Comparative Analysis of Traditional and Advanced Smoke and Fire Detection Systems

Nisuli Hettiarachchi, Thathsarani Walagedara, Oshen Geenath, Prasan Yapa
BSc (Hons) Artificial Intelligence and Data Science
Robert Gordon University, Aberdeen, UK
{n.hettiarachchi, t.weerasiri-walagedara, o.arul-jeganathan}@rgu.ac.uk, prasan.y@iit.ac.lk

***Abstract* — This paper brings out a significant review which is conducted on how cutting-edge technologies like artificial intelligence (AI) have been used to enhance fire and smoke detection systems. We investigate the effectiveness of AI-powered algorithms in augmenting security measures, focusing on the integration of AI with advanced techniques. This paper also provides a critical analysis of advanced object detection-oriented Convolutional Neural Network (CNN) models, including Faster-RCNN, R-FCN, SSD, and YOLO v7, particularly in the context of image fire detection. The potential of AI to surpass conventional detection methods and establish more sophisticated and proactive security protocols is a key theme of the paper. The authors illustrate how AI is reshaping safe environments, enabling rapid communication with emergency services, and paving the way for a future where AI is integral in creating safer environments. This study emphasizes the latest developments in AI-based security systems and the ongoing paradigm shift towards AI-centric approaches in safety and emergency response mechanisms.**

Keywords — Artificial Intelligence, Fire and Smoke Detection Systems, CNN, YOLO v7, Real-time Analysis

I. INTRODUCTION

The exponential growth in volume and complexity of construction have led to an increase in the issues associated with fire control as the economy has developed quickly. To reduce fire losses, early fire detection, and highly sensitive and accurate systems are essential. However, in huge spaces, intricate buildings, and places with plenty of disturbances, conventional fire detection technologies—such as smoke and heat detectors—face limits. The implementation of early fire alerts has been hampered by these restrictions, which lead to false detections, erroneous alerts, and detection delays.

The focus of research has shifted to image-driven fire and smoke identification as a potential solution to these problems. This novel approach provides flexible installation, high accuracy, early detection, and effectiveness in large and complex structures. This technology's key component is its detection algorithm, which directly impacts how well the image fire detector works. Three essential phases are involved in developing image fire and smoke detection algorithms: fire and smoke detection, feature extraction, and image preprocessing. Among them, feature extraction is a crucial component. Due to their reliance on specialized knowledge, traditional methods are limited in their adaptability as they frequently use machine learning classification and manually chosen fire features. The results of a plethora of research on smoke and flame image features have been limited to basic features like color, edges, and textures.

Seeing the need for more sophisticated feature extraction techniques, this work explores the field of sophisticated convolutional neural networks (CNNs) for image fire detection. With a focus on CNNs specifically, this review offers a thorough examination of the approaches and architectures used in image-based fire and smoke detection. The analysis explores the essential elements of CNNs, giving importance to the significant function of convolutional layers in recognizing attributes and classifying areas connected to fire.

The review methodically investigates the Inception Resnet [15], a cutting-edge CNN, to clarify how feature extraction has changed throughout various convolutional layers. It becomes clear that more complex properties are captured by deeper layers; therefore, deep networks must be used to extract complex image features that are necessary for reliable fire and smoke detection, particularly in difficult real-world scenarios. The paper then examines the architectures of the fire and smoke detection performance of four well-known object detection networks: Faster R-CNN, R-FCN, SSD, YOLO v3 [27], and YOLO v7 [5]. Known for its increases in speed and accuracy, faster R-CNN has proven effective in several areas, such as fire detection. R-FCN uses position-sensitive ROI pooling to provide better object localization, thereby addressing the depth-accuracy problem in deep learning. The single-shot technique of YOLO v7 and the one-stage architecture of SSD are examined.

II. LITERATURE REVIEW

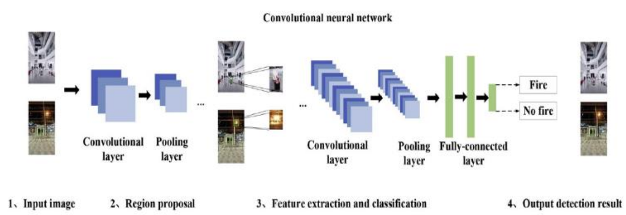

Figure 1. CNN

A. CNN

The flow of image fire detection methods inspired by convolutional neural networks is built in Figure 1 by the object detection algorithms idea. The detection CNN [3] comprises region proposals, feature extractions, and final image classification. First, using techniques like convolution and pooling, the CNN uses an image as input to produce area suggestions. Second, the item detection based on region CNN uses fully connected layers, pooling layers, convolutional layers, etc. To determine whether there is a fire in certain regions or not.

A deep normalization and convolutional neural network (DNCNN) with 14 layers for smoke detection has been introduced to achieve low false alarm rates and high detection rates. This method employs normalization and convolutional layers for efficient feature extraction and classification [16]. Fire pixel region detection has been implemented in real-time without temporal information, using reduced complexity CNN architectures. It achieves high accuracy and increases computational performance, processing up to 17 fps on contemporary hardware [17]. A spatial-temporal CNN has been proposed for video smoke detection, combining intra-frame appearance and inter-frame motion features. It achieves a high detection rate and low false alarm rate, significantly outperforming existing methods [21]. A study on a novel CNN algorithm for fire and smoke image detection has been conducted, using adaptive piecewise linear units and a newly created dataset. The approach achieves high accuracy and detection rates with a very low rate of false alarms [22]. Focusing on real-time flame detection, a CNN-based algorithm is proposed. It includes a candidate target area extraction algorithm and a deep neural network model, effectively identifying fire and ensuring real-time fire warning performance [26].

