# Flame and Smoke detection in Surveillance Videos

**Rinky Dwivedi**
Associate Professor, Maharaja Surajmal Institute of Technolgy
rinkydwivedi@msit.in

***Abstract*— In the field of image processing, fire detection has gathered a lot of attention since the past decade. The advances in computer vision have led to many approaches for the vision-based systems to detect flame or smoke in surveillance footages [1]. Also, the approach of detecting flame and smoke using color-motion models has been investigated numerous times. However, such methods either slow and memory-exhausting, or are not accurate enough which restrict their implementation in real-time surveillance systems. In this research article, we propose an effective and accurate combination of flame and smoke detection for surveillance videos. The models for both are inspired from InceptionV2 architecture which offers reasonable complexity, while the combination of flame and smoke detections enables fire detection, even in cases where flame is not visible. According to the experiments with videos and real-time footage, the proposed algorithm detects fire in varied situations with reasonable accuracy. This algorithm would be especially suitable for areas with limited human intervention like forests, oil fields and the like.**

## I. Introduction

During the past few years, artificial intelligence and computer vision technology have evolved rapidly, enabling development of several applications in varied domains. In daily life, many unexpected occurrences can happen which can pose a threat to life and property. Early warnings can greatly minimize the impact of such occurrences, either by providing time to evacuate or by stopping the disaster before it becomes threat to life and property. Fire is the most common happening disaster and its early detection and warning during surveillance can help avoid huge fire disasters.

On an average, about 25000 fire related deaths occur every year in India, as per statistics by National Crime Records Bureau. Most of these are only due to home fires. One of the main causes of fire deaths can be delay in warning for fire. The traditional fire alarm systems need higher intensity fires or high proximity to the fire. Such systems fail to detect fire early when it is still controllable. Thus, a system is needed which would detect fire in its early stages so it can be easily managed, and huge fires can be prevented.

Recently, the need for fire detection systems using CCTV cameras has also increased in many other environments apart from homes and residential areas. The conventional thermal or smoke sensors work well for the detection of nearby and indoor fires, but they cannot be used to inform on remote situations. The most distinct advantage of camera-based systems is that they can be used outdoors for wildfire detection, whereas thermal and smoke sensors are mostly for indoor applications [4]. Early detection is especially important in areas like forests, as once a forest fire gains a certain scale it can be hardly controlled. Similar is the case with oil fields where apart from causing huge damage to property and environment, fire can lead to serious injuries and deaths of many people.

However, camera-based fire detection systems also have some disadvantages, such as sensitivity to fire-colored and blinking objects, and to some light sources, which frequently generate false positives. Thus, efficient and intelligent recognition algorithms are required for camera-based fire alarm systems in order to reduce false-positive alarms as well as missed-detection rates.

Thus, fire detection algorithms with less computational complexity, reduced false positives and higher accuracy are required. The existing fire detection methods in graphical content can be classified into two categories: flame detection and smoke detection. Most of the methods consider only one of the two, believing that one can compensate for the other. However, there are many cases were one of fire and smoke may be visible while the other may be not. In case of flames, there may be scenes where the flames are either too small or hidden out of sight. In case of smoke, there may be scenes where smoke is not visible clearly due to similar colored background. Here using both of flame and smoke detection is considered so that fire may be detected in most of the situations. However, because of the time and memory constraints, both of fire and smoke detections cannot be done at the same time. Thus, in this paper, flame detection is given more importance. Only in cases flame is not detected, smoke detection is done of the image/frame. This allows to detect most of the fires without a substantial increase in time and memory complexity.

The major contributions of this paper are summarized below:

1. Many existing methods were analyzed and algorithm for flame detection and smoke detection in real-time CCTV footage is designed considering the limitations of these methods. The time and energy consuming process of building detection models from scratch is avoided as transfer learning is used to create models from existing suitable models.

