

Video Based Fire and Smoke Detection Using Deep Learning

Tapas Saini¹, Dr. Govardhan Rao², Mannem Sri Nishma³

¹M.Tech Scholar, Computer Science & Engineering, University College of Engineering, Osmania University

²Assistant Professor, Department of CSE, University College of Engineering, Osmania University, Hyderabad, Telangana

³CDAC, Hyderabad

Abstract

The objective of this work is to design models for computer vision-based fire and smoke detection. CCTV cameras are installed at major public locations, homes, and buildings in every city. These cameras are often used for surveillance purposes and used mostly for forensic purposes. As a side advantage, these cameras can also be used for detection of fire and smoke in the camera field of view. IoT based sensors are the traditional way of detecting fire and smoke. Additionally, video streams from the camera can also be processed to detect fire and smoke events. This will enable additional security in cases when IoT sensors fail to detect a real fire and smoke event. It is proposed to use the advancements of AI/ML for computer vision-based fire and smoke detection.

Keywords: Computer Vision, Machine Learning, Deep Learning, Fire Detection, Smoke Detection

1. Introduction

Fire is one of the major disasters causing loss to life and properties around the globe. Therefore, it is crucial to develop a robust and reliable system for early detection of fire. Detecting fire at early stages will increase the chances of survival and reduce the loss to property. A report from the agency National Fire Protection Association (NFPA) shows that in United States of America (USA), the average number of fires per year are about 1.5 million approximately, with a high cost in terms of lives (with more than 3000 civilian fire deaths) and losses in economy—the cost for fire losses is estimated to be about 55 billion US Dollars (USD) per year REF . Hence, with the development of the Internet of Things (IoT) domain along with more recent Artificial Intelligence (AI) and the increasing need of safety in public places, an early fire-smoke detection system needs to be implemented for the benefit of all citizens. Under this work, a video or Visual based approach for the recognition of fire smoke is a promising method. In fact, the video has a wide view of the area under surveillance and often Closed-Circuit Tele Vision (CCTV) systems for surveillance purpose are already installed in buildings, in public places in various cities, and sometimes onboard passenger vehicles of public transport systems. Exploiting an already existing video infrastructure allows the reduction of purchase/installation costs of additional add-on products, increasing only the complexity of the algorithm used to detect smoke and fire.

2. Related Work

Fire Detection

Toreyin et al. (2006) [16] proposed a novel method to detect fire and/or flames in real-time through processing of the video frames data generated by an ordinary CCTV camera monitoring a scene. Along with ordinary motion and color clues, flames and fire flicker were detected by analyzing the video input in the wavelet domain. By using temporal wavelet transform periodic behavior in flame boundaries was detected. Color variations in flame regions were detected by computing the spatial wavelet transformation of moving fire-colored regions. Another trick which was used in the fire detection algorithm was the non-regularity of the boundary of the fire-colored region. All the above-mentioned clues were combined to reach a final decision. Experimental results showed that the proposed method was successful in detecting fire and/or flames. In addition, the method heavily reduced the false alarms on ordinary fire-colored moving objects in comparison to the methods using only motion and color clues.

Venugopal (2012) [17] proposed novel approach for forest fire detection using the image processing technique is proposed. A rule-based color model for fire pixel classification was used. The proposed algorithm used RGB and Y C b C r color space. One of the advantages of using Y C b C r color space is that it can separate the chrominance and luminance more effectively than RGB color space. The performance of the proposed algorithm was tested on two sets of images, one set contained fire and the other contained fire-like regions (but no fire). Standard methods were used for calculating the performance of the algorithm. As a result, the proposed method had both a higher detection rate and a lower false alarm rate. Since the algorithm was light in computation, in addition to city applications, it could also be used for real-time forest fire detection.

