The Scientific Temper (2024) Vol. 15 (4): 3290-3298
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.4.38
https://scientifictemper.com/
RESEARCH ARTICLE

Early detection of fire and smoke using motion estimation algorithms utilizing machine learning

Suprabha Amit Kshatriya[1*], Jaymin K Bhalani[2]

Abstract

An essential part of early warning and fire incident prevention in video surveillance systems is fire detection. The present study presents methodology that integrates motion estimation methods with the state-of-the-art convolutional neural network (CNN) architecture, YOLOv5, to provide effective fire detection. The methodology combines motion estimation techniques to improve the detection of dynamic changes suggestive of fire in video frames by the YOLOv5 model. The model incorporates motion analysis techniques, such as optical flow, to capture the spatial context and temporal relationships that are essential for differentiating between fire incidents and background activities. The research makes use of annotated datasets that cover a range of fire scenarios as well as non-fire activities, which guarantees reliable training and assessment of the YOLOv5 model. The outcomes of the experiments show how well the suggested strategy works to achieve high detection accuracy and real-time processing capabilities. Comprehensive performance indicators and comparison analysis are used to confirm the model's ability to accurately pinpoint flames in the presence of changing ambient variables and motion dynamics. By utilizing YOLOv5 and motion estimation algorithms, this research advances the field of fire detection technologies and provides a scalable and effective solution that can be integrated into emergency response frameworks, smart cities, and surveillance systems. The results highlight the possibility for improved situational awareness and proactive fire management through the integration of CNN architectures with motion analysis techniques. This abstract highlights the improvements in accuracy and real-time applicability of YOLOv5 with motion estimation methods for fire detection, outlining the research emphasis, methodology, experimental validation, and possible consequences.

Keywords: Machine learning, CNN architectures, Motion Estimation, Fire detection.

Introduction

In order to prevent and mitigate fire dangers as soon as possible and safeguard people and property, fire detection is a vital component of contemporary safety and security systems. This study examines different approaches and developments in the field of fire detection technology, with an emphasis on how computer vision techniques—such as convolutional neural networks (CNNs)—can be applied to increase the efficiency and accuracy of fire detection. It is impossible to exaggerate the significance of prompt fire. Detection. The National Fire Protection Association (NFPA) reports that fires cause substantial damage and human casualties annually throughout the world. Conventional fire detection systems frequently depend on heat sensors, smoke detectors, or manual surveillance; these methods may not be as reliable in various conditions or have as low a false alarm rate (Chen *et al.*, 2004). On the other hand, computer vision-based methods present a viable remedy by utilizing artificial intelligence to evaluate image data and instantly recognize patterns connected to fire.

In order to extract hierarchical features from input data and enable the model to distinguish between different objects, such as flames, smoke, and environmental variables suggestive of fire breakouts, a CNN's architecture usually consists of numerous layers. This work aims to accomplish two main goals: firstly, it reviews the state-of-the-art CNN-based fire detection techniques, encompassing model architectures, training approaches, and performance measures. Second, to suggest and assess methods that tackle current issues, such as integration with current fire safety, scalability to large-scale settings, and detection accuracy in changing lighting circumstances. This research

[1]Research Scholar, Parul University, Vadodara, Gujarat, India.

[2]Research Guide, Professor & Vice Principal, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India.

***Corresponding Author:** Suprabha Amit Kshatriya, Research Scholar, Parul University, Vadodara, Gujarat, India, E-Mail: suprabha.k-polyec@msubaroda.ac.in

How to cite this article: Kshatriya, S.A., Bhalani, J.K. (2024). Early detection of fire and smoke using motion estimation algorithms utilizing machine learning. The Scientific Temper, **15**(4):3290-3298.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.4.38

Source of support: Nil

Conflict of interest: None.

Received: 21/10/2024 **Accepted:** 17/11/2024 **Published** : 20/12/2024

aims to contribute to the development of more robust and reliable fire detection systems that can minimize response times, reduce false alarms, and ultimately improve overall safety and security in both residential and commercial settings by pushing the boundaries of current fire detection(Muhammad *et al*., 2018). The theoretical underpinnings of CNNs, pertinent literature on fire detection techniques, experimental results and analysis, and future research directions to propel the field of computer vision-based fire detection forward are covered in detail in the sections that follow in this paper.

