FIRE DETECTION SYSTEM FOR WAREHOUSE ASSET PROTECTION

Prof. Vipul Lande[1], Mohd Murtuza Ateeque Sheikh[2], Parnika Shahu[3], Ayush Kabra[4], Akash Kumar Singh[5]

[1]Professor, Electronics and Communication, Shri Ramdeobaba College of Engineering and Management
[2]Student, Electronics and Communication, Shri Ramdeobaba College of Engineering and Management
[3]Student, Electronics and Communication, Shri Ramdeobaba College of Engineering and Management
[4]Student, Electronics and Communication, Shri Ramdeobaba College of Engineering and Management
[5]Student, Electronics and Communication, Shri Ramdeobaba College of Engineering and Management

ABSTRACT

This paper emphasizes the use of Haar cascade machine learning object detection for identifying the presence of fire in a warehouse area. A data set of numerous positive image (showing fire) and numerous negative images (not having fire) is trained and saved in the form of an XML file. We are using Raspberry Pi for running the program which is scripted in Python and utilizing the libraries of OpenCV and Numpy. The feed is taken from a remote surveillance IP camera in real time and sent to the Raspberry Pi where the algorithm breaks it into frames, each frame undergoes the classification and if presence of fire is detected then necessary flags are raised for further decision making. Raspberry Pi is used to act as a separate processing element and to avoid putting strain on the existing surveillance processing unit. We used both RPi3 and RPi4 to estimate various performance parameters as well as various types of cameras were used such as Pi-Cam, ESP32-Cam, offline videos from smart phones.

Keyword: - *Fire detection, Haar Cascade, Machine Learning, Raspberry Pi, OpenCV, Python, Numpy.*

1. INTRODUCTION

Fire is one of the most common problems that storage facilities, buildings, warehouses etc. face today. Therefore, a proper fire detection system is a necessity to avoid big disasters as well as for human health and security. The most commonly used fire detection systems include Smoke Detector systems and Temperature Sensors. However, these systems possess a major flaw in the sense that not every fire produces a significant amount of smoke. Also, not every smoke is originated from a fire. The possibility of sensors being damaged and not in a proper working condition can also be a major setback. In such a case, there will be a major loss caused to the property and human life. Therefore, these systems may not be as reliable as one may perceive.

Cases, like these, call for a much more reliable system which not only detects fire, but also makes the appropriate decisions for the actions to be taken against the fire. This method for fire detection is considerably cost effective and much more reliable. It also offers a faster response time as compared to the previously mentioned methods.

We used both, RPi3 and RPi4, to estimate various performance parameters. Various types of cameras were also used such as Pi-Cam, ESP32-Cam and offline videos from smart phones. Object detection using Haar Cascade Classifier is a machine learning approach where we are training a cascade function with the help of many positive and negative images. This particular object detection method was proposed in 2001 by Paul Viola and Michael Jones in their paper “Rapid Object Detection using a Boosted Cascade of Simple Features”. OpenCV is a library that contains numerous functions on image processing and machine learning. Out of these functions, we are using the functions concerned with Haar Cascade Classifier. Python allows a large number of libraries; it also offers flexibility to choose programming styles as per the developer’s choice.

2. OVERVIEW OF PLATFORM

Haar Cascade Classifier is a machine learning based approach to detect objects. The identification of objects is done using features of an image or a video, as was proposed by Paul Viola and Michael Jones in 2001, in their paper on "Rapid Object Detection using a Boosted Cascade of Simple Features". In this, a cascade function will be trained using a set of positive and negative images, and then used for object detection. Since we are using Haar Cascade Classifier for the purpose of detection of fire, we will first collect samples of positive and negative images for the training of the cascade. Here, positive images will be the images that will include fire, and negative images will be the images without fire. After the images are collected, we will use these images for training purpose of the classifier. Then we will have to extract the features from this.