The central component of CNNs is the convolutional layer. In contrast to other neural networks that employ weighted sums and connection weights, the convolutional layer creates feature maps of the original pictures using image transform filters known as convolution kernels. A collection of convolution kernels makes up the convolutional layer. To create a feature map, the convolution kernel glides over the images and calculates a new pixel by a weighted sum of the pixels it floats over. The original image's features for each aspect are reflected in the feature map. The convolution layer calculation formula is computed using the equation below [7].

$$y = \sum_{j=0}^{J-1} \sum_{i=0}^{I-1} w_{ij} x_{m+1,n+j} + b, (0 \le m \le M, 0 \le n \le N)$$

Figure 2. Calculation for Convolutional Layer

In Figure 2, x represents the size of an input image. W * H, where w stands for a convolution kernel with size J * I, y stands for output feature maps, and b stands for bias. In real life, training is used to ascertain the values of w and b. The range of indices that are valid is indicated by the notations $0 \le m \le M$ and $0 \le n \le N$. Any integer value between 0 and $M-1$ (inclusive) can be assigned to m, and any integer value between 0 and $N-1$ (inclusive) can be assigned to n. This indicates that the array x's elements that fall inside the boundaries given by M and N are all iterated over by the sum.

Figure 3 displays the 32 feature maps of a fire image produced by the first convolutional layer's 32 kernels in Inception Resnet, a cutting-edge CNN. Feature maps and convolution kernels have the same number. For instance, three feature maps are produced since this layer contains three convolution kernels. Pixel color indicates the level of activations. White pixels in the feature map at a certain spot denote highly positively activated pixels in the original image at that same location. Excessively negative activations are shown by black pixels. Grey pixels indicate weak activations. The feature map produced by this layer's convolutional kernel 14 is active on edges when compared to the original image. On the top and lower margins, the dark and light regions are activated positively, and the bright and dark areas are active negatively. Since the whiter pixels in the feature map match the orange regions in the original image, the convolutional kernel 26's output is triggered on the orange pixels [7].

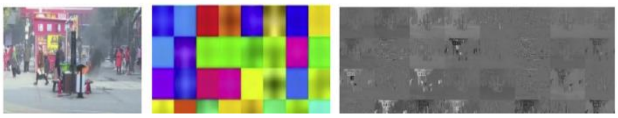

Figure 3. Features Extracted in the First Layer

It indicates basic properties like color, borders, and so on are mostly learned and extracted by the kernels in previous layers. When scenes are complicated and contain numerous interference occurrences, it is found through feature map analysis that simple features are unable to distinguish between fire and disturbance. As a result, algorithms for image fire detection that can extract intricate image features for fire recognition in practical situations must be developed. In this regard, deep convolutional neural networks perform better. Shen displays samples of the kernels found in Inception Resnet V2's first, third, and sixth convolutional layers. Figure 3 displays samples of the kernels found in Inception Resnet V2's first, third, and sixth convolutional layers. It suggests that at deeper levels, the networks capture more intricate properties. Thus, a deep network is required for the extraction of complicated image features.

B. Faster-R-CNN

To create image fire detection algorithms, four image object detection networks—Faster-RCNN, R-FCN, SSD, and YOLO v7—with superior performance in terms of speed and accuracy of detection are chosen. The four algorithms' architectures are displayed in Figure 5**.** Faster R-CNN [1] is a prominent deep-learning technique in the field of computer vision, notable for its evolution from earlier models such as R-CNN and Fast R-CNN. This advancement brings significant improvements in both speed and accuracy, making it an essential tool in object detection.

In terms of its operational framework, Faster R-CNN works in two primary stages. [29] The first stage, known as the Region Proposal Network (RPN), involves the generation of feature maps from the input image using networks like VGG, ResNet, or Inception. The RPN then predicts potential object-containing regions, each assigned with an abjectness score and location The output of item presence probabilities and box regression, which is carried out with the help of a two-class softmax layer and the Smooth L1 loss function, mark the end of this stage.

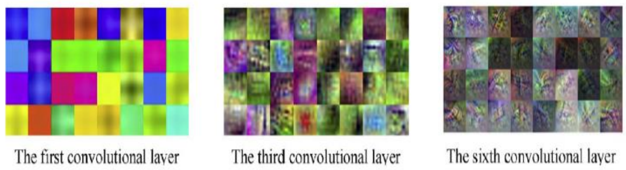

Figure 4. Kernels in Other Layers

The second stage of Faster R-CNN [2] focuses on class prediction and box refinement. It uses the locations identified by the RPN to extract features from the same feature map via Region of Interest (ROI) pooling. After processing, these regional feature maps are used to refine box positions and forecast class results. This stage is optimized for computational efficiency by reusing the feature map for cropping proposals, thus reducing the need for redundant computations. However, the processing speed of the system is dependent on the number of proposal regions generated by the RPN.