Object detection requires a few crucial steps

Detection head

YOLOv5's detection head employs a number of convolutional layers and up-sampling methods to forecast bounding boxes, class probabilities, and confidence ratings. This head architecture makes it easier to precisely locate and categorize objects in variety of settings.

Model variants

There are several sizes of YOLOv5, each with a different depth and computing requirements. Users can select a model version according to particular hardware limitations and performance needs thanks to this scalability.

Training and optimization

YOLOv5 makes use of sophisticated training methods like progressive scaling, data augmentation procedures, and automated hyperparameter optimization. By using techniques, models become more resilient and have a better ability to generalize across different types of datasets. Input Processing: Prior to being fed into the model, images undergo pre-processing to normalize pixel value dimensions.

Feature extraction

The input image's hierarchical features are extracted by the backbone network, which gathers spatial data necessary for object detection tasks. Bounding Box Prediction: For every class of identified objects in the image, the detection head predicts bounding boxes (coordinates), class probabilities, and confidence scores. Post-processing: The final collection of identified items is refined based on confidence scores by applying non-maximum suspension (NMS) to filter out redundant bounding boxes. YOLOv5 can be optimized for fire detection using particular datasets that show pictures or video frames of fires, smoke, and pertinent ambient factors. The model gains great accuracy and efficiency in identifying and localizing objects connected to fire by receiving training. YOLOv5, which offers state-of-the-art performance in terms of speed, accuracy, and scalability, represents a substantial leap in object identification utilizing CNN architectures. Its use in fire detection highlights its adaptability and possible influence on improving safety precautions and emergency response procedures. Convolutional Neural Networks (CNNs) are the main structural component of YOLOv5. With the help of these networks, which are made to automatically recognize patterns in visual data, the model can identify objects—including fires—using learned features as opposed to manually created criteria. To enhance its detecting skills in a variety of settings and sizes, YOLOv5 makes use of specialist modules (such as SPP and PANet) and a sequence of convolutional layers.

Benefits of YOLOv5 in Fire Safety

Real-time Detection

YOLOv5 allows for real-time fire detection in static photos or video streams, even on devices with limited computational power. It does this by operating efficiently.

Accuracy

YOLOv5 has a high degree of precision in differentiating between fires and non-fire objects thanks to substantial training on a variety of datasets (Redmon *et al*., 2016).

Adaptability

The design can manage all kinds of fires and is flexible enough to adapt to different surroundings, guaranteeing reliable performance under trying circumstances.

Utilizing image processing techniques on visual data (such as photos or video frames) to detect the presence of flames or smoke is a common method of utilizing machine learning to detect fires.

Acquisition of Datasets:

Compile a dataset of pictures or videos that show both non-fire (normal scenes) and fire-related circumstances

The dataset ought to be varied and inclusive of various fire kinds, settings, and circumstances.

- Preparation: Pre-process the pictures or video frames to standardize them by doing things like resizing and normalizing pixel values.
- Model Selection: Select a suitable model for machine learning. CNNs, or convolutional neural networks, are widely utilized for image-based applications such as fire detection.
- Training: Divide the dataset into test, validation, and training sets. Use the training set to train your model, aiming for performance metrics like F1-score, accuracy, precision, and recall. Using the validation set, validate your model and make any necessary hyperparameter adjustments to avoid overfitting.
- Evaluation: Examine how well your model performs on the test set to determine how accurate it is in identifying fires.
- Deployment: After you're happy with the performance, use your model to start fires in situations that happen

in real-time. If real-time detection is needed, integrate it with the proper sensors or cameras to record a live video stream.

- Monitoring and improvement: Keep an eye on how well your model is working in practical applications. Gather input and information so that the model can be refined over time and made to fit differently.
- Problems: Variability in the dataset: Making sure it includes a variety of fire and non-fire settings.

Block Diagram for Fire Detection

The steps involved in the fire detection using CNN in YOLOv5 is explained in details using the block diagram and CNN architecture related work for fire detection (Figure 1).

