of interest to note that the performance of each descriptor differs from that of the other at each trial. This observation has motivated us to modify our proposed fire detection system so that both descriptors are used, and their classification results are fused using a linear Support Vector Machine (SVM). The architecture of the new detection system is shown in Figure 7. In this new model, we used two different CNNs, each of which is trained according to the data obtained either from HOG or Shi-Tomasi descriptors. The last result in Table 3 reports the accuracy of this new system. Figure 8 shows the accuracy over 10 independent trials. The average of these trials is 97.5%; a value outperforms the state-ofthe-art published results depicted in Table 4, and also outperforms the result of our baseline model when raw videos are being used. Note that the authors of [26], cited in Table 4, exploited hand-crafted features and neural network for fire detection purpose. Static texture analysis features like Completed Robust Local Binary Pattern (CRLBP) and Gray Level Co-occurrence Matrix (GLCM) along with Gabor Transform and Discrete Wavelet Transform (DWT), in addition to dynamic features like 2D spatial wavelet decomposition in temporal domain and 3D Wavelet Decomposition, are fused and fed to extreme learning machine, which is a one hidden layer feed forward neural network. This approach is different from the baseline model, which only uses raw images with CNN model to detect the presence of fire in scene, which can compromise the privacy under monitoring. Moreover, it achieves the intended goal of the current work by preserving the privacy of the surrounding environment of area under surveillance. FIGURE 8. Accuracy of fused results for 10 independent trials. Flame detector are generally only used in high hazard areas, such as fuel loading platforms,industries areas and etc. Second, a novel descriptor based on a bag-of-words approach has been proposed for representing motion.

7. EXISTING METHOD

In general, the use of flame detectors is restricted to "No Smoking" areas or anywhere where highly flammable materials are stored or used. Existing method followed the rules for filtering fire pixels in the HIS color space. This simple idea inspires several recent methods: for instance, fire pixels are recognized by an advanced background subtraction technique and a statistical RGB color model: a set of images have been used and a

region of the color space has been experimentally identified, so that if a pixel belongs to this particular region, then it can be classified as fire. that they are particularly sensitive to change ss, hig lse to the presence of shadows or to different tona

FIGURE5. Descriptors results of non-fire.