Faster R-CNN [1] has been particularly effective in various domains, including fire detection. In the realm of fire detection, it has been used in combination with multidimensional texture analysis for image-based fire detection. It simplifies wildland forest fire smoke detection by avoiding manual feature extraction, which is a significant advantage over traditional video smoke detection methods. Faster R-CNN has been demonstrated in comparative studies to perform better in fire and smoke detection scenarios than other models such as YOLO and SSD. Experiments have demonstrated its capacity to increase detection speed, reduce false alarm rates, and enhance overall accuracy. Its adaptability has been proven to effectively detect fire and smoke in both outdoor and indoor environments.

C. R-FCN

R-FCN (Region-based Fully Convolutional Networks) is a sophisticated object detection framework that significantly improves upon the limitations of earlier models such as Faster R-CNN [1]. It is tailored to boost both the speed and accuracy of object detection tasks, making it highly suitable for critical applications like fire and fire detection systems.

In addressing the depth versus accuracy challenge in deep learning, R-FCN focuses on the need for CNNs to be deep enough to extract complex features, thereby enhancing the accuracy of the networks. However, a challenge arises with deeper convolutional layers as they tend to generate small-scale feature maps. These maps are less sensitive to translation, which can result in inaccuracies in object localization. To combat this, Faster R-CNN incorporates a Region Proposal Network (RPN) in tandem with an intermediate convolutional layer to maintain object location data for more accurate detection.

One of the limitations of the Faster R-CNN framework is found in its standard ROI pooling operations. These operations typically lack translation variance, necessitating the individual processing of each regional feature map in the object detection network. This requirement not only slows down the process but also affects the overall efficiency of the detection system.

R-FCN optimizes the object detection process through selective ROI pooling [29]. Before prediction, it pools ROI from the final layer of feature maps, which reduces the amount of computing needed. Furthermore, R-FCN introduces a position-sensitive ROI pooling operation that enhances the accuracy of object localization. This operation is designed to encode position information into each ROI, thereby countering the reduced sensitivity to translation in small-scale feature maps from deeper layers.

In terms of applications in fire detection systems, R-FCN offers several advantages. Its position-sensitive ROI pooling enables more accurate localization of objects, which is essential for identifying intruders or detecting the sources of fire in complex environments. The framework's efficient computation allows for the rapid processing of images, crucial for real-time detection in surveillance systems. Moreover, R-FCN's versatility in handling deep feature maps makes it adaptable to various detection scenarios, whether detecting an intruder in a restricted area or identifying the early stages of a fire.

In conclusion, R-FCN represents a significant advancement in object detection technology. Its position-sensitive ROI pooling [29] and efficient computational design provide considerable improvements over traditional models like Faster R-CNN, especially in terms of speed and localization accuracy. R-FCN implementation in fire detection systems can greatly improve the system's capacity to identify and address issues quickly, boosting safety and security measures in a variety of settings. The balance R-FCN maintains between deep feature extraction and efficient computation positions it as an effective tool in the realm of deep learning-based object detection.

D. SSD

Particularly in large settings like woods, the Single Shot MultiBox Detector [8] (SSD) has become a powerful instrument for real-time fire surveillance in recent years. With its unified, one-stage architecture for the simultaneous forecasting of boundary locations and object categorization, the SSD architecture is a trailblazing solution in the field of object detection. In this overview, the essential elements, and characteristics of SSD [8] are explained, with a focus on how practical and effective they are, especially when it comes to real-time object recognition.

The three fundamental components of SSD's architecture illustrate how effective it is at enabling quick and accurate object identification. A feature extraction convolutional network is used in the first phase, which allows for the selection of different architectures, including ResNet, VGG, Inception, Inception Resnet-v2, and MobileNets [8]. The use of MobileNets in the studied work highlights its effectiveness for embedded and mobile vision applications, offering a strong balance between computational efficiency and accuracy.

Convolutional layers are introduced in the second part to provide multi-scale feature maps that support the recognition of objects of various sizes. SSD is a flexible option that could potentially be used in a variety of real-world situations because of its adaptable characteristics. To anticipate bounding box locations and confidences for various object categories, the last component uses final convolutional layers with an insignificant kernel, consolidating the architecture's capabilities for comprehensive object detection.

The study's implementation of SSD incorporates the MobileNets [8] base model, enhanced by extra feature layers and classifier convolutional layers. With the help of Non-Maximum Suppression, this setup makes it easier to create many boxes per class and gives priority to boxes with the highest confidence scores. The input images are standardized to 300 x 300 (SSD300) dimensions, which is consistent with the architecture's ability to detect in real-time.

SSD uses a neural model to evaluate default boxes at different scales and locations in different feature maps during the training phase. The architecture's commitment to accurately predicting bounding box locations and class confidences is reflected in the model loss, a mix of localization loss (Smooth L1), and confidence loss (softmax).

SSD sets itself apart by being able to recognize objects at numerous scales, which eliminates the need for laborious proposals. This efficiency is especially important for the intended use of UAV-based fire surveillance, where real-time reaction is essential. The addition of MobileNets, which is distinguished by its focus on depth-wise detachable convolutions, further establishes SSD as a precise yet lightweight substitute that is especially useful in computing contexts with limited resources.

The SSD's single shot [11], multi-scale detection method, combined with the prudent use of MobileNets, highlights its relevance and effectiveness in real-time applications. The architecture's versatility is demonstrated by its successful application in UAV-based fire monitoring, which also highlights the architecture's capabilities to handle complex, dynamic settings where precise and quick object detection is critical.