CNN Architecture related work in YOLOv5 for Fire Detection (Table 1)

There are multiple processes involved in utilizing a CNN to detect fire in video. This is a general strategy that you can use (Figure 2):

Compiling and preparing the dataset

- *Gather videos of fire and non-fire*

Compile a collection of videos, both with and without fire sequences. Make sure there are a variety of non-fire (regular activities, scenes from nature) and fire scenarios.

- *Labeling*

Indicate whether or not there is a fire in each frame.

Data Pre-processing

- *Frame extraction*

Take individual video frames and combine them into a huge set of pictures.

- *Normalization*

Set each frame's pixel values to a standard scale, usually ranging from 0 to 1.

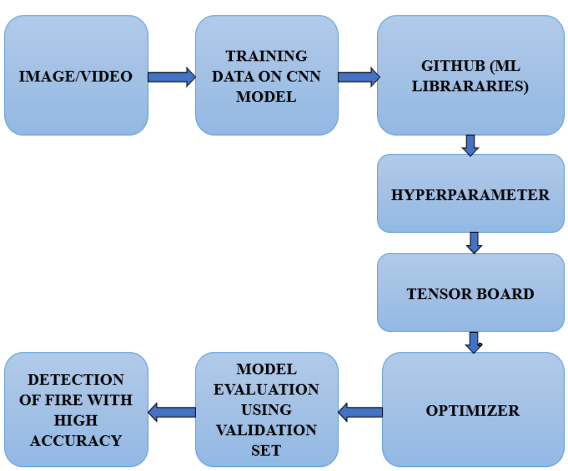

Figure 1: Block diagram of fire detection system

CNN Architecture Selection

Select or create a CNN architecture that is appropriate for classifying video frames. Popular options consist of: 3D Convolutional Networks: Handle simultaneous processing of spatial and temporal data. 2D CNNs + LSTM: Employ a combination in which LSTMs model temporal dependencies and 2D CNNs extract spatial features.

CNN + Optical flow

To capture motion, use CNNs for spatial characteristics and add optical flow data. Training of the Model

Split Data

Separate the dataset into test, validation, and training

Augmentation

Use techniques like random harvests and flips to add variation and robustness to data.

Instruction

Use the training set to instruct the CNN To prevent overfitting, keep an eye on the validation set's performance. size, regularization strategies, and learning rate.

Test set evaluation

Assess the accuracy, precision, recall, and F1-score of the trained model using the test set.

Thresholding

Establish a limit for the output likelihood of the model to identify a frame as having fire.

Deployment inference

Classify frames in live videos using the learned model.

Integration

Apply the model to a broader fire detection system, like a fire monitoring program.

Real-time constraints

If real-time detection is required, take into account the competing demands and adjust the hardware and model accordingly.

The many layers of the well-known object recognition model YOLOv5, which is built on CNNs, have varied functions in processing input data and extracting features required for identifying things in films, including fires. The main components of the YOLOv5 architecture, which is specially designed for fire detection, are broken down here along with their functions:

CSPDarknet53, the backbone network

Goal

The feature extractor is the backbone network or CSPDarknet53.

Function

It extracts hierarchical characteristics at several scales by processing the supplied video frames. Ensuring the capture of both local and global context in image frames is crucial for the accurate separation of fires.

Features

Activation functions (such ReLU), batch normalization, and convolutional layers, which are meant to learn, usually make up the backbone.

Layers Around the Neck

Goal

YOLOv5's neck layers, like PANet's, combine functionality from several backbone network tiers.

Function

By combining features from various network scales or levels, they improve feature representation. This enhances the ability to locate and identify fires in video frames of different sizes and orientations.

Features

To create strong feature maps for later detection, neck layers frequently incorporate procedures like feature concatenation, skip connections, and spatial pyramid pooling (Figure 3).

Head Divisions

Goal

YOLOv5's head layers are made up of layers that are particular to detection.

Function

Based on the fused features, they produce predictions for confidence scores, bounding boxes, and object classes (including fire).

Features

Activation functions and output layers come next, with head layers usually consisting of convolutional layers with limited spatial dimensions.

Loss Function

Goal

The loss function computes the difference between the ground truth annotations and the anticipated outputs. Examples of such loss functions are CIoU and Focal loss, which are exclusive to YOLOv5.