2. We trained and fine-tuned two models with architecture of InceptionV2 [15] with Single Shot Detection (SSD) [16] for flame and smoke detection. Both classification and localization

of fire and smoke has been done through transfer learning with carefully selected and modified datasets.

3. The proposed framework provides a balance between the accuracy and computational complexity of fire detection and detects fire in most cases due to combining of flame and smoke detection and using higher accuracy models. This would help prevent fire disasters in most cases.

## II. RELATED WORK

Extensive studies have been done to detect flames and smoke in smoke and videos. However very few of them consider both flame and smoke, choosing to focus on only one of the two.

Many works include the use of deep learning for flame and smoke detection. Muhammad et al. [1] used GoogleNet based CNN to detect fires in surveillance videos. Zhang et al. [2] used a Faster R-CNN model on synthetic smoke images to detect forest fire smoke. Frizzi et al. [6] used a nine layered convolutional neural network for identifying fire in videos. Hohberg et al. [5] utilized convolutional neural network to detect wildfire smoke. Zhang et al. [7] trained a joined deep convolutional neural network which operated in cascaded fashion for forest fire detection.

Another popular method to detect fire is based on analyzing color and motion properties of fire to determine patterns that would detect flame and smoke by searching for similar patterns in images and videos. Lee et al. [3] used the hue-saturation-value (HSV) color model and motion information to detect fire and smoke in videos. Chen et al. [8] used chromatic and disorder measurement in RGB space with growth and disorder dynamics to detect flame and smoke. Marbach et al. [9] used YUV color model to detect temporal variation of fire intensity and further combined features to verify flame regions. Another method was proposed by Toreyin et al. [10] by computing temporal and spatial wavelet transform of fire regions, along with determining irregularities of boundary of fire regions. Han et al. [11] compared the consecutive video frames and their color-motion features to detect flame and smoke in tunnels while reducing false positives from car lights and exhaust fumes. Celik et al. [12] used rule-based color model with YCbCr which effectively separates chrominance component from luminance component. The method not only can detect flame regions but can also separate high temperature regions from low temperature ones but is more suitable for large fires. Borges et al. [13] proposed a method which analyses consecutive frames for low level features such as color, area size, boundary roughness and skewness and combines these results using Bayes classifier to recognize fire in surveillance and newscast videos.

## III. THE PROPOSED FRAMEWORK

This section outlines the basic framework proposed in this paper.

In this paper, both flame and smoke detections are considered to detect fire in surveillance videos. Since, fire can cause huge damage to both life and property, it is necessary to increase accuracy so that no fire goes undetected. That is the main inspiration for using both flame and smoke models. But to balance the computational complexity, both flame and smoke detections are not done at the same time. Flame detection is given more importance. If flame is not present in the video frame only then the testing for smoke is done. This ensures that if either of flame and smoke is present in the frame, then fire is detected, and suitable action is taken.

An overview of the framework for fire detection in CCTV surveillance videos is shown in Figure 1. The models for flame and smoke detection are created using transfer learning which was achieved through pretrained models provided by Tensorflow Object Detection API. An input image or frame is checked for flames using flame detection model. If it returns false, then it is checked for smoke using smoke detection model. If this also returns false, then fire is not present in image. If either of the above conditions return true, then it means that fire is present in the image/frame and alarm is sounded.

### *A. Preparing Datasets*

The main datasets utilized for training of models are the FireSense database [17], FurgFire Dataset [18] and wildfire smoke detection dataset [19].

The **FireSense database** contains 11 videos containing flames along with 16 non flame videos for testing flame and 13 videos containing smoke with 9 non smoke videos for testing

**FurgFire** is a video dataset introduced in [18]. It contains 24 fire and non-fire videos.

**Wildfire smoke detection dataset** contains 4695 images composed from the video shot through video surveillance cameras in lookout towers and unmanned aerial vehicle (UAV).