E. YOLO v7

In the field of object detection, the You Only Look Once (YOLO) technique is a revolutionary approach that differs from traditional methods, such as the Convolutional Neural Network (CNN) [6]. Unlike CNN, which uses localization or classification models, YOLO presents a new single-shot method that can identify objects at an astounding 45 frames per second. YOLO is positioned as a significant progress by this noticeable speed improvement, especially when compared to more conventional techniques like R-CNN [9].

According to the YOLO approach, an image is segmented into an S x S grid, with bounding boxes and confidence values defined for each grid unit. The confidence value is a probabilistic measure that indicates the probability that an object is present within the bounding box that has been defined. It is determined by the intersection over union (IOU) metric. Notably, YOLO's single-shot architecture, which does away with the necessity for iterative proposals, simplifies the detection process. It is supported by a neural network with 24 convolutional layers.

Bounding box prediction, classification probability calculation, and the use of non-maximum suppression—a technique that maximizes the choice of the most relevant bounding boxes—are all included in the architecture of YOLO. To evaluate the model's performance objectively, the creative design also includes a loss function that carefully balances localization and confidence losses.

The ensuing development of YOLO, represented by YOLOv4 and it is a more compact version, Tiny YOLO [5], expands the potential of the initial approach. YOLOv4 brings unique features like a detecting head with more modules, improved feature extraction, and CSPDarknet53. Meanwhile, examples like fire monitoring in smart cities demonstrate how effective lightweight models like MobileNetV3 [10] provide a streamlined solution, especially for real-time applications.

Using YOLOv8 for smart city fire detection, the Fire-YOLO model is a new application in this framework. YOLOv8, which is well-known for its anchor-free detection method, minimizes model size while retaining accuracy [4]. The incorporation of EfficientNet into the Fire-YOLO model mitigates the constraints present in previous models, resulting in enhanced precision, instantaneous detection abilities, and a decrease in false alerts. Notable among these is Fire-YOLO's use of depth-wise separable convolution, which optimizes computational complexity and makes real-time picture processing and deployment easier in resource-constrained contexts.

The YOLO strategy, together with its later variations, is a groundbreaking and revolutionary technology for object detection. Its uses range from generic object recognition to specialized fields like smart city fire detection. YOLOv8 and the Fire-YOLO model are two examples of how YOLO models are still evolving and how well they work to solve the difficulties of real-time detection in a variety of settings.

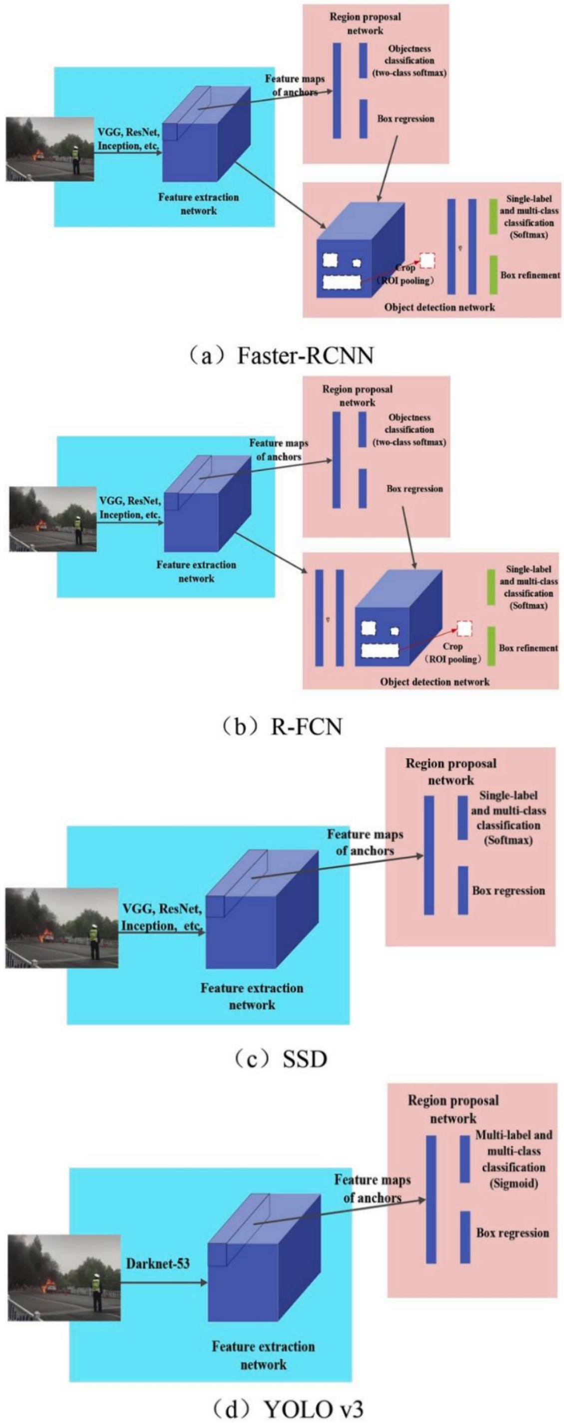

Figure 5. Fire Detecting Algorithms Diagram

F. Research Gap

Existing fire detection systems cannot analyze human behavior for early signs of distress, presenting a critical gap. To address this, this research should focus on incorporating AI algorithms capable of detecting indicators of panic or distress as early signs of fire.