Kim and Lee (2019) [14] proposed a deep learning-based fire detection method using a video sequence, which imitates the human way of fire detection process. The proposed method used Faster Region-based Convolutional Neural Network (R-CNN) to detect the suspected regions of fire (SRoFs) and of non-fire based on their spatial features. Then, the features which are summarized within the bounding boxes in successive frames were accumulated by Long Short-Term Memory (LSTM) to classify whether there is a fire or not in a short-term period. The decisions for successive short-term periods were then combined in the majority voting for the final decision for the long-term period. In addition, the areas of both flame and smoke are calculated and their temporal changes were reported to interpret the dynamic fire behavior with the final fire decision. The experiments proved that that the proposed long-term video-based method may successfully improve the fire detection accuracy compared with the still image-based or short-term video-based method by reducing both the false detections and the misdetections.

Wang et al. (2022) [18] addressed the problem of limited number and modality of fire detection training datasets. They constructed a 100,000-level Flame and Smoke Detection Dataset (FASDD) based on multi-source heterogeneous flame and smoke images. According to them, FASDD was currently the most versatile and comprehensive dataset for fire detection. It provided a challenging benchmark to drive the continuous evolution of fire detection models. Additionally, they formulated a unified workflow for preprocessing, annotation and quality control of fire samples. Their extensive performance evaluations based on classical methods showed that most of the models trained on FASDD can achieve

satisfactory fire detection results, and especially YOLOv5x achieves nearly 80% mAP@0.5 accuracy on heterogeneous images spanning two domains of computer vision and remote sensing. And the application in wildfire location demonstrated that deep learning models trained on their dataset could be used in recognizing and monitoring forest fires. They proposed that their solution can be deployed simultaneously on watchtowers, drones and optical satellites to build a satellite-ground cooperative observation network, which can provide an important reference for large-scale fire suppression, victim escape, firefighter rescue and government decision-making.

Smoke Detection

Toreyin et al. (2008)[19], Aleksic (2004)[1], triggers an alarm within several minutes when combustion operation is producing flames and increasing the temperature of environment around. In Amer and Daoud (2007)[3], Amer and Daoud (2005)[2] a photo electric smoke detector with a real smoke chamber was combined with smoke temperature. Current standards such as EN50155 for onboard train safety recommend setting a delay of 60 seconds between the start of fire and its detection. Also, commercial solution uses point-based optical and temperature smoke detectors as per Corp (2020a)[5] for onboard train antifire systems. These sensors aim at alarming the presence of a fire with a time delay within 60 seconds. On the other hand, the system in Corp (2020a)[5] works on the principle that hot air produced by a fire moves from the bottom to the top of the train coach, and hence the smoke moves upwards toward the sensor placed on the roof of the train coach.

Gagliardi and Saponara (2020a) [10] proposed Advised, a novel video smoke detection algorithm for antifire surveillance systems, for detection in both outdoor and indoor application scenarios. To optimize installation costs, they considered a fixed single camera, working in the visible spectral range, already installed in a close circuit television system for surveillance purposes. They adopted multiple techniques which are Kalman based motion detection technique, image segmentation, color analysis, blob labeling, time/edge-based blob analysis, geometrical features analysis, and M of N decisor function, based on which the system was able to generate alarm signals with improvements in estimation performance compared with the state-of-art techniques.

In Corp (2020b)[6], fire flames were identified by measuring absolute temperature and its own gradient through an intelligent Infra-Red (IR) sensor. But the detection systems presented have the drawback of reacting slowly. Moreover, the approach needs active fans or air conditioning to speed up the process of smoke/fire detection, hence avoiding a very high measuring latency. The purpose of this work is to develop an innovative video-based smoke detection technique able to trigger the alarm within a few seconds. Such an algorithm has been already implemented into several IoT embedded devices to develop a distributed antifire system accessible via web browser and able to signal a fire alarm from different camera nodes, discussed in Gagliardi and Saponara (2020b)[11]. With this paper, the authors intend to extend that discussion by focusing on the smoke detection algorithm aspects.

Gagliardi et al. (2021) [9] presented a hybrid approach to use the fast and accurate identification of smoke in a video input. The algorithm combined a traditional feature detector based on Kalman filtering and motion detection and used a lightweight short convolutional neural network. The technique allowed the automatic selection of specific regions of interest within the image by the generation of bounding

boxes for black and white colored moving objects. In the final steps, the convolutional neural network verified verification of the true presence of smoke in the proposed regions of interest. The algorithm also provided an alarm generator that can trigger an alarm signal if the smoke is persistent in a time window of 3000 ms (about 3 seconds). The proposed technique was compared to the state-of-the-art methods available in literature by using several videos of public and nonpublic dataset showing an improvement in the metrics. Finally, the authors developed a portable solution in embedded systems domain and evaluated its performance for the Raspberry Pi 3 embedded board as well as Nvidia Jetson Nano board.