Table 1: Comparison of different Yolo Version for best result for Fire Detection

S. No	*Algorithm*	*mAP (%)*	*FPS*	*Size (MB)*
1	YOLOv3	75.7	52	235
2	YOLOv4	82.7	54	244
3	YOLOv5	91.6	71	14

Table 2: Dataset categories for detection

S. No	*Fire images*	*Fire images*	*Non fire images*	*Videos*
1	Training	127	80	02

Function

It directs the training procedure by penalizing erroneous guesses and motivating the model to increase its precision and accuracy in fire detection localization.

Features

By combining terms for confidence estimation, classification, and bounding box regression, the loss function optimizes the model to precisely identify fires while reducing false positives and negatives. 5. Reprocessing
Goal: Refinement of output predictions is the goal of post-processing

Qualities

Post-processing procedures are crucial for enhancing the accuracy and dependability of fire detection, particularly in dynamic video situations where several fires may occur in close proximity to one another. Every layer in the YOLOv5 architecture is essential for analyzing video frames and extracting pertinent information.

CNN architecture for fire and Smoke detection

Convolutional neural networks (CNNs) can automatically learn and extract relevant features from visual data. They are commonly employed for fire detection in photos and videos(Ren *et al*., 2015) and YOLO (Redmon *et al*., 2016). CNN architectures intended for fire detection typically include

Table 3: Results of fire detection

Epoch	*GPU_mem*	*box_loss*	*obj_loss*	*cls_loss mAP*
0/59	0.914G	0.1064	0.003233	00.029
1/59	0.919G	0.08236	0.02655	00.118
2/59	0.919G	0.02963	0.02655	00.225
.	.	.	.	.
.	.	.	.	.
57/59	0.919G	0.01418	0.01418	00.7
58/59	0.919G	0.01304	0.01304	00.768
59/59	0.919G	0.01394	0.01393	00.783

Table 4: Dataset categories for fire and smoke detection

S. No	*Type*	*Train*	*Val*	*Test*
1	Smoke Only	3412	278	312
2	Smoke with Fire	3233	301	298
3	Non-Smoke	2476	304	331
4	Fire only	1789	231	256
5	Foggy Images	120	68	45