Random frames were captured from video datasets and were saved as images. From image datasets, small amount images were selected to be included in final dataset. Apart from this some more images were downloaded from the Internet to construct the flame dataset with 242 images and the smoke dataset with 257 images.

The final models must return the position of flame or smoke in frame/image. After constructing the final datasets, images were annotated to indicate the boundary boxes of flame or smoke regions in the image. These positions would be used so that the models can return the boundary boxes of flame and smoke regions in any image/frame.

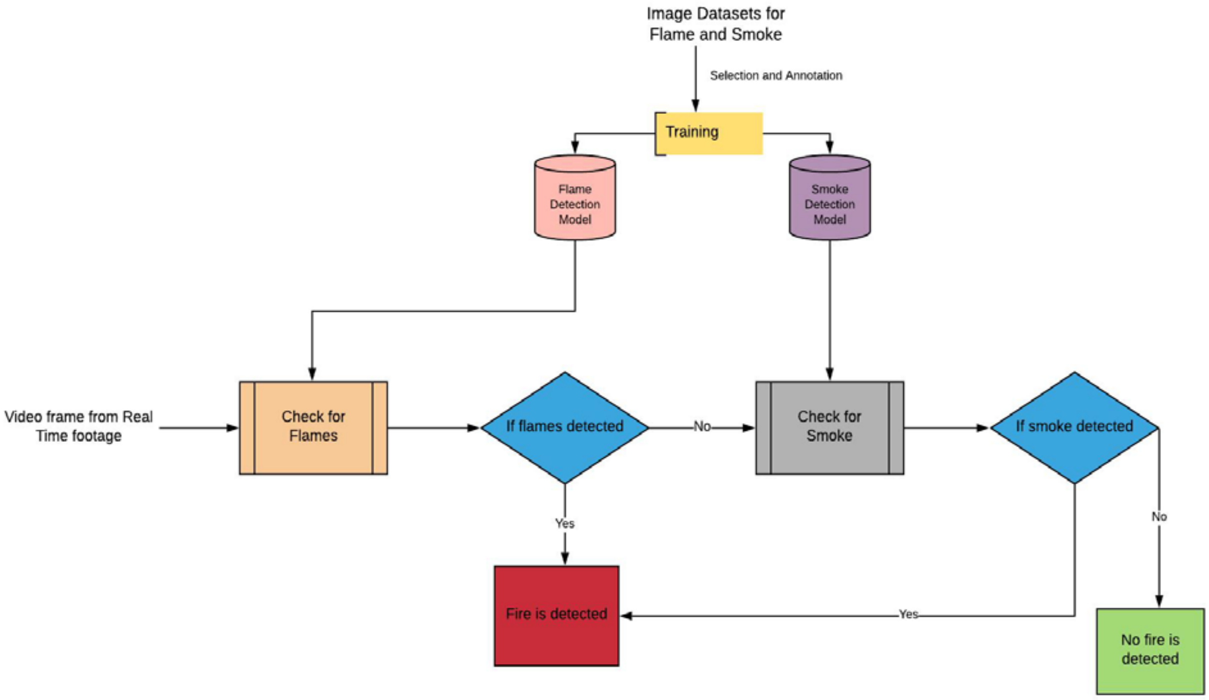


Fig. 1: The Proposed Framework

### B. *SSD Inception v2 COCO*

The TensorFlow Object Detection model used for flame and smoke detection is SSD Inception V2 COCO. As the name suggests, it uses SSD (Single Shot Detection) [15] to detect objects and is built on Inception V2 architecture [16]. It was pre-trained on COCO dataset. According to TensorFlow Object Detection API, it processes a 600×600 image in 42ms when tested on COCO dataset using Nvidia GeForce GTX TITAN X card [14].

SSD is suitable for real time object detection as it does not require Region Proposal Networks. The performance of SSD in object detection is measured by Total Loss which is combination of Localization and Confidence Loss[16].