Here, we will use the concept of integral images. We obtain an integral image by performing cumulative addition of each subsequent pixel intensity in horizontal and vertical axes. Integral images simplify the pixel-by-pixel calculation for feature extraction, thus, making the process faster. Out of all the extracted features, we identify the best features, i.e. features that will be most relevant in identifying the object, with the help of Adaboost. Adaboost is an algorithm used for improving performances. Basically, Adaboost gives more weight to the instances that are difficult to classify instead of focussing on those instances that can be easily handled. Therefore, we are able to identify the most relevant features. The fire detection can then be made in a more reliable and clear manner.

We will be using an IP camera for the purpose of fire detection. The use of an IP camera comes with certain advantages like manueverablity and portability. IP cameras can detect fire origin, magnitude and the area of spread of the fire more accurately. These cameras can also help in the detection of fire location. They require less cabling and when connected to a power adapter, they can be readily used anywhere merely by connecting to a switch board.

3. EXISTING SYSTEMS

3.1 Smoke Detectors

A smoke detector is one of the most commonly used fire detection system. Instead of identifying fire, it tends to sense smoke and assume that the source of that smoke is a fire, thus indicating the presence of fire. Although the system is said to be popular, the fact that it holds a few disadvantages cannot be overlooked. These disadvantages may include hypersensitivity (may detect a small smoke from a cigarette or from cooking and in turn, trigger an alarm for fire) and assumption of source (may assume that the smoke is generated from a fire even when not the case).

3.2 Temperature Sensors

Another popular fire detection system is the method of Temperature Sensing. In this method, a thermistor or a thermocouple is used for the detection of presence of fire. Although there are certain advantages to the system like: cost effectiveness, portability, versatility in terms of placement of the sensor etc., the disadvantages would also exist such as: less sensitivity, non-linearity, limited temperature range, inability to detect weather changes and high maintenance. Therefore, it may not be very reliable in serving the purpose of fire detection.

3.3 Flame Detectors

This particular system works of the principle of flame detection. These flame detection systems may or may not include a response system. Instead of security reasons, these systems are more likely to be used for industrial process confirmations. Basically, these provide an information to the operator whether a furnace is working properly or not; it also gives information on whether to turn on or turn off an ignition etc..

4. PROPOSED SYSTEM

4.1 Real time analysis with IP camera system :

The real time streaming of the area under observation are taken by low power IP cameras.

4.2 Feed access system :

The real time camera feed from IP camera is accessed by the Single Board Computer which acquires the feed frame by frame.

4.3 Haar Cascade Classifier based Object detection system :

The haar cascade classifier is used for detection of objects based on some knowledge . A collection of images containing fire are used to train and create a dataset, which is used by the haar cascade classifier to identify presence of fire from acquired frames from IP cameras.

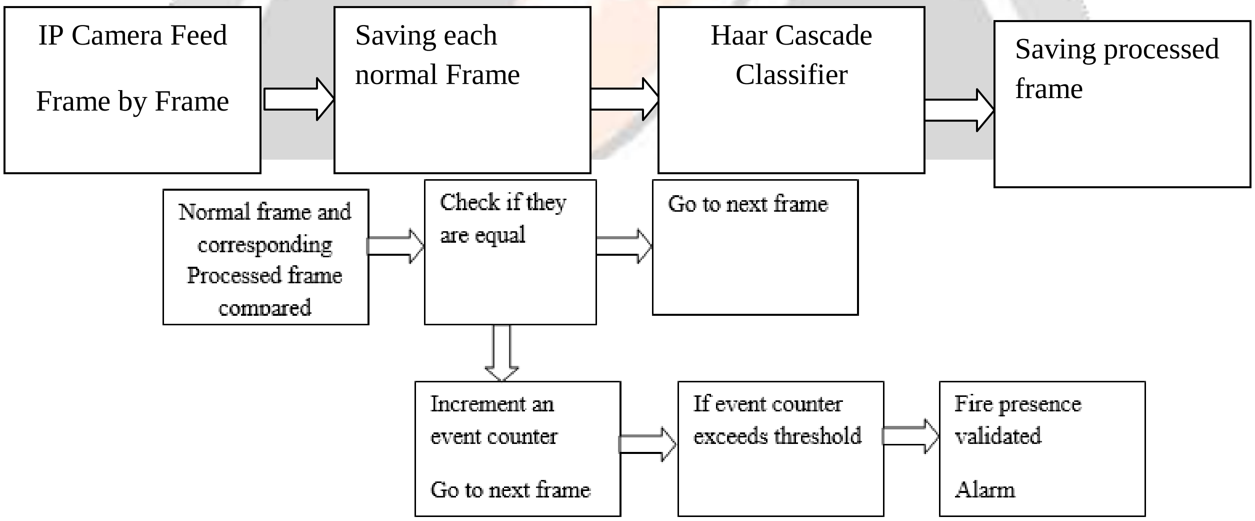
4.4 Alarming system :