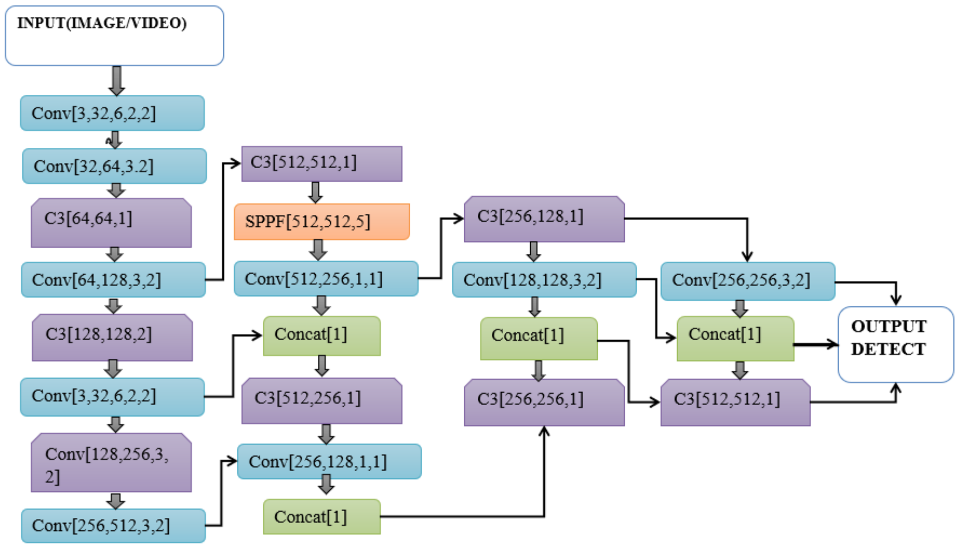

Figure 2: CNN architecture for detection of fire and smoke

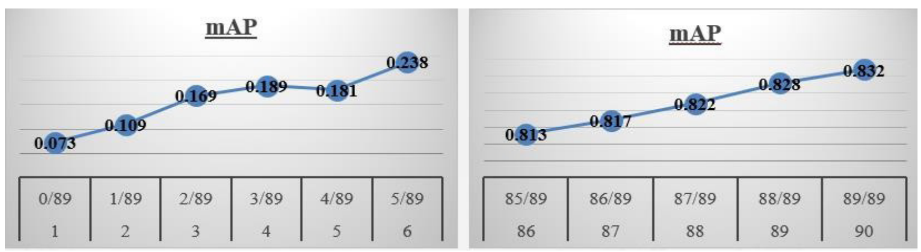

Figure 3: Results of mAP

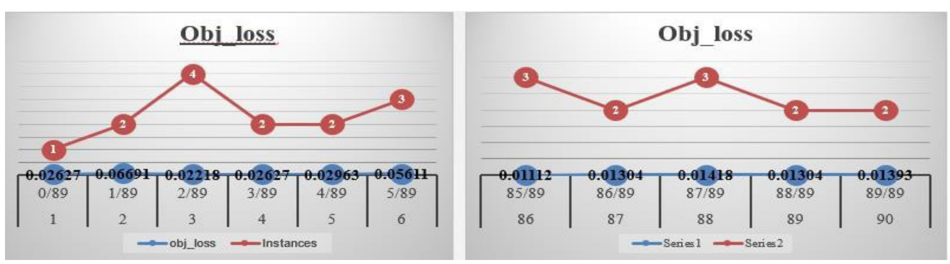

Figure 4: Results of object loss for detection of fire and smoke

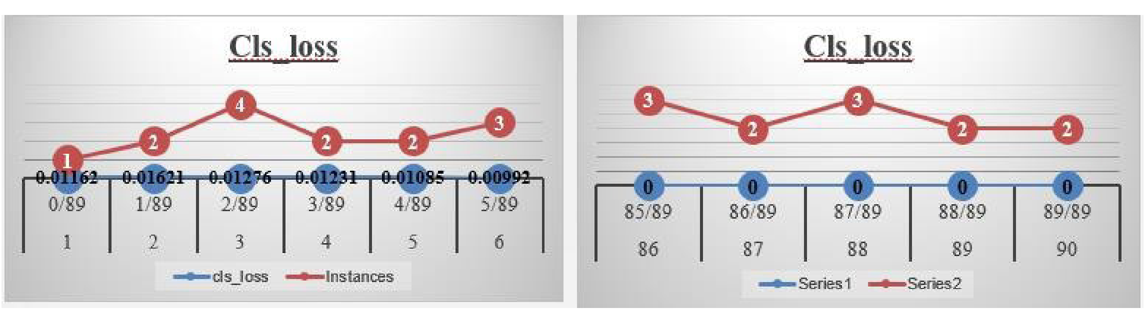