Localization loss measures the difference between real and predicted position of the objects which is represented by bounding boxes. Confidence Loss is given as error in classifying objects, which deals with whether the object is present in the given area or not.

Total Loss is given by:

$$( , , , ) = \frac{1}{} ( \qquad ( , ) + \qquad ( , , )) \, [16]$$

where c is the class score, l is the predicted boundary box, g is the real bounding box, N is the number of positive match and α is the weight for the localization loss.

### C. *Creation of models with TensorFlow Object Detection API*

TensorFlow Object Detection API is a proven library for creating models that would detect any type of object in images or video frames. It provides several pretrained models for this purpose. Among these models SSD Inception V2 COCO was used for flame and smoke detection, as mentioned previously.

The basic outline of steps that were followed for training and creating detection models using this API are illustrated in Fig. 2 and are described below. These steps are followed separately for flame and smoke detection models

1. The annotated dataset was split into train (80%) and test (20%).
2. TensorFlow requires a label map, which maps each of the used labels to an integer value. This label map is used both by the training and detection processes.
3. The annotations were converted into the TensorFlow Record Format.
4. The pre-trained model SSD Inception V2 COCO and its configuration file were downloaded from TensorFlow Model Zoo.
5. The configuration file was modified to indicate the path of the downloaded pre-trained model, TFRecord files for train and test sets, the labels files. Number of Steps was changed to 50000 and Batch Size to 8.

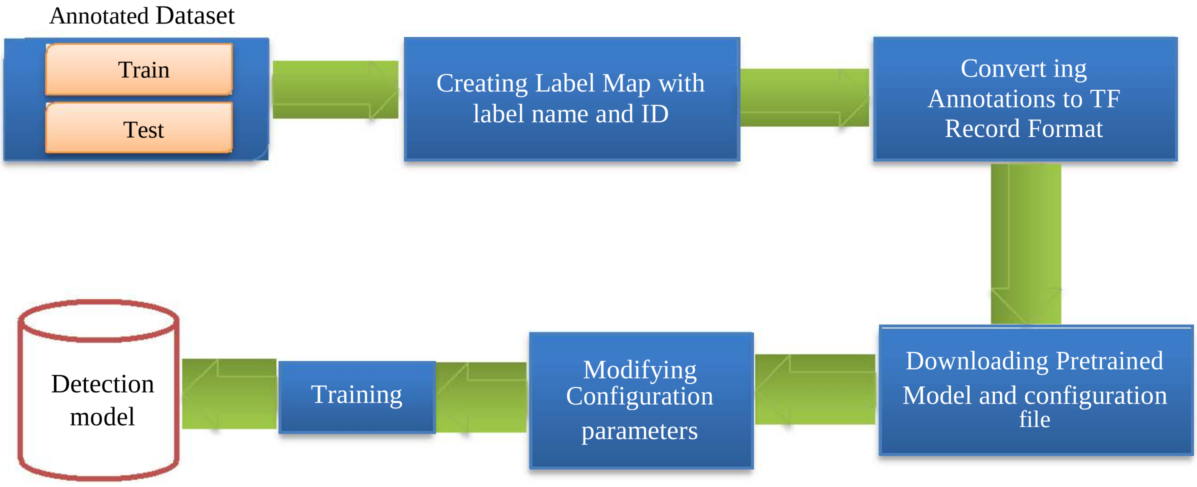

Annotated Dataset
Train
Test
Creating Label Map with label name and ID
Convert ing Annotations to TF Record Format
Downloading Pretrained Model and configuration file
Modifying Configuration parameters
Training
Detection model

Fig. 2: Overview of training process used to create flame/smoke model

6. Checkpoints were created by running the provided training script. The final checkpoint was used to create final model that would detect the respective object in image/frame.

These models are used in framework shown in Fig. 1.

.