Furthermore, there is a need to enhance mobile notification systems associated with fire alerts. Integrating advanced AI algorithms can streamline and improve the effectiveness of notifications, ensuring swift responses to smoke or fire incidents.

In the realm of image and video analysis, there is an existing research gap in leveraging computer vision and deep learning techniques for real-time monitoring. Exploring these technologies can significantly enhance the accuracy of fire detection by analyzing images or video streams from various sources.

Lastly, a comprehensive exploration of the impact of detection models, building types, indoor space sizes, and camera positioning is essential. Researching these parameters will contribute to the development of tailored fire detection solutions capable of addressing diverse environments and scenarios.

In summary, addressing these research gaps through the integration of AI in behavioral analysis, optimizing notification systems, and leveraging advanced image and video analysis techniques will lead to more effective and adaptable fire detection systems.

III. Deep Learning Approach

A. Dataset Selection

The Flame and Smoke Detection Dataset (FASDD) is an extensive collection of about 95,000 carefully labeled images that show a wide variety of fire and smoke-related events. This dataset has become essential to the creation and assessment of numerous applications for the detection of smoke and flames. To augment the dataset's richness and diversity, additional images are carefully selected from numerous other datasets, guaranteeing a strong portrayal of real-world situations.

The class names for each image in the FASDD are carefully labeled, classifying the instances as either Fire, Smoke, Both Fire and Smoke, or Neither Fire nor Smoke. Table 2 presents the image data count for these labeled categories. This thorough labeling improves the accuracy and dependability of the dataset as a benchmark for fire and smoke detection tasks in addition to making the training and assessment of machine learning models easier.

Given the variety of needs involved in creating strong models, the FASDD has been carefully divided into three subsets: training, testing, and validation datasets. Researchers and practitioners can rigorously train models on one subset, assess their performance on another, and adjust their parameters using the validation set thanks to this strategic division. By taking such care to ensure that the models created with FASDD are accurate and well-suited to new, unseen data, this methodical approach makes FASDD an invaluable tool for advancing the state-of-the-art in smoke and flame detection technology.

B. Image Preprocessing

To optimize training and enable accurate model testing calculations, the images in the FASDD go through a rigorous preprocessing pipeline in which they are scaled to different resolutions according to the needs of different models. The dataset's flexibility in resolution sizes guarantees that it can accommodate a wide variety of models, each with different architectural preferences and computational requirements.

The customization of resolution goes beyond pixel size and includes color representation as well. Because different models have different requirements, some cases in the dataset require a single-channel grayscale representation, while other cases require a more complex three-channel color representation. This strategic approach mirrors the real-world variability in visual data that these models are intended to analyze, while also accommodating the diversity of model architectures.

The dataset images are converted back to their original color channel configurations after the model-specific preprocessing, by the specifications of every model. By ensuring that the dataset maintains its fidelity and relevance during the training and testing stages, this meticulous transformation enables the models to efficiently learn from and generalize across various visual data scenarios. Essentially, this advanced preprocessing pipeline shows a dedication to flexibility, allowing the FASDD to function as a strong and flexible tool for advancing machine learning-based flame and smoke detection.

TABLE I. COMPARISON OF VARIOUS DL MODELS

Index	Class	Algorithm	Missed Detection Rate	False Alarm Rate (%)
1	Manual Feature Extraction	Chen [19]	11.76	14.29
2	CNN	Celik[28]	29.41	0
3		Raffiee [25]	17.65	7.14
4		Habibuglu [23]	5.88	14.29
5		De Lascio [30]	13.33	0
6		Foggia [31]	11.67	0
7		Muhammad [20] (AlexNet)	9.07	2.13
8		Muhammad [13] (GoogleNet)	0.054	1.5
9		Faster R-CNN [1]	0.018	0.69
10		R-FCN	0.018	0.97
11		SDD [8]	0.036	1.29
12		YOLO v3 [27]	0	0.46

C. Transfer Learning

As a result of transfer learning, convolutional neural networks (CNNs) are used to create a vision-based fire and smoke detection system. A pre-trained Faster R-CNN with Inception V2 model is used to apply transfer learning to modify its capabilities for the specific objective of detecting smoke and fire. By using this technique, the model can achieve good detection performance with little data and training time requirements. Our study highlights the significance of this transfer learning method in enhancing the efficacy of the detection model and facilitating its prompt identification and response to the distinct features of smoke and fire in indoor environments. Results show that the Faster R-CNN with Inception V2 is a good fit for the application, with promising outcomes when it comes to detection accuracy and performance evaluation.

Using transfer learning, the model is trained for 90,973 steps, resulting in an average loss of 0.10534, and satisfactory loss function convergence. Following the validation of the detection model on 120 test photos, the results show that the detection model can accurately classify up to 74.73% of fire and 93.85% of smoke. The study obtained 79.75% accuracy for fire and 97.83% accuracy for smoke using the Intersection over Union (IoU) accuracy method. It's interesting to note that the model consistently forecasts with accuracy across a range of environmental contexts and material quantities in video streams.

Furthermore, a review of the detection performance during the testing of video feeds reveals that, respectively, 47.24% and 85.97% of legitimate fire and smoke detections were made. The study shows how adaptable the model is in a range of scenarios, even though smoke detection is not as accurate as fire detection. The confusion matrices illustrate variations in true positive findings between studies, with an average of 48.54% for fire and 0% to 45.10% for smoke. The combined F1 Score for smoke and fire is 0.3088 and 0.5471, respectively. These results demonstrate that the Faster R-CNN with Inception V2 model holds potential for real-time fire and smoke detection, providing gains in accuracy and usability, particularly when paired with efficient transfer learning.