Zhang et al. (2022) [20] proposed a video based spatial-temporal convolutional neural network for fire smoke recognition. The model works by concatenating the motion features and appearance features, and later by a convolution layer to implement fusion of spatial and temporal feature. To reduce the influence of background of no-smoke, the authors used an attention module to capture salience features from the input image. Experiments on their own self-created dataset showed that the presented method is valid, which achieved a detection rate of 97.5% and accuracy rate of 96.8%.

3. Proposed Methodology

3.1 Data Collection

Fire Data Collection: Images of Fire have been downloaded from the Internet. These images were collected from Google Images search engine. The images consist of various fire incidents and accidents that happened all over the world. Following are the types of collected fire scenarios:

1. Indoor Fire
2. Outdoor Fire



Smoke Data Collection: Images of Fire and Smoke are also downloaded from the Internet. These images were collected from Google Images search engine and mostly part of any fire images. The smoke scenarios are mostly outdoor and are majorly of two colors:

1. White Smoke
2. Black Smoke



Figure 3.1.2: Sample Smoke images

3.2 Training Deep learning models for Fire and Smoke

- **Deep learning-based Approach:** We have used the Deep learning-based approach for fire and smoke detection through Video/Image input as it has following advantages:
 1. It does not require any additional hardware sensor.
 2. RAW data in the form of Images of fire & smoke are considered. No feature engineering is required.
 3. It may reuse the existing video feeds available through CCTV cameras
 4. It can work for longer distances for fire and smoke detection.
 5. Video data from CCTV cameras can be effectively utilized

- **YOLO for Object Detection:** YOLO stands for "You Only Look Once," and is a popular object detection algorithm. It is known for its real-time processing speed and higher accuracy. YOLO works by solving the object detection problem as a regression problem and performs the detection of multiple objects within an image. The YOLO approach divides the input image into a grid of cells and predicts bounding boxes and respective class probabilities for each cell. Each of the grid cells is responsible for predicting a fixed number of bounding boxes, regardless of the number of objects present in that cell.

For example, if there are five bounding box predictions per cell, YOLO will predict five potential objects. The bounding box prediction consists of four parameters: the x and y coordinates of the center of the box, the width, and the height of the box. Technically, these values are predicted relative to the size of the grid cell. The class probabilities indicate the likelihood of each object class being present in the box.

During training, YOLO uses a labeled dataset to learn the prediction of the bounding boxes and class probabilities. It works by optimizing the model by minimizing the sum of the squared differences between the predicted and ground-truth bounding box coordinates and class probabilities. Specifically, in terms of architecture, YOLO typically uses a deep Convolutional Neural Network (CNN) as its backbone. This CNN processes the entire image to extract relevant features. The feature maps are further fed into additional layers to evaluate the final predictions for each grid cell. One of the advantages of YOLO algorithms is its speed. As it processes the entire image in one pass, it can achieve real-time object detection on a standard Graphical Processing Unit (GPU). But this speed brings a limitation as smaller objects become harder to detect accurately. Over the years, YOLO has undergone several improvements in architecture, resulting in different versions such as YOLOv2, YOLOv3, and YOLOv4 and the latest being YOLOv8. These versions introduced various enhancements to improve detection accuracy, handle different object scales, and address other limitations present in the original YOLO algorithm.