### D. *Evaluation of the final models*

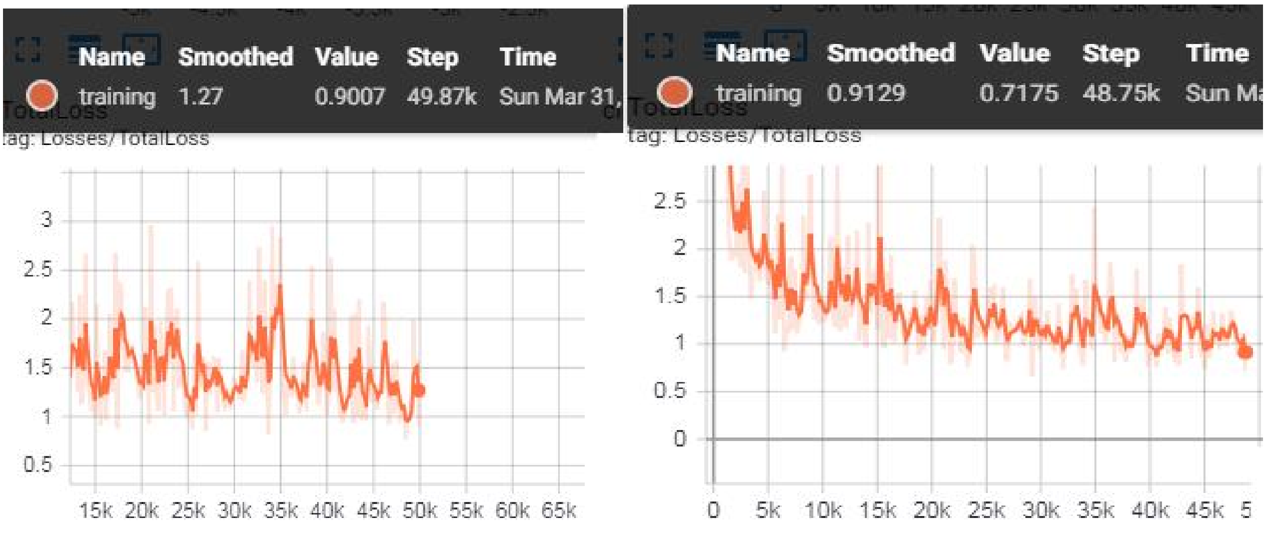
The training ran for 50,000 steps and the model created from the last checkpoint is selected as final model.

Total Loss for Flame Detection Model: **0.9007**

Total Loss for Smoke Detection Model: **0.7175**

The graphs showing variation of total loss over 50000 steps and final value loss are shown in Fig. 3. These were visualized using TensorBoard.

The objection detection training takes 25-26 seconds for each step in an environment without GPU. Thus, to complete training of 50000 steps, it would take about 14 days. Therefore, training for object detection in an environment without GPU is not feasible. Using a GPU environment brought down the time for each step to only 0.5-0.9 seconds, bringing the total time down to about 8-9 hours.



| Name | Smoothed | Value | Step | Time |
|---|---|---|---|---|
| training | 1.27 | 0.9007 | 49.87k | Sun Mar 31, |

| Name | Smoothed | Value | Step | Time |
|---|---|---|---|---|
| training | 0.9129 | 0.7175 | 48.75k | Sun Ma |

(a) (b)

Fig. 3: Graphs showing the total loss over 50000 steps and the final value of total loss in (a) Flame Detection and (b) Smoke Detection model

## IV. EXPERIMENTS

The model formed in the previous section were used for analysis and experiments mentioned in this section.

The models were tested on images and videos downloaded from the Internet. The models were also tested on real time video to check performance in real-time. The models were tested only on CPU. However, in an GPU enabled device the performance of the models would improve leading to higher processing rate.