Accuracy of HOG

Accuracy of Shi-Tomasi
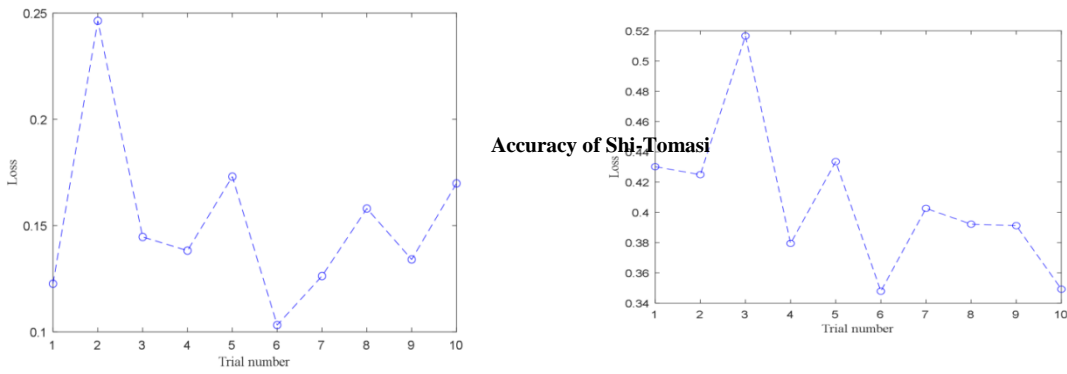
Loss of HOG

Loss of Shi-Tomasi
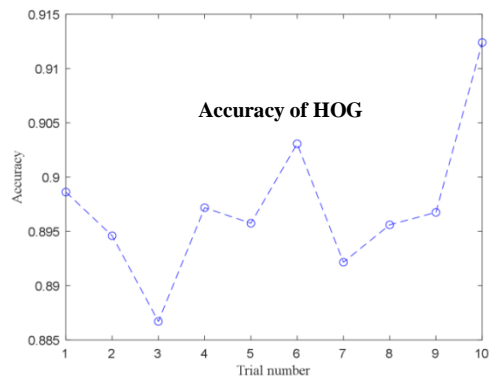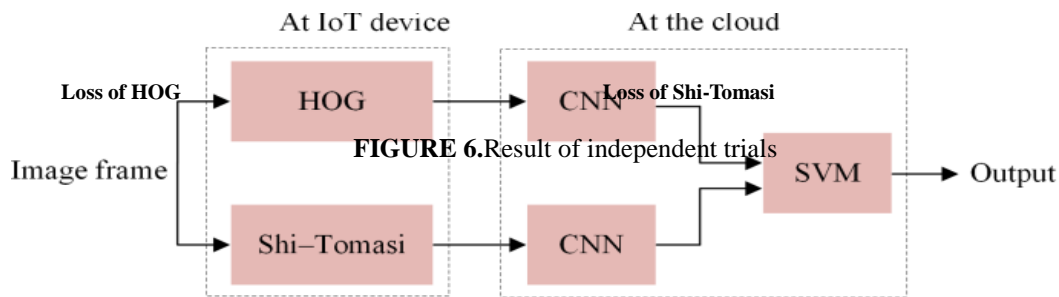
FIGURE 6. Result of independent trials

FIGURE 7. System architecture

8. HARDWARE IMPLEMENTATION

In this section, we consider the hardware realization pertaining to the extraction of video descriptors for the purpose of sending them to the cloud for further processing using advanced machine learning classification algorithms. Among the available hardware platforms, Raspberry Pi 4 is a good choice fulfilling the requirements of problem at hand. This platform packs cutting-edge features like USB-C power input, two video outputs that can each runs an external 4K monitor, and a choice of RAM size with 4GB maximum. The Raspberry Pi 4 is a tiny single-board computer, which means that all of its components, from the memory to the USB ports, fit on one PCB without add-on cards or accessories [28].
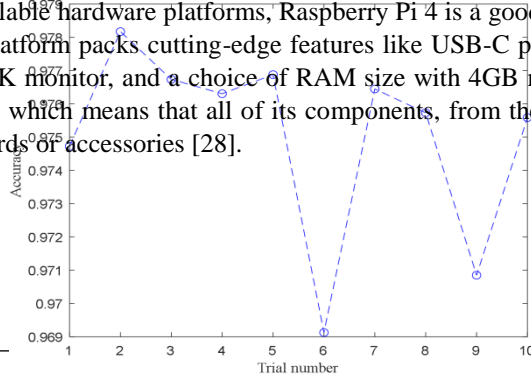
FIGURE 7. Accuracy of fused result for 10 indepent trials

The official operating system for the Raspberry Pi 4 is Rasbian, which comes with a Python IDE (Integrated Development Environment), called Thonny. Prior to the use of the Raspberry Pi, the video descriptors were built and developed using an IDE called Pycharm. However, we decided to switch to Thonny given its flexibility compared to Pycharm. n our development, a simple USB camera with 480p30 display resolution is integrated with the Raspberry Pi 4, as shown in Figure 9. The camera is attached through Port 4, while Ports 1,2, and 3 are for power supply, high definition multimedia interface (HDMI), and mouse and keyboard inputs, respectively. Note that a high quality, custom designed add-on board, 8-megapixel Sony IMX219 image sensor can also be used.
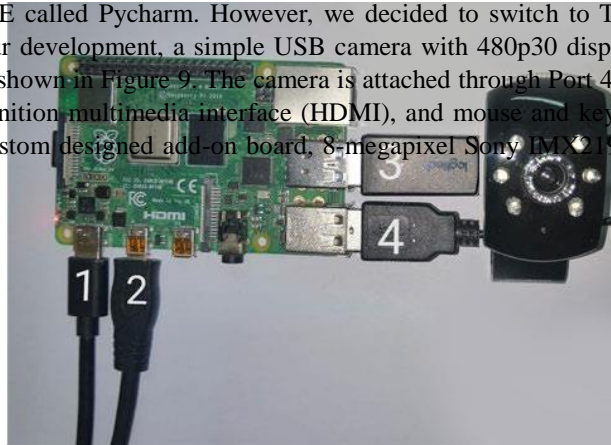

FIGURE 9. The Raspberry Pi 4 with camera attached to the main board.

This camera is featuring a fixed focus lens. It can capture 3280×2464-pixel static images, and also supports 1080p30, 720p60, and 480p60/90 videos. These names are shorthand for a family of display resolution, for example, 480p60/90 means that the number of vertical and horizontal pixels are 480 and 640, respectively. 'p' stands for progressive scan. The frames rate is the number given after 'p', in this example, it could be 60 or 90 frames per second. The camera is attached by way of one of the small sockets on the board upper surface, designed especially for interfacing to cameras. The board itself is tiny, of size around $25mm \times 23mm \times 9mm$. It also weighs just over 3g, making it perfect where size and weight are important. The developed video descriptors algorithms are downloaded into the Raspberry pi platform. The video frames are first captured by the camera and then directly processed to produce the key points of video descriptors. It has been found by several experimentations that the average processing speed to extract key points from a current frame is approximately 100ms. This speed is fast enough to satisfy the requirement for detecting fire in practical scenarios.