The Haar Cascade Classifier detects presence of fire in some frames, if the number of frames in which presence of fire was detected exceed a certain number then the alarm will be raised.

5. DESIGN

There are 3 different tasks the fire detection system follows :

1. Accessing real time camera feed from IP cameras
2. Detecting presence of fire using haar cascade classifier
3. Reporting the situation to the concerned authority



5.1 Haar Cascade Classifier Dataset

To detect fire using our classifier, it requires a prior knowledge base to identify weather the object to be detected is fire. For this we train a dataset using images. Positive images and Negative images are used for training purpose where, Positive images contain the object to be detected (Fire in our system) and Negative images contain objects other than the one we want to detect. We train these images and create a dataset file in .xml extension which will be later used by the Haar Cascade Classifier.

5.2 Placing identifier on the fire & Validating presence of fire

After passing the acquired frame through the Haar Cascade Classifier a rectangular identifier is applied wherever the fire is. Each processed frame is saved in a separate file.

Fire might not be identified in some frames. For this we now compare the original frame and the corresponding processed frame every time. If they are equal then that means the classifier was not able to place an identifier on the fire, if they are not equal then that means the classifier placed an identifier on the fire in the frame which is making a difference between both the frames. If the number of frames in which the fire was detected exceed a certain threshold, then there is presence of fire.

5.3 Raising alarm

After successful validation of presence of fire an alarm will be raised for evacuation.

6. RESULTS

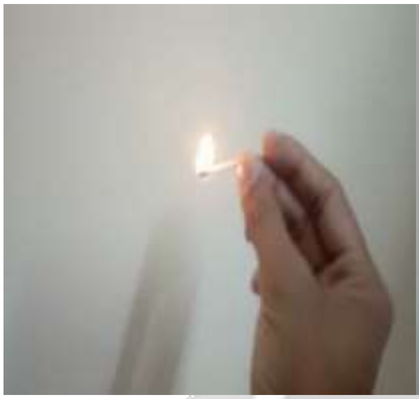


Fig 1: Orignal Frame

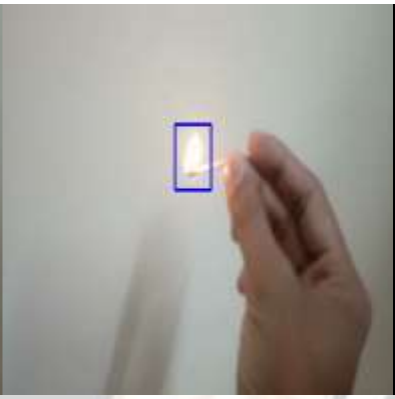


Fig 2: Processed Frame

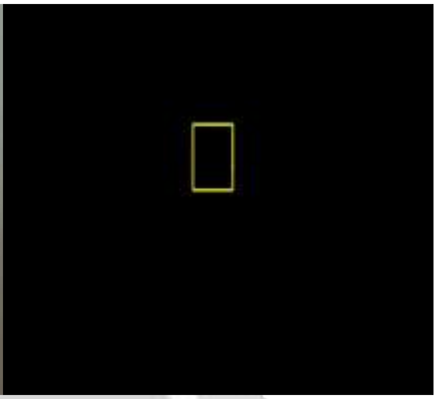


Fig 3: Difference Frame

6.1 Explanation of Haar Cascade Processed Frames

Fig1 shows an original frame from the input video stream from IP camera which undergoes processing and the resulting output is shown in the Fig2 where a rectangular identifier is placed where fire is. The Haar Cascade Object identifying object first recognizes weather there is fire or not, if there is fire it's attributes are matched with the dataset and the function returns the in-pixel co-ordinates of the rectangular identifier which is later placed using suitable OpenCV.