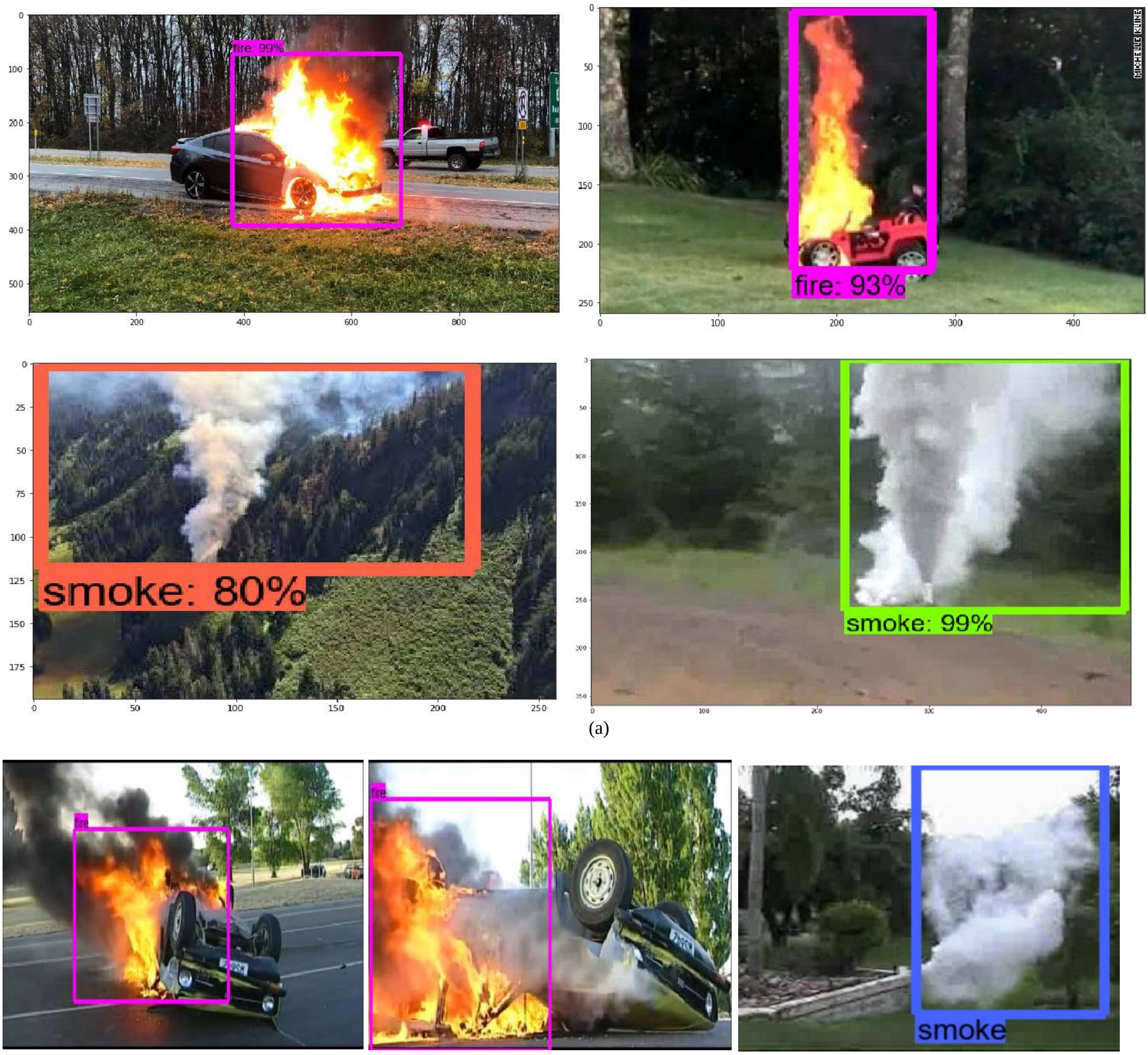
### A. *Testing models on images and videos*

The downloaded images were passed to the flame and smoke detection models. The models return the bounding boxes of flames / smoke and their confidence scores which are displayed on the image. By default, only bounding boxes with confidence score greater than 0.5 (50%) are displayed.