TABLE II. IMAGE STATISTICS

Smoke	Fire	Smoke & Fire	Regular	Total
23,414	12,550	20,151	39,199	95,314

IV. RESULTS AND DISCUSSION

A. *Performance*

The test set 1 is a benchmark fire video database consisting of 31 videos [20]. This database contains 17 non-fire videos and 14 fire videos gathered from various scenarios. Table 1 contains the evaluation results for algorithms 1–8. The publication year, the database used for evaluation, and the fire features used for detection go into choosing the comparison algorithms. The first six algorithms use human selection to extract features, while the seventh and eighth algorithms use machine learning based on CNNs for image classification to extract features automatically. This paper proposes the algorithms 9–12 in Table 3. The findings show that the CNN-based algorithms outperform the conventional method of manually extracting fire features. In this type of less complex scene, such as test set 1, the suggested algorithms based on object detection CNNs can quickly identify the presence of fire. Compared to the other algorithms, the rates of false alarms and missed detections are lower. Nevertheless, the four suggested algorithms find it easy to navigate test set 1, making it impossible to discern the pros and cons of these algorithms. Therefore, using the self-built test set 2, a more thorough evaluation is carried out.

The image fire test set known as test set 2, which contains 14,590 images including 3890 smoke samples and 5322 fire samples, was developed in this paper. The test set 2 is extremely difficult because it gathers images from more scenarios with a lot of fire-like and smoke-like disturbances. As such, it is better suited for assessing how well the suggested algorithms perform.

Figure 6. Image Samples from the Dataset

B. *Detection Time*

Interestingly, the average precision of all four algorithms is remarkably high, highlighting the efficacy of using object detection Convolutional Neural Networks (CNNs) for image fire detection. One-stage algorithms have a significant advantage in terms of detection speed; they can detect more than 15 frames per second and can thus achieve real-time performance.

Among the four algorithms, the YOLO v3-based algorithm performs the best, achieving the highest accuracy of 83.7% [7] and the fastest detection time of 28 frames per second. This algorithm's performance is especially noteworthy. This finding emphasizes YOLO v3's potential as an especially effective and precise option for object detection CNN-based real-time fire detection.

Figure 7. Image Samples from the Dataset

C. *Robustness*

For reliable and accurate detection results, a strong algorithm's stability and dependability must be guaranteed. With bootstrapping in particular, statistical analysis offers important insights into the distribution of test results, including important metrics like average precision and detection rate. These distributions should closely resemble a normal distribution in the case of a well-designed algorithm. Any departure from this pattern could indicate algorithmic problems that could lead to erratic detection results.

TABLE III. APPLICATION OF VARIOUS DL MODELS

Algorithm	AP (%)		mAP (%)	Detection Speed (FPS)
	Smoke	Fire		
Faster-RCNN	79.7	84.9	82.3	3
R-FCN	78.5	83.3	80.9	5
SSD	72.8	82.8	77.8	16
YOLO v3	81.2	87.8	84.5	28

Skewness and kurtosis tests are used to examine the bootstrapped mean average precision (mAP) distribution to investigate the algorithm's stability. The distributions that are shown in Figure 8 provide a visual depiction of the algorithm's performance. A key indicator of instability is a skewness value greater than 1 (DIS>1), signaling a departure from a normal distribution.

The results show that R-FCN and Faster-RCNN-based algorithms display distributions that are suggestive of instability, indicating that improvements are necessary to improve their dependability when it comes to fire detection. On the other hand, algorithms based on SSD and YOLO v3, especially the latter, show stable distributions. This highlights the robustness of these algorithms and shows that they are appropriate for applications that require consistent and dependable fire detection.

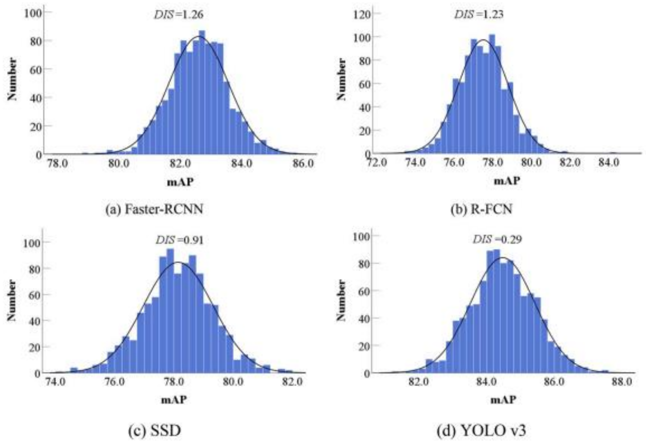

Figure 8. Histogram of the Distribution of Bootstrapping AP

D. *YOLO v7 Training Using the Widest Weight Configuration*

To optimize the model's accuracy potential, the YOLO v7 model training was started from scratch with the widest weight configuration. The model learns to recognize and categorize fire and smoke incidents inside the Flame and Smoke Detection Dataset (FASDD) during 40 epochs of training. It was a calculated move to train the model from scratch rather than with pre-trained weights to customize its learning to the unique properties of smoke and fire in the dataset.