6.2 Explanation of Presence Estimation

Fig1 and Fig2 shows the original and processed frame, in many frames from the video stream even if there is fire the haar classifier fails to recognize presence of fire and hence fails to apply the rectangular identifier on the fire. If there is no identifier then both original and processed frame will be same. To overcome this we take difference of both the original frame and it's corresponding processed frame. We then count the number of non zero pixels, if there are non zero pixels then that means there is difference between the original and processed image due to the rectangular identifier which is shown in Fig3 and that signifies that fire is detected, we count this event for numerous frames. If the pixel count comes to zero then that means the classifier failed to add a rectangular identifier and so both the frames are same and hence their difference is going to be zero. The event counter that holds the value of events when haar classifier successfully placed an identifier over the fire present in frame and when the value of this counter exceeds a certain threshold value then presence of fire is validated and hence alarm is raised.

7. CONCLUSION

A fire detection is a system that detects the presence of fire and atmospheric changes relating to smoke. In this project we made an algorithm for Raspberry Pi to acquire feed from ESP32-CAM wirelessly. ESP-32 is a low cost, low power internet protocol camera module so it can easily be programmed and is very efficient to use in your project. We used image processing using OpenCV repositories and libraries in a Python environment for fire detection. Use of Haar cascade classifier for enhancing the detection of fire. Used grid view to find the exact location of fire. Implemented the algorithm to a multi camera setup of esp32 cam, which will be controlled wirelessly based on a master-slave communication.

Taking measures after fire detection which will include controlling sprinkler systems. The fire detector operates to alert people to evacuate a location in which a fire or smoke accumulation is present. The important of these is it gives sound from fire detection when fire attacks to industries, warehouses, etc. and a counter measure will take place to overcome the fire. It can be used to provide security at warehouses where storage and protection of various types of goods is main concern and works home security too. Those distinct sound exists to allow the notification to be heard. The fire alarm constructed by this project work is reliable at low cost.

8. FUTURE SCOPE

Companies have been finding ways to detect fire, gas, smoke more accurately.
If sensors are already present in the buildings, then it becomes easier for the fire extinguishing team to strategize the steps to be taken and handle the situation. This ensures safety too.
In the areas of forests which are very prone to fire during summer, drones are playing an important role in finding the hotspots for the fire. It has always been difficult to propose a novel and efficient communication strategy of sensors in the non-deterministic area.

The fact that at any time a certain joint in the connection can malfunction (due to an unknown/unexplained reason) cannot be ignored. Therefore, fault tolerance should be added to increase the productivity. Fire detectors are today more reliable than the past, but are being designed to minimize the false alarms. These false alarms are today the biggest challenge to the industry and it would not be surprising if at some point in the near future we see a tightening of standards. One of the issues is dust. Dust is enemy of fire detector as it can hinder the detection sensors visible area similar to this small particles, such as talcum powder can also be mistaken as smoke.

9. REFERENCES

1. Installation of OpenCV on Raspberry Pi:-
https://www.pyimagesearch.com/2016/04/18/install-guide-raspberry-pi-3-raspbian-jessie-opencv-3/

2. Study of ESP32 CAM , Programming , Knowledge of Arduino IDE
https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/

3. Study of Haar Cascade Classifier GUI Trainer
Training GUI Haar Cascade https://amin-ahmadi.com/cascade-trainer-gui/

4 Study of Haar Cascade Classifier Face Detection algorithm
https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

5.Study of Image Difference Evaluation
https://pysource.com/2018/07/19/check-if-two-images-are-equal-with-opencv-and-python/

6.Obtaining IP camera Feed in Raspberry Pi using python
https://www.pyimagesearch.com/2015/03/02/convert-url-to-image-with-python-and-opencv/

7. Study of basic OpenCV
https://opencv.org/