9. PROBLEM DEFINITION

The existing system uses only contrast based approach. It does not give efficient result. It takes long time identification and also the result is not accurate.

10. PROBLEM ANALYSIS

The purpose of the system Analysis is to produce the brief analysis task and also to establish complete information about the concept, behavior and other constraints such as performance measure and the system optimization. The goal of System Analysis is to be completely specify the technical details for the main concept in a concise and unambiguous manner. Our method does no use end-to-end training because the GAP within bounding boxes, and the majority voting processes are included.

11. CONCLUSION

Recent trend in the development of fire detection systems makes use of IoT devices often equipped with simple detection algorithms, having compromised accuracy and false alarm rate, to allow their real-time implementation over such resource-limited platforms. In this work, we proposed a privacy-preserving fire detection algorithm executable at the cloud. In this approach, an IoT device is first used to capture videos of the environment, extract features from the videos, and then send these features to the cloud for further processing to detect fire and possibly send warning messages to intended users. In this paper, we have used video descriptors to extract features. Specifically, seven different descriptors have been evaluated with respect to their capability to preserve features pertaining to the presence of a fire in videos. Among the seven descriptors, we have found HOG and Shi-Tomasi are of superior performance. These two descriptors have been considered with CNN classifier. Therefore, we have two fire detection architectures: HOG+CNN and Shi-Tomasi+CNN. These architectures have been evaluated using a dataset containing videos with real fire and no-fire scenes. The results show that HOG and Shi-Tomasi achieve accuracy of 96.9% and 89.7% respectively, when the CNN is employed as a classifier. Based on these results, we built a new system architecture where we fuse the output of the CNN utilizing the HOG descriptor with that utilizing the Shi-Tomasi descriptor. The fusion process is performed using linear SVM. The overall performance of the new system is 97.5%, which is outperforming the performance of state-of-the art systems. This makes the proposed fire detector is as reliable as other available systems, with the advantage of having a privacy-preserving capability. Further, a Raspberry Pi platform with 1.5GHz CPU, and 4GB RAM, and a simple camera has been used to implement the proposed video descriptors. Experimentations show that this IoT device is capable to extract features (key points) with an average processing performance of 100ms per frame, which well satisfies practical requirements.

REFERENCES

1. 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 Trans. Syst., Man, Cybern. Syst., vol. 49, no. 7, pp. 1419–1434, Jul. 2019.
2. K. Muhammad, S. Khan, M. Elhoseny, S. Hassan Ahmed, and S. Wook Baik, "Efficient fire detection for uncertain surveillance environment," IEEE Trans. Ind. Informat., vol. 15, no. 5, pp. 3113–3122, May 2019.
3. S. Sendra, L. García, J. Lloret, I. Bosch, and R. Vega-Rodríguez, "LoRaWAN network for fire monitoring in rural environments," Electronics, vol. 9, no. 3, p. 531, Mar. 2020.
4. G. Roque and V. S. Padilla, "LPWAN based IoT surveillance system for outdoor fire detection," IEEE Access, vol. 8, pp. 114900–114909, 2020.
5. P. S. Raju, M. Mahalingam, and R. A. Rajendran, "Review of intellectual video surveillance through Internet of Things," in The Cognitive Approach in Cloud Computing and Internet of Things Technologies for Surveillance Tracking Systems. New York, NY, USA: Academic, 2020, pp. 141–155.
6. P. Dasari, G. K. J. Reddy, and A. Gudipalli, "Forest fire detection using wireless sensor networks," Int. J. Smart Sens. Intell. Syst., vol. 13, no. 1, pp. 1–8, 2020.

7.   P. Foggia, A. Saggese, and M. Vento, ''Real-time fire detection for videosurveillance applications using a combination of experts based on color, shape, and motion,'' IEEE Trans. Circuits Syst. Video Technol., vol. 25, no. 9, pp. 1545–1556, Sep. 2015.

8.   S. Li, Q. Yan, and P. Liu, ''An efficient fire detection method based on multiscale feature extraction, implicit deep supervision and channel attention mechanism,'' IEEE Trans. Image Process., vol. 29, pp. 8467–8475, 2020.