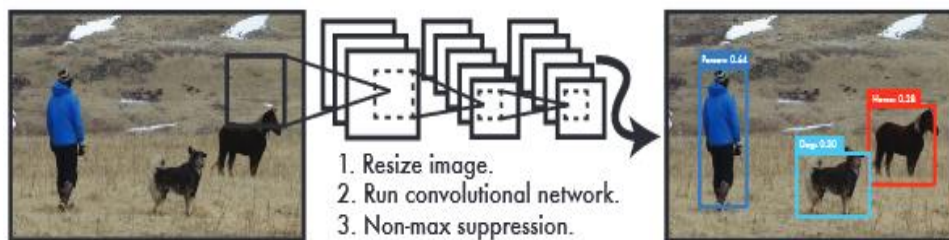


Figure 3.2.1: The YOLO Detection System

Figure 3.2.2 shows YOLO detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

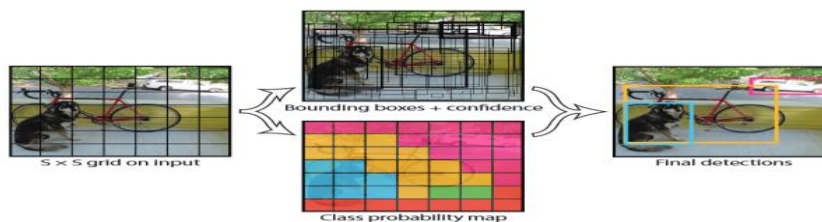


Figure 3.2.2: Generic YOLO Approach



Figure 3.2.3: YOLO model approach for fire detection

• **YOLO for Fire & Smoke Detection:** Instead of formulating the problem as classification of Fire & smoke in video or images, we went one step ahead and formulated as classification as well as localization for the fire and smoke detection problem. For this purpose, we choose YOLO Redmon et al. (2016) model-based approach due to its popularity and speed. A brief description of You Only Look Once (YOLO) is as below:

Figure 3.2.3 shows that input image having fire split into $S \times S$ grid cell, each grid predicts the bounding boxes and the confidence scores and finally, the score encodes the probability with enclosed bounding box on the detected fire (object).

4. DATA SETS

4.1 Data sets for Fire and Smoke

S.No	Dataset	Type	Description
1.	Foggia et al. (2015)[8]	Fire	14 fire videos and 17 videos without fire
2.	Chino et al. (2015)[4]	Fire	119 fire images and 107 images without fire
3.	Jadon et al. (2019)[12]	Fire	160 non-fire images, 46 fire videos, and 16 non-fire videos
4.	Fire-Flame-Dataset Olayemi Abimbola (2019)[15]	Fire & Smoke	Dataset containing Fire, smoke and Neutral images of both Training set and test set
5.	Own dataset	Fire & Smoke	Collected more than 1400 images of fire and smoke

Table 4.1: Video/Image Datasets for Fire Detection

Foggia et al. (2015) [8] proposed a method which had been tested on a very large dataset of fire videos acquired both in real environments and from the web. The dataset is composed by two main parts: the first 14 videos characterized by the presence of fire and the last 17 videos which do not contain fires; in particular, the second part is characterized by objects or situations which can be wrongly classified as containing fire: a scene containing red objects may be misclassified by color based approaches, while a mountain with smoke, fog or clouds may be misclassified by motion based approaches. Such composition allows us to stress the system and to test it in several conditions which may happen in real environments.

Chino et al. (2015) [4] consists of 226 images with various resolutions. Dataset was divided into two categories: 119 images containing fire, and 107 images without fire. The fire images consist of emergency situations with different fire incidents, such as buildings on fire, industrial fire, car accidents, and riots. These images were manually cropped by human experts. The remaining images consist of emergency situations with no visible fire and images with fire-like regions, such as sunsets, and red or yellow objects.

Jadon et al. (2019) [12] To evaluate the efficiency of the proposed approach, we tested it also with Datasetv2 Jadon et al. (2019). The test bench consisted of 160 non-fire images, 46 fire videos, and 16 non-fire videos. The dataset is limited, but it is challenging, e.g., it consists of videos for no-fire/smoke,

which contains sunset light, see Figure 4.1. Such a dataset was used for Jadon et al. (2019) [12] method, also based on deep learning approach.



Figure 4.1: Sample examples of FireNet Dataset_v2, which shows sunset light in non-fire videos

5. EVALUATION METRICS

5.1 Evaluation of Metrics and reporting results

We have used the standard evaluation metrics for determining models' performance.

Following are the common metrics considered for model evaluations:

Accuracy: Accuracy is the ratio of correctly classified items and total items. Accuracy can be a misleading metric for imbalanced data sets. Consider a sample with 97 negative and 3 positive values. Classifying all values as negative in this case gives 0.97 accuracy score. In this case Recall and/or Precision are more suitable metrics.