Figure 5: Results of class loss for detection of fire and smoke

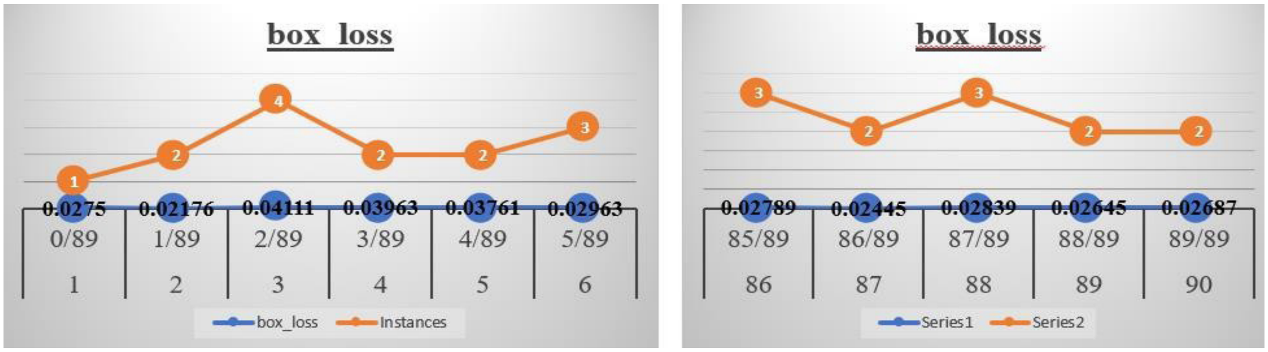

Figure 6: Graph of box loss for detection of fire and smoke

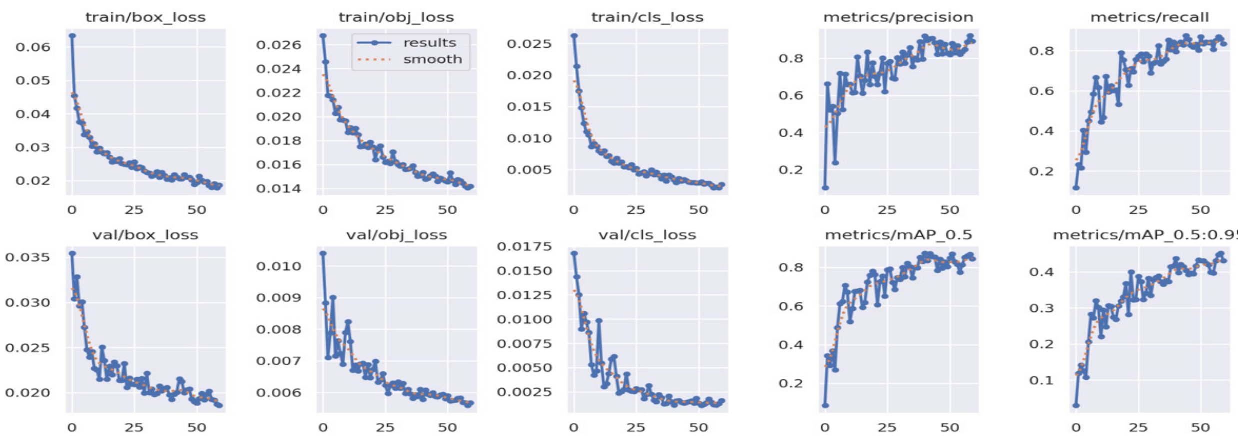

Figure 7: Screenshot of the process of fire and Smoke detection with different losses