For videos, a similar procedure is used. A video can be considered to be a collection of frames. Therefore, for each frame, the steps mentioned above are followed. The frames with bounding boxes drawn are displayed. These frames are combined to form an output video.

While testing on videos, the frame processing rate is also checked as it would affect performance on real time videos. It is seen that when only flame detection is applied the frame processing rate is about 6-11 FPS. Which drops to 3-5 FPS when flame detection return False and smoke detection also must be applied.

Fig 4 shows testing of models on images and videos.



(a)

(b)

Fig. 4: Results of testing on (a) Images, (b) Video Frames

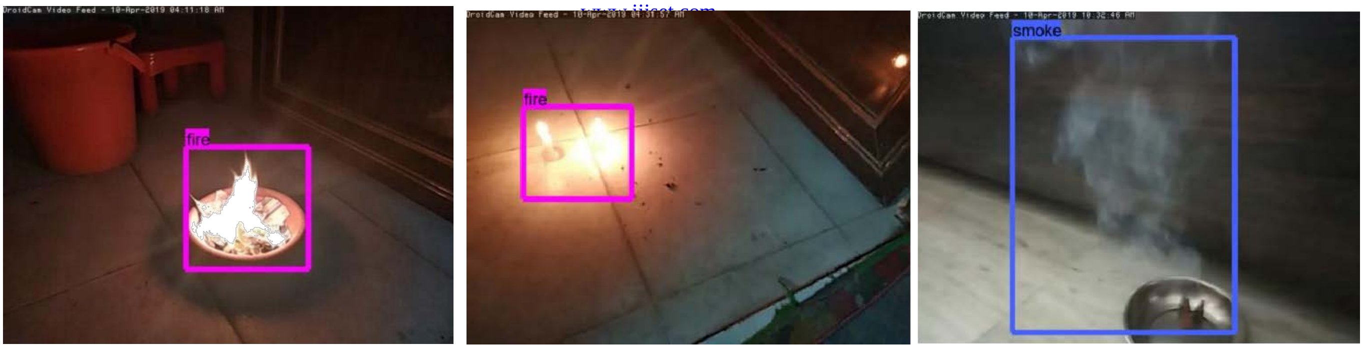

DroidCam Video Feed - 10-Apr-2019 04:11:18 AM
fire
DroidCam Video Feed - 10-Apr-2019 04:31:57 AM
fire
DroidCam Video Feed - 10-Apr-2019 10:32:46 AM
smoke

Fig. 5: Results of testing on real time footage

### B. *Testing on Real Time Footage*

The testing on real time footage was done by simulation of a simple surveillance camera by a smartphone camera using an application called DroidCam. DroidCam transfers feed from smartphone camera to a desktop computing device, effectively turning it into a simple surveillance camera.

The live camera feed was captured from the smartphone camera through DroidCam and processed simultaneously in the similar way the videos were processed earlier and output frames are combined to form output videos. Fig. 5 shows testing of flame and smoke models on real time footage.

## V. CONCLUSION

In this paper, we have a discussed a method for simultaneous detection of detection of fire through detection of flame and smoke in an image or video. This could have wide-ranging applications as elaborated in the above sections.

It is seen that the models for flame and smoke detection still provide some false positives i.e. wrong predictions. Thus, our future work will be towards increasing the accuracy of our models so that they provide more reliable predictions.

This is an era of smartphones and other portable smart devices. Thus, an application which could connect the user's smartphone or other devices to surveillance footage from his home/workplace and give urgent notification in case of fire detection would be suitable and could prevent many fire disasters. Thus, our future work in this direction would be developing such kind of application.

## ACKNOWLEDGMENT

This work was supported by the faculty and the students of the Department of Computer Science and Engineering, Maharaja Surajmal Institute of Technology, New Delhi.