Recall: Recall is also referred to as the true positive rate or sensitivity.

Precision: Precision is also referred to as positive predictive value (PPV)

F1-Score: The F1 score is the harmonic mean of the precision and recall. It hence symmetrically represents both precision and recall in one metric.

Area Under Curve (AUC): It captures the area under the curve and compares the relation with the True Positive Rate (TPR) and False Positive Rate (FPR). It is used when it is about the classification problem with probabilities.

In context of fire, we consider Positive as Fire and Negative as No-Fire. Similarly in the context of smoke, we consider Positive as Smoke and Negative as No-Smoke. In the context of Fire detection:

- True Positive:** Model predicted Positive on input (say image) and Actually Positive
- False Positive:** Model predicted Positive and Actually Negative
- False Negative:** Model predicted Negative and Actually Positive
- True Negative:** Model predicted Negative and Actually Negative

The following are the metrics we will use for evaluation of our models for Smoke & Fire detection.

$$Accuracy = \frac{(True\ Positive + True\ Negative)}{(True\ Positive + False\ Positive + False\ Negative + True\ Negative)} \quad (5.1)$$

Accuracy (equation 5.1) is the main metric of evaluation of any ML model. For any skewed distribution, accuracy may not be the best metric. In this scenario, Precision (equation 5.2) and Recall (equation 5.3) will be suitable measures of metric.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (5.2)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5.3)$$

F1 score (equation 5.4) is the harmonic mean of Precision and Recall. Having F1score as 1 means we have achieved the best model.

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5.4}$$

ROC AUC captures the area under the curve and compares the relation with the True Positive Rate (TPR) and False Positive Rate (FPR).

True positive rate (TPR) is also called Recall or sensitivity (SEN). False positive rate (FPR) is also called the probability of false alarm. TPR is defined as per equation 5.5 and FPR is defined in equation 5.6

$$TPR = \frac{TruePositive}{TruePositive + FalseNegative} \tag{5.5}$$

$$FPR = \frac{FalsePositive}{FalsePositive + TrueNegative} \tag{5.6}$$

ROC AUC example is shown in Figure 5.1. It is used for classification problems with probabilities of result as decision makers. AUC compares two models as well as evaluates the same model's performance across different thresholds. ROC AUC is calculated by plotting TPR versus FPR at different thresholds.

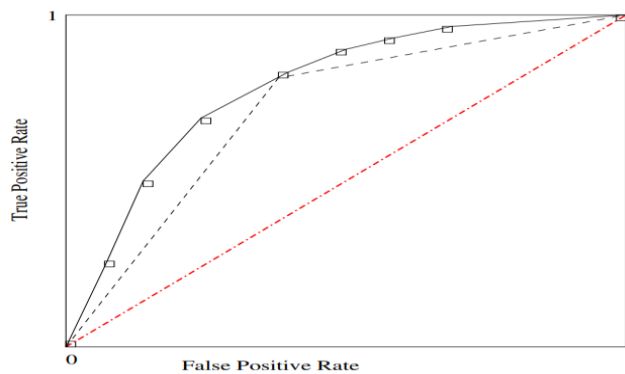


Figure 5.1: ROC AUC Curve

6. EXPERIMENTAL ANALYSIS

Considering the success of YOLO in object detection, we tried to use YOLO model and trained for Fire and an objection system. We used the base implementation of YOLOv5 Jocher (2020) [13] model.

We re-used the implementation from <https://github.com/ultralytics/yolov5> for training a testing the fire and smoke detection. The above link provides many variants of YOLO models. Figure 6.1 shows variants of YOLOV5 models along with their size, inference time and accuracies.