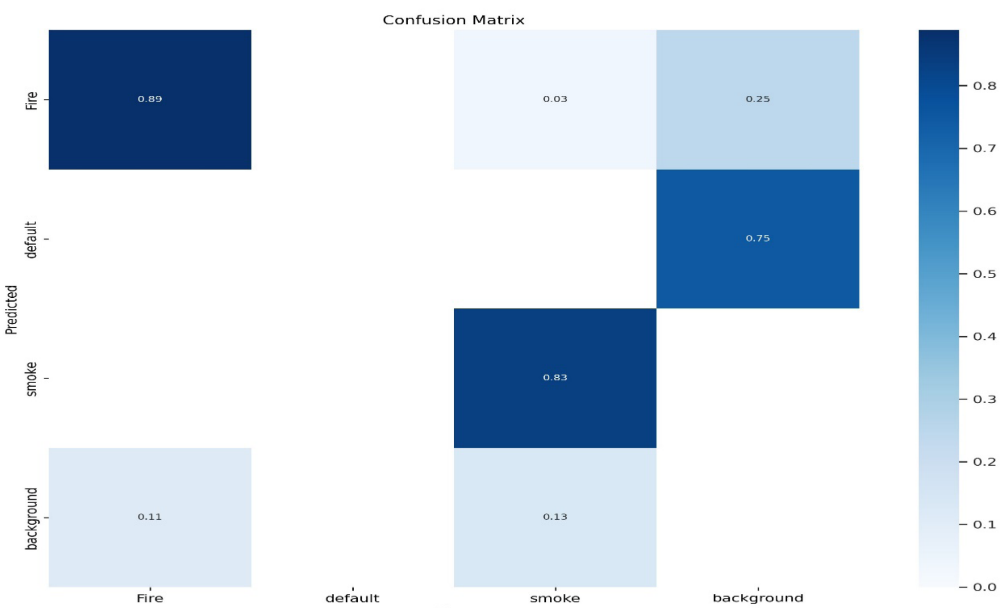

Fig. 8 The confusion matrix of the real and predicted categories

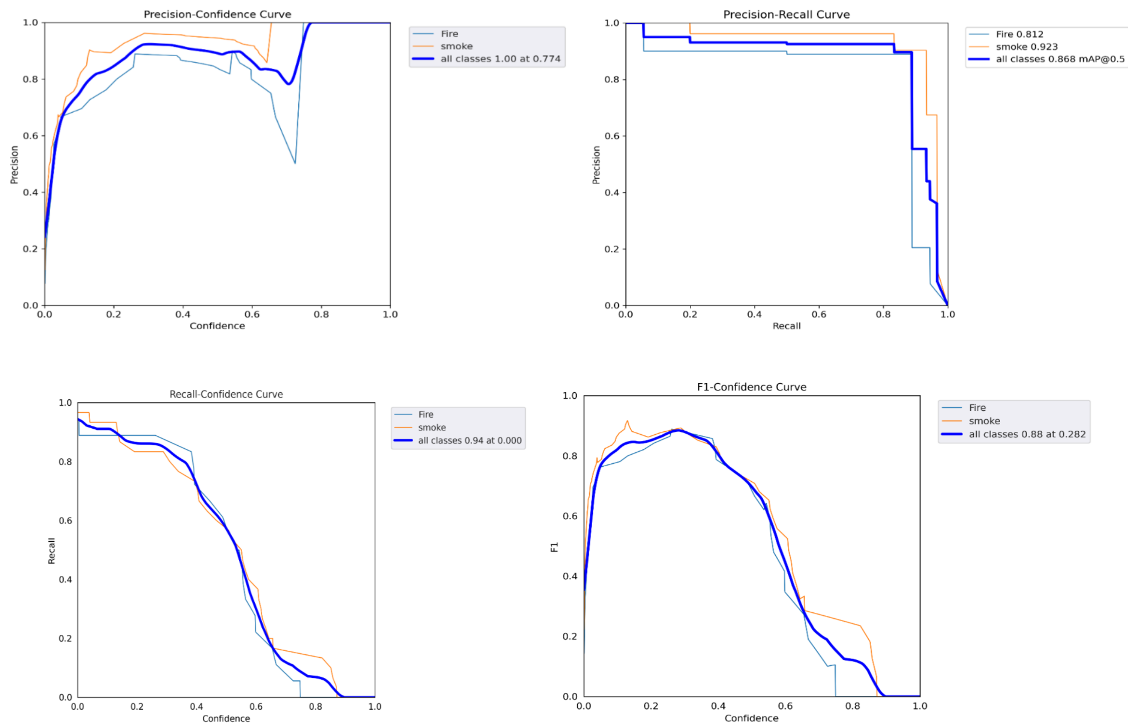

Figure 9: Screenshot of precision-recall curve

Figure 10: Screenshot of the result of fire detection

Table 5: Results of fire detection

Epoch	*GPU_mem*	*box_loss*	*obj_loss*	*cls_loss*	*mAP*
0/89	0.914G	0.0275	0.03233	0.011	0.078
1/89	0.919G	0.02176	0.02155	0.012	0.109
2/89	0.919G	0.04111	0.02055	0.016	0.289
.	.	.	.	.	.
.	.	.	.	.	.
87/89	0.919G	0.01487	0.01113	0	0.819
88/89	0.919G	0.01233	0.01267	0	0.833
89/89	0.919G	0.01309	0.01206	0	0.828

multiple layers, each of which serves a distinct purpose. The many layers that are frequently employed in CNNs for fire detection are broken down in detail below: The input layer receives the input frame from the video or image. The size of the input image or frame determines the input layer's dimensions (height, width, number of color channels, etc.). CNNs are mostly composed of convolutional layers (Figures 4-10 and Tables 2-5).