## REFERENCES

[1] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho and S. W. Baik, "Convolutional Neural Networks Based Fire Detection in Surveillance Videos," in *IEEE Access*, vol. 6, pp. 18174-18183, 2018.

[2] Qi-xing Zhang, Gao-hua Lin, Yong-ming Zhang,Gao Xu and Jin-jun Wang, "Wildland Forest Fire Smoke Detection Based on Faster R-CNN using Synthetic Smoke Images", in *Proceedings of 8th International Conference on Fire Science and Fire Protection Engineering (ICFSFPE)*, pp. 441-446, 2017.

[3] Dae-Hyun Lee, Sang Hwa Lee, Taeuk Byun, and Nam Ik Cho, "Fire Detection using Color and Motion Models", in *IEIE Transactions on Smart Processing & Computing*, vol. 6, pp. 237-245, 2017

[4] A. V. Joshi, N. Hattiwale and H. D. Gadade, "Optimal Fire Detection Using Image Processing", in *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, pp. 248-252, 2017

[5] S. P. Hohberg, "Wildfire smoke detection using convolutional neural networks", in *Technical report, Freie Universitt Berlin, Berlin, Germany*, 2015..

[6] S. Frizzi, R. Kaabi, M. Bouchouicha, J. Ginoux, E. Moreau and F. Fnaiech, "Convolutional neural network for video fire and smoke detection", in *proceedings of IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, pp. 877-882, 2016.

[7] Q. Zhang , J. Xu and L. Xu, "Deep Convolutional Neural Networks for Forest Fire Detection", in *proceedings of International Forum on Management, Education and Information Technology Application*, 2016.

[8] T.H. Chen, P.H. Wu, and Y.C. Chiou, "An early fire-detection method based on image processing", in *Proceedings of International Conference on Image Processing (ICIP)*, pp. 1707-1710, 2004.

[9] G. Marbach, M. Loepfe, and T. Brupbacher, "An image processing technique for fire detection in video images", in *Fire safety journal, vol. 41*, pp. 285-289, 2006.

[10] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection", in *Pattern recognition letters, vol. 27*, pp. 49-58, 2006

[11] D. Han and B. Lee, "Development of early tunnel fire detection algorithm using the image processing," in *Proceedings of International Symposium on Visual Computing*, pp. 39-48, 2006.

[12] T. Celik and H. Demirel, "Fire detection in video sequences using a generic colour model", in *Fire Safety Journal, vol. 44*, pp. 147-158, 2009.

[13] P. V. K. Borges and E. Izquierdo, "A probabilistic approach for vision-based fire detection in videos", in *IEEE transactions on circuits and systems for video technology, vol. 20*, pp. 721-731, 2010.

[14] Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S and Murphy K, “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors”, *in proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3296-3297, 2017.

[15] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens and Zbigniew Wojna. “Rethinking the Inception Architecture for Computer Vision.”, in *proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , pp. 2818-2826, 2016.

[16] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu and Alexander C. Berg, “SSD: Single Shot MultiBox Detector.” , in *proceedings of European Conference on Computer Vision (ECCV)*, 2016

[17] Nikos Grammalidis, Kosmas Dimitropoulos, and Enis Cetin, FIRESENSE database of videos for flame and smoke detection [Data set]. *Zenodo*, 2017 .

[18] C. R. Steffens, R. N. Rodrigues and S. S. d. C. Botelho, "An Unconstrained Dataset for Non-Stationary Video Based Fire Detection," in *proceedings of 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR), Uberlandia*, pp. 25-30, 2015.

[19] Gao Xu, Yongming Zhang, Qixing Zhang, Gaohua Lin, Zhong Wang, Yang Jia and Jinjun Wang, “Video Smoke Detection Based on Deep Saliency Network”, in *Fire Safety Journal, vol. 105*, pp. 277-285, 2019.