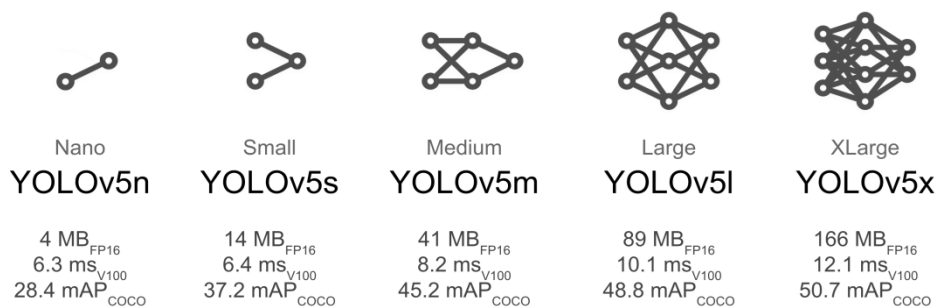


Figure 6.1: Pretrained Variants of YOLO Models

6.1 Dataset Creation for YOLO Training

YOLOv5 models must be trained on labeled data in order to learn classes of objects in that data. In our case the objects are fire and smoke blobs present in the view of image or video.

6.1 Create dataset. yaml: dataset, yaml and config file are created for fire and smoke detection model training that defines

1. the dataset root directory path and relative paths to train / val / test image directories (or *.txt files with image paths)
2. a class names dictionary:

```

1 path: ../datasets/fire_dataset/fire_and_smoke/train_data
2 # dataset root dir
3
4 train: images/train # train images (relative to 'path')
5 val: images/train # val images (relative to 'path')
6
7 # class names
8 names:
9   0: fire
10  1: smoke
11

```

Listing 6.1: dataset. yaml example

6.1.2 Annotations per image file

Annotations were created for all image files using two annotation tools:

1. MAKESENSE.AI

The website with link: <https://www.makesense.ai> was initially used for bounding box creations. But due to unavailability of saving of progress, the team moved to a new tool CVAT.

2. CVAT

<https://www.cvat.ai> is industrial grade image annotation tool providing the features of bounding box creation as well as saving of the progress. The CVAT tool was used for annotation of fire and smoke images. The bounding boxes are exported in a CSV text file of YOLO format. The sample CSV text file is shown as per in Table 6.1

0	0.366524	0.631974	0.075536	0.074678
1	0.469121	0.193587	0.938242	0.366983
1	0.207838	0.464371	0.415676	0.153207
1	0.663895	0.437648	0.358670	0.121140
1	0.972684	0.160036	0.049881	0.320071
1	0.085511	0.592637	0.166271	0.096200
1	0.276722	0.564133	0.206651	0.032067

Table 6.1: Sample annotations from a text file

The exported labels are in YOLO format, with one *.txt file per image (if no objects in image, no *.txt file is required). The *.txt file specifications are shown in table are

1. One row per object
2. Each row is class x_center y_center width height format.
3. Box coordinates must be in normalized xywh format (from 0 - 1). If your boxes are in pixels, divide x_center and width by image width, and y_center and height by image height.
4. Class numbers are zero-indexed (start from 0).

6.1.3 Training

For the purpose for Batch enabled training on Dataset, we used batch size(images) of 16 due to limitation in resources. Image size was rescaled/stretched to 640 pixels resolution. We used SGD as optimizer for training. We choose the initial learning rate(lr) as 10^{-2} and stop whenever rate(lr) reaches 10^{-5} . We used patience value as 100 and Weight decay by 0.1. We trained for 300 epochs. A total of 1500 images of fire and smoke with Average 10 bounding boxes per image (both fire/smoke) were used for training.

We re-used models of YOLOv5s and YOLOv5m with pretrained on COCO128 dataset. We used PyTorch framework for model training and testing. As said earlier, we used pretrained weights from YOLOv5s and YOLOv5m models and further fine-tuned and adapted weights for fire and smoke detection purposes. We used the above listed dataset. yaml as input labels for YOLO formatted dataset. For Training purposes, we used NVIDIA GTX 1080 GPU for acceleration in training of YOLO Deep learning model.