CNN architecture for detection of fire In contrast, recall is a false positive observation ratio, as previously shown in research. With 1.2% false positives and 91% accuracy, our recommended model performed admirably. Equations (1) and (2) can be used to obtain the mean precision and recall rates of our proposed method: A precision-recall curve is a graph that results from plotting the accuracy ratio against the recall rate (P-R plot). The model's FM score may also be used to gauge its efficacy. Another criterion used in this investigation was the average accuracy of each detection (AP).

$$\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FP}} \;......\mathbf{(1)} \qquad (1)$$

$$\text{Recall} = \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FP}} \;.....\mathbf{(2)} \qquad (2)$$

The following is a definition for the FM score:

$$FM = \frac{2 \times precision \times recall}{precision + recall}$$

AP = Precision (Recall)*d*(Recall) 1

Conclusion

This study used deep CNN models and the YOLOv5 object detector to construct a fire detection system. Fire image datasets with various fire situations were used to train the suggested fire detection system. It used films and pictures to detect fire. For the purpose of training and validating the model, we produced a dataset for fire detection that had 3233 fire photos and also some videos. We conducted experiments to assess the suggested system's performance, both quantitatively and qualitatively, by contrasting it with other well-known one-stage object detectors. With 89% mAP, the assessment and testing results demonstrated that the YOLOv5 model outperformed other YOLO versions and was robust on our fire detection dataset. Because of its effectiveness and versatility, the suggested fire detection approach enables researchers to detect fires at an early stage. Furthermore, other methods can be applied to understand chemical-based fire with high accuracy.

Author contribution

Suprabha Amit Kshatriya: Conducted the research, collected and analyzed data, and wrote the manuscript.

References

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

Cappellini, V., Mattii, L., & Mecocci, A. (2004). An intelligent system for automatic fire detection in forests. In Proceedings of the 3rd International Conference on Image and Signal Processing (pp. 351-357).

Çetin, A. E., Dimitropoulos, K., Gouverneur, B., Grammalidis, N., Günay, O., Habiboğlu, Y. H., & Verstockt, S. (2016). Video fire detection–review. Digital Signal Processing, 23(6), 1827-1843.

Chen, T. H., Wu, P. H., & Chiou, Y. C. (2004). An early fire-detection method based on image processing. In 2004 International Conference on Image Processing, 2004. ICIP'04. (Vol. 3, pp. 1707-1710). IEEE.

Dimitropoulos, K., Barmpoutis, P., & Grammalidis, N. (2015). Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. IEEE transactions on circuits and systems for video technology, 25(2), 339-351.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., & Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 2758-2766).

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In Scandinavian conference on Image analysis (pp. 363-370). Springer, Berlin, Heidelberg.

Foggia, P., Saggese, A., & 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), 1545-1556.

Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. Artificial intelligence, 17(1-3), 185-203.

Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, L. Changyu, & Tkianai. (2021). ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations (v5.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.4679653

Kang, S. M., Wildes, R. P., & Loy, C. C. (2017). Detect or track: Towards cost-effective video object detection/tracking. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 8517-8524).

Ko, B. C., Cheong, K. H., & Nam, J. Y. (2009). Fire detection based on vision sensor and support vector machines. Fire Safety Journal, 44(3), 322-329.

Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In IJCAI (Vol. 81, pp. 674-679).

Muhammad, K., Ahmad, J., Lv, Z., Bellavista, P., Yang, P., & Baik, S. W. (2018). Efficient deep CNN-based fire detection and localization in video surveillance applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(7), 1419-1434.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28, 91-99.

Sharma, J., Granmo, O. C., Goodwin, M., & Fidje, J. T. (2017). Deep convolutional neural networks for fire detection in images. In International Conference on Engineering Applications of Neural Networks (pp. 183-193). Springer, Cham.

Sun, D., Yang, X., Liu, M. Y., & Kautz, J. (2018). PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In Proceedings of the IEEE conference on computer vision and

pattern recognition (pp. 8934-8943).
Töreyin, B. U., Dedeoğlu, Y., & Çetin, A. E. (2005). Flame detection in video using hidden Markov models. In 2005 IEEE International Conference on Image Processing (Vol. 2, pp. II-1230). IEEE.
Verstockt, S., Van Hoecke, S., Tilley, N., Merci, B., Sette, B., Lambert, P., & Van de Walle, R. (2011). FireCube: a multi-view localization framework for 3D fire analysis. Fire Safety Journal, 46(5), 262-275.