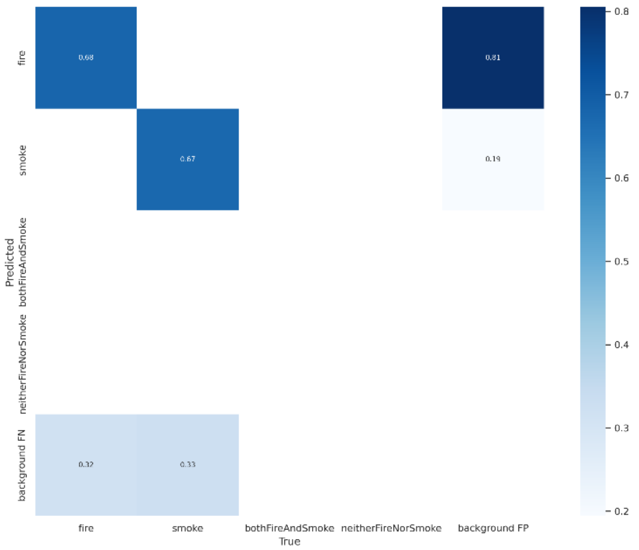

Figure 9. Confusion Matrix of the Trained YOLO v7 Model

E. *Upcoming Enhancements by use of Pre-training*

Even though the existing model has produced encouraging results, using pre-training procedures could increase accuracy. The ability of the model to generalize and adapt to new fire and smoke patterns may be improved by pre-training it on a related dataset. This approach has been successfully used in other studies, including those by Zhang et al. [12], who found that pre-trained models on similar tasks performed better on datasets.

F. *Distribution of Datasets for Training and Validation*

The dataset was carefully split, with 20% designated for validation and 80% for training. With a considerable subset kept aside for testing the model's performance, this distribution made sure that a sizable amount of the data was used for the model's learning. As Huang and Rathod et al. [24] have demonstrated in their seminal work on object detector training, the validation set is essential for fine-tuning the model and avoiding overfitting.

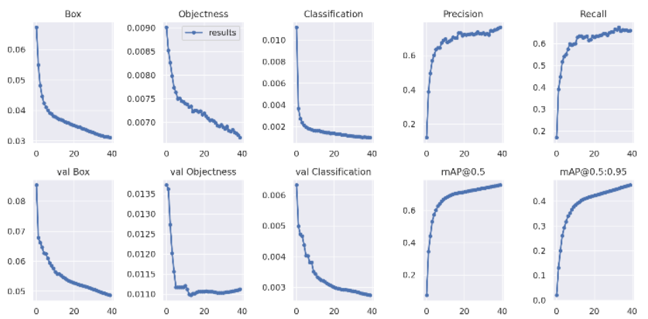

Figure 10. Evaluation Matrices

G. *Writing Code and Preparing Datasets for Processing*

The dataset was preprocessed using unique scripts that included image scaling, normalization, and augmentation methods [18] to increase the variety of the training set. The scripts were also in charge of dividing the dataset into sets for training and validation. This preprocessing stage is essential because it influences how well the model learns from the data; He and Zhang [14] have confirmed this point in their research on deep learning for picture categorization.

To sum up, the initial training results of YOLO v7 suggest that it is feasible to start from scratch with the broadest weight setting. To increase accuracy even more, pre-trained models might be included in training cycles in the future. The model's present success is largely due to the meticulous preprocessing and dataset partition procedures, which will remain a mainstay of continuous efforts to improve the model

Figure 11. Image Samples from the Dataset

V. CONCLUSION

The advanced object detection CNNs of Faster-RCNN, R-FCN, SSD, and YOLO v3 are used to develop image fire detection algorithms to enhance the performance of image fire detection technology. The suggested algorithms are capable of successfully detecting fire in various scenarios and automatically extracting complex image fire features. The following are the results of the evaluation experiments:

A. The four suggested algorithms all achieve high average precision, according to Test set 2 experiments, suggesting that it is completely possible to use object detection CNNs to detect fire in images. Real-time performances of one-stage algorithms can detect more than 15 frames per second.

B. The YOLO v3 algorithm's average precision for fire detection differs statistically significantly from the other three. However, there are no appreciable differences in the average precision of smoke detection between the YOLO v3 and Faster-RCNN algorithms.

C. With 83.7% accuracy, the most accurate algorithm based on YOLO v3 detects fire the fastest (28 FPS) and is the most robust.

References

[1] Barmpoutis, P, et al. "Fire Detection from Images Using Faster R-CNN and Multidimensional Texture Analysis | IEEE Conference Publication | IEEE Xplore." *Ieeexplore.ieee.org*, 2019, ieeexplore.ieee.org/document/8682647. Accessed 19 Nov. 2023.

[2] Guo, Q, et al. "An Improved Faster R-CNN for High-Speed Railway Dropper Detection | IEEE Journals & Magazine | IEEE Xplore." *Ieeexplore.ieee.org*, 2020, ieeexplore.ieee.org/abstract/document/9110596. Accessed 1 Nov. 2023.

[3] Huang, Lida, et al. "Fire Detection in Video Surveillances Using Convolutional Neural Networks and Wavelet Transform." *Engineering Applications of Artificial Intelligence*, vol. 110, Apr. 2022, p. 104737, https://doi.org/10.1016/j.engappai.2022.104737. Accessed 20 Oct. 2023.

[4] F. M. Talaat and Hanaa ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Computing and Applications*, Jul. 2023, doi: https://doi.org/10.1007/s00521-023-08809-1

[5] K.-Y. Wong, "Official YOLOv7," *GitHub*, Sep. 12, 2022. Available: https://github.com/WongKinYiu/yolov7. [Accessed: Nov. 25, 2023]

[6] Lesari, D, et al. *Ieeexplore.ieee.org*, 2019, ieeexplore.ieee.org/abstract/document/8940842. Accessed 10 Nov. 2023.

[7] Li, Pu, and Wangda Zhao. "Image Fire Detection Algorithms Based on Convolutional Neural Networks." Case Studies in Thermal Engineering, vol. 19, June 2020, p. 100625, https://doi.org/10.1016/j.csite.2020.100625. Accessed 19 Nov. 2023.

[8] Nguyen, A, et al. Ieeexplore.ieee.org, 2021, ieeexplore.ieee.org/abstract/document/9352080. Accessed 5 Nov. 2023.

[9] L. Zhao, L. Zhi, C. Zhao, and W. Zheng, "Fire-YOLO: A Small Target Object Detection Method for Fire Inspection," *Sustainability*, vol. 14, no. 9, p. 4930, Apr. 2022, doi: https://doi.org/10.3390/su14094930

[10] ieeexplore.ieee.org/abstract/document/9601362. Accessed 10 Nov. 2023.

[11] S. Wu, "Using Popular Object Detection Methods for Real Time Forest Fire Detection | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. Apr. 25AD. Available: https://ieeexplore.ieee.org/abstract/document/8695482. [Accessed: Nov. 21, 2023]

[12] Zhang, Qi-xing, et al. "Wildland Forest Fire Smoke Detection Based on Faster R-CNN Using Synthetic Smoke Images." *Procedia Engineering*, vol. 211, 2018, pp. 441–446, https://doi.org/10.1016/j.proeng.2017.12.034. Accessed 26 Oct. 2023.

[13] K. Muhammad, J. Ahmad, I. Mehmood, et al., Convolutional neural networks based fire detection in surveillance videos, IEEE Access 6 (2018) 18174–18183.

[14] C. Tao, J. Zhang, P. Wang, Smoke detection based on deep convolutional neural networks, in: 2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2016, pp. 150–153.

[15] A. Filonenko, L. Kurnianggoro, K. Jo, Comparative study of modern convolutional neural networks for smoke detection on image data, in: 2017 10th International Conference on Human System Interactions (HSI), 2017, pp. 64–68.

[16] Z. Yin, B. Wan, F. Yuan, et al., A deep normalization and convolutional neural network for image smoke detection, IEEE ACCESS 5 (2017) 18429–18438.

[17] A.J. Dunnings, T.P. Breckon, Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection, in: 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 1558–1562.

[18] A. Namozov, Y. Cho, An efficient deep learning algorithm for fire and smoke detection with limited data, Adv. Electr. Comput. Eng. 18 (2018) 121–128.

[19] C. Thou-Ho, W. Ping-Hsueh, C. Yung-Chuen, An Early Fire-Detection Method Based on Image Processing, in: 2004 International Conference on Image Processing, vol. 3, 2004, pp. 1707–1710.

[20] K. Muhammad, J. Ahmad, S.W. Baik, Early fire detection using convolutional neural networks during surveillance for effective disaster management, Neurocomputing 288 (2018) 30–42.

[21] Y. Hu, X. Lu, Real-time video fire smoke detection by utilizing spatial-temporal ConvNet features, Multimed. Tool. Appl. 77 (2018) 29283–29301.

[22] A. Namozov, Y. Cho, An efficient deep learning algorithm for fire and smoke detection with limited data, Adv. Electr. Comput. Eng. 18 (2018) 121–128.

[23] Y.H. Habiboglu, O. Günay, A.E. Çetin, Covariance matrix-based fire and flame detection method in video, Mach. Vis. Appl. 23 (2012) 1103–1113.

[24] Y. Luo, L. Zhao, P. Liu, D. Huang, Fire smoke detection algorithm based on motion characteristic and convolutional neural networks, Multimed. Tool. Appl. 77 (2018) 15075–15092.

[25] A. Rafiee, R. Dianat, M. Jamshidi, et al., Fire and smoke detection using wavelet analysis and disorder characteristics, in: 2011 3rd International Conference on Computer Research and Development, 2011, pp. 262–265.

[26] Z. Zhong, M. Wang, Y. Shi, W. Gao, A convolutional neural network-based flame detection method in video sequence, Signal Image Video Process. 12 (2018) 1619–1627.

[27] S. Bianco, R. Cadene, L. Celona, P. Napoletano, Benchmark analysis of representative deep neural network architectures, IEEE Access 6 (2018) 64270–64277.

[28] T. Çelik, H. Demirel, Fire detection in video sequences using a generic color model, Fire Saf. J. 44 (2009) 147–158.

[29] J. Huang, V. Rathod, C. Sun, et al., Speed/accuracy trade-offs for modern convolutional object detectors, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3296–3297.

[30] R. Di Lascio, A. Greco, A. Saggese, M. Vento, in: A. Campilho, M. Kamel (Eds.), Improving Fire Detection Reliability by a Combination of Videoanalytics, Springer International Publishing, Cham, 2014, pp. 477–484.

[31] P. Foggia, A. Saggese, M. Vento, Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion, IEEE T CIRC SYST VID 25 (2015) 1545–1556.