7. NETWORK ARCHITECTURES & EXPERIMENTAL RESULTS

We have re-used pretrained models of YOLOv5m and YOLOv5s and further trained these models for our fire and smoke detection problem, as per Transfer learning methodology. As per domain adaptation, the Pretrained models are well suited for similar domain and can be adopted to newer domain by re training of limited layers. For testing purpose, we have used FireNetV2 Jadon et al. (2019) [12] as Fire test set for evaluation of accuracies which are displayed in Table 7.1

Model	Accuracy (%)	AUC
yolov5m1	56.75	0.8236
yolov5s	46.39	0.7554
yolov5m2(Best)	-	0.8778

Table 7.1: Model Results for Fire Detection

For testing purpose for Smoke Detection, we have used Fire-Flame-Dataset Olayemi Abimbola (2019) [15] as Fire, smoke and Neutral test set for evaluation of accuracies using detection threshold as 0.25 which are displayed in Table 7.2

Model	Problem of Detection	Accuracy (%)
yolov5m2	Smoke	84.00
yolov5m2	Fire	94.00

Table 7.2: Model Results for Smoke and Fire Detection

Additionally, we have calculated detailed metrics in the form of Accuracies, Precision, Recall and F1-Score on Fire-Flame-Dataset Olayemi Abimbola (2019) [15] for both Fire and smoke using model "yolov5m2" with revised detection threshold as 0.3, as displayed in Table 7.3.

Problem of Detection	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Smoke	76.50	73.80	82.00	77.72
Fire	64.00	61.80	73.00	66.90

Table 7.3: Model Results for Fire Detection

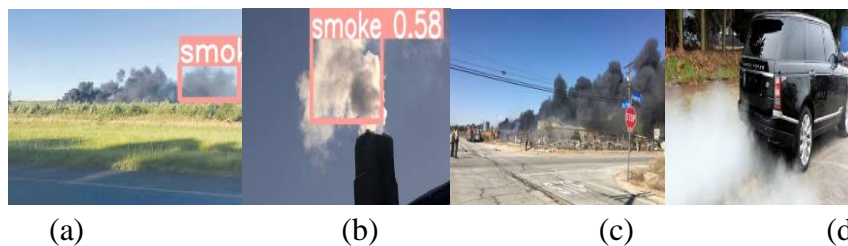


Figure 7.1: Smoke Detection Results on Smoke Test set Images (a) Smoke Detected (TP) (b) Smoke Detected (TP) (c) Smoke Not Detected (FN) (d) Smoke Not Detected (FN)

7.1 Detection Results

Following Figure 7.1, Figure 7.2 and Figure 7.3 shows the best model detection results on Fire-Flame-Dataset Olayemi Abimbola (2019) [15]. Only the test set of this dataset was used for testing the model efficiency.

7.2 Detection Results on Videos

We finally used the model for Detection on videos. As it is well known that Video is just a collection of images called as frames in video domain terminology. We tested on few selected videos and achieved detection results as shown on Figure 7.5



Figure 7.2: Fire Detection Results on Fire Test set Images (a) Fire Detected (TP) (b) Fire Detected (TP) (c) Fire Not Detected (FN) (d) Fire Not Detected (FN)

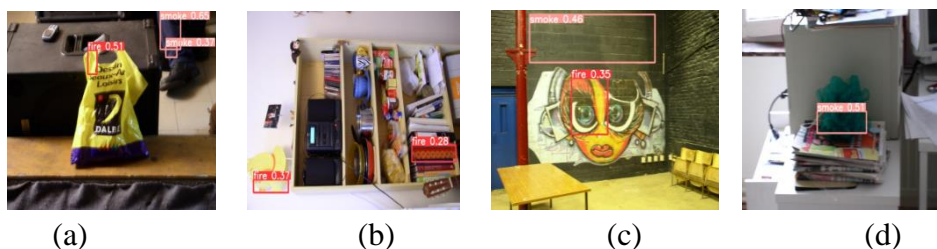


Figure 7.3: Fire & Smoke Detection Results on Neutral Test set Images (a) Fire Detected (FP) (b) Fire Detected (FP) (c) Fire & Smoke Detected (FP) (d) Smoke Detected (FP)



Figure 7.4: Fire & Smoke Non-Detection Results on Neutral Test set Images (a) Fire & Smoke Not Detected (TN) (b) Fire & Smoke Not Detected (TN)



Figure 7.5: Fire & Smoke Non-Detection Results on Videos (Screenshots) (a) Fire Detected in CCTV Video Frame (b) Fire & Smoke Detected in Video Frame

8. CONCLUSIONS

We have developed and trained advanced YOLO based DL models for the purpose of Smoke and fire detection. These models have been trained on suitable datasets and metrics are calculated. The data preprocessing is also used to improve the overall accuracy of the models.

9. FUTURE ENHANCEMENTS

The following activities will be part of future scope of work

- To further train deep learning models for fire and smoke detection and get improved accuracies
- To achieve more than 95% accuracy and accurate precision and recall values for the models on firenet dataset.
- To deploy the models on edge devices for example: Raspberry PI and NVIDIA Jetson Nano.

References

1. Aleksic, Z. (2004). "The analysis of the transmission-type optical smoke detector threshold sensitivity to the high-rate temperature variations." IEEE Transactions on Instrumentation and Measurement, 53(1), 80–85.
2. Amer, H. and Daoud, R. (2005). "Fault-secure multi-detector fire protection system for trains." 2005 IEEE, Instrumentation and Measurement Technology Conference Proceedings, Vol. 3. 1664–1668.
3. Amer, H. H. and Daoud, R. M. (2007). "Fault-secure multidetector fire protection system for trains." IEEE Transactions on Instrumentation and Measurement, 56(3), 770777.

4. Chino, D. Y. T., Avalhais, L. P. S., Rodrigues, J. F., and Traina, A. J. M. (2015). "Bowfire: Detection of fire in still images by integrating pixel color and texture analysis." 95–102.
5. Corp, W. (2020a). "www.wabtec.com/products/8582/smir"
6. Corp, W. (2020b). "www.wabtec.com/products/8582/smir"
7. Cortes, C. and Vapnik, V. (1995). "Support-vector networks." *Machine learning*, 20(3), 273–297.
8. Foggia, P., Saggese, A., and Vento, M. (2015). "Real-time fire detection for video surveillance applications using a combination of experts based on color, shape, and motion." *IEEE Transactions on Circuits and Systems for Video Technology*, 25.
9. Gagliardi, A., Gioia, F., and Saponara, S. (2021). "A real-time video smoke detection algorithm based on kalman filter and cnn." *Journal of Real-Time Image Processing*, 18.
10. Gagliardi, A. and Saponara, S. (2020a). "Advised: Advanced video smoke detection for real-time measurements in antifire indoor and outdoor systems." *Energies*, 13(8).
11. Gagliardi, A. and Saponara, S. (2020b). "Distributed video antifire surveillance system based on iot embedded computing nodes." *Applications in Electronics Pervading Industry, Environment and Society*, Saponara and A. De Gloria, eds., Cham. Springer International Publishing, 405411.
12. Jadon, A., Omama, M., Varshney, A., Ansari, M. S., and Sharma, R. (2019). "Firenet: A specialized lightweight fire and smoke detection model for real-time iot applications.
13. Jocher, G. (2020). "YOLOv5 by Ultralytics" <https://github.com/ultralytics/yolov5>
14. Kim, B. and Lee, J. (2019). "A video-based fire detection using deep learning models." *Applied Sciences*, 9, 2862.
15. Olayemi Abimbola, M. O. (2019). "<https://github.com/deepquestai/fire-smoke-dataset>.
16. Töreyn, B. U., Dedeoglu, Y., Güdükbay, U., and Çetin, A. E. (2006). "Computer vision based method for real-time fire and flame detection." *Pattern Recognition Letters*, 27(1), 49–58.
17. Venugopal, V. (2012). "Image processing based forest fire detection." *International journal of advanced research in engineering & technology*
18. Wang, M., Jiang, L., Yue, P., Yu, D., and Tuo, T. (2022). "Fasdd: An open-access 100,000-level flame and smoke detection dataset for deep learning in fire detection." *Earth System Science Data Discussions*, 2022, 1–22.
19. Toreyin, B. U., Soyer, E. B., Urfalioglu, O., and Cetin, A. E. (2008). "Flame detection using pir sensors." 1–4.
20. Zhang, Z., Jin, Q., wang, L., and Liu, Z. (2022). "Video-based fire smoke detection using temporal-spatial saliency features." *Procedia Computer Science*, 198, 493–498. 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare