

## Article

# Fire Detection Method in Smart City Environments Using a Deep-Learning-Based Approach

Kuldoshbay Avazov <sup>1</sup>, Mukhiddin Mukhiddinov <sup>1</sup> , Fazliddin Makhmudov <sup>1</sup> and Young Im Cho <sup>2,\*</sup> 

<sup>1</sup> Department of IT Convergence Engineering, Gachon University, Seongnam-Si 461-701, Korea; kuldoshbay@gachon.ac.kr (K.A.); mukhiddinov18@gachon.ac.kr (M.M.); fazliddin12@gachon.ac.kr (F.M.)

<sup>2</sup> Department of Computer Engineering, Gachon University, Seongnam-Si 461-701, Korea

\* Correspondence: yicho@gachon.ac.kr

**Abstract:** In the construction of new smart cities, traditional fire-detection systems can be replaced with vision-based systems to establish fire safety in society using emerging technologies, such as digital cameras, computer vision, artificial intelligence, and deep learning. In this study, we developed a fire detector that accurately detects even small sparks and sounds an alarm within 8 s of a fire outbreak. A novel convolutional neural network was developed to detect fire regions using an enhanced You Only Look Once (YOLO) v4 network. Based on the improved YOLOv4 algorithm, we adapted the network to operate on the Banana Pi M3 board using only three layers. Initially, we examined the original YOLOv4 approach to determine the accuracy of predictions of candidate fire regions. However, the anticipated results were not observed after several experiments involving this approach to detect fire accidents. We improved the traditional YOLOv4 network by increasing the size of the training dataset based on data augmentation techniques for the real-time monitoring of fire disasters. By modifying the network structure through automatic color augmentation, reducing parameters, etc., the proposed method successfully detected and notified the incidence of disastrous fires with a high speed and accuracy in different weather environments—sunny or cloudy, day or night. Experimental results revealed that the proposed method can be used successfully for the protection of smart cities and in monitoring fires in urban areas. Finally, we compared the performance of our method with that of recently reported fire-detection approaches employing widely used performance matrices to test the fire classification results achieved.

**Keywords:** fire detection; smart city; YOLOv4; surveillance system; fire-like lights



**Citation:** Avazov, K.; Mukhiddinov, M.; Makhmudov, F.; Cho, Y.I. Fire Detection Method in Smart City Environments Using a Deep-Learning-Based Approach. *Electronics* **2022**, *11*, 73. <https://doi.org/10.3390/electronics11010073>

Academic Editor: Chiman Kwan

Received: 29 November 2021

Accepted: 22 December 2021

Published: 27 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Fire has contributed to the development of human society in various fields since its inception. However, once a fire gets out of control, it can cause severe damage to human life and property. It is critical to avoid such losses of human life and property. Statistics released by the Korean National Fire Agency revealed that 40,030 fires occurred across South Korea in 2019, resulting in 284 deaths and 2219 injuries. Furthermore, an average of 110 fires and 0.8 fire-related fatalities occurred daily, leading to KRW 2.2 billion in fire property damages. In 2020, more than 50 people were killed in a fire after a 33-story tower block building completely burned down in Ulsan, and a warehouse blaze occurred in Incheon, both incidents in major Korean cities. During the combustion process, fire exhibits many physical properties. Researchers and engineers have developed vision-based fire detectors (VFDs) as well as fire detectors that are sound-sensitive, flame-sensitive, temperature-sensitive, gas-sensitive, or solid-sensitive to address these diverse properties [1]. The chemical properties of smoke are detected by sensors, causing an alarm to sound. However, this approach may result in false alarms. The alarm will not go off until the smoke gets close enough to the sensors to activate them. These detection systems were created as components of traditional alert systems, and they detected certain properties of the flame, such as its smoke and temperature. In some cases, involving broad coverage areas and wild

(forest areas) or high temperatures, a sensor-based detection system [2,3] is impractical because it will result in numerous false warnings. Furthermore, such systems do not provide visual information that would be critical for aiding firefighters in comprehending the situation at a fire incident rapidly. The surroundings can easily interfere with infrared or ultraviolet detectors [4,5], and because of their low detection range, such detectors are not appropriate for broad, open regions. Although satellite remote sensing [6] is effective in identifying large-scale forest fires, it is not capable of detecting early-stage fires. If a real fire occurs at various locations, such as in vast mountains, buildings, and industries, conventional sensor-based fire detection systems can detect them early. To improve the detection of fire in real-time and control false-positive warnings, a VFD system is essential.

The VFD is a unique non-contact fire detector. The VFD can identify a fire from a distance based on the flame data or shape of the smoke of the fire. VFDs have progressively gained popularity with the advancement of computer vision and machine learning, and they have been in use for many years. Moreover, VFDs are more reliable and well-built than traditional fire detection technologies. As a result, VFDs have significantly improved the detection of fires in large areas, such as forests, warehouses, and retail malls. The early identification of fires is essential to emergency prevention. The major objective of VFDs is to achieve a robust detection rate with a low false alarm rate. Furthermore, VFDs are applicable in daytime and nighttime surveillance.

At present, several public areas feature closed-circuit television (CCTV), installed for surveillance and monitoring. CCTV can be employed in fire detection; however, it is not a “smart” system in this regard. As a result, artificial intelligence (AI) techniques must be applied to train the fire detection system, and the system must first recognize the fire before the alarm can be activated. Subsequently, the system will determine the fire intensity and volume. In this scenario, the VFD system can assist in preventing disasters by detecting large and hazardous fires early. To this end, the VFD system provides numerous benefits, including a high accuracy, rapid fire detection, and adjustable custom installation. The VFD and alarm system can be built using a low-cost camera and a small yet robust embedded computer, such as Raspberry Pi, Banana Pi, or Arduino. The input image data will be processed in real-time via AI methods to evaluate the presence of a flame or flame hazard. Image processing algorithms can be applied to detect fire through various approaches. The simplest approach is to use an RGB color model with a threshold because the fire region in an image is typically red. However, this approach is prone to false positives owing to the presence of red colors in the surroundings. Meanwhile, AI- and deep learning (DL)-based image recognition methods can learn and recognize complex image data automatically. Such methods have received considerable attention and produced robust results in diverse applications, including autonomous driving, object detection and recognition, medical diagnostics, and visual search. Several researchers have applied convolutional neural networks (CNNs) and transformers to the field of vision-based fire detection, resulting in the development of models trained in collecting fire image characteristics [7–14]. Many CNNs have been studied for fire detection applications, including faster R-CNN [15], YOLO [16], and SSD [17]. Modern city planning aims to create sophisticated, safe, and sustainable smart cities. Several technologies, such as fire-detection and fire-alarm systems, have to be considered while building AI-based smart cities. There is a high demand for early and accurate fire detection systems to ensure the safety of human life and property. Surveillance cameras are being installed in smart cities to monitor every process, and the use of such cameras for fire detection may present an additional reasonable and reliable solution.

To solve the problem mentioned above, we developed a new fire detection method that can be applied to eliminate disastrous situations, including fire accidents. We improved the performance of the classic YOLOv4 network to enable the detection of fire hazards at the initial stage to reduce the loss of human lives. About 70–80% of fire-related deaths occur owing to toxic gases; therefore, it is important to detect fire scenes rapidly without false alarms using AI methods.

The remainder of the manuscript is organized as follows. Section 2 reviews the related research and introduces the information pertaining to well-known fire detection algorithms. Section 3 describes the proposed fire-detection approach; we describe our dataset, system and training results in detail. Section 4 contains the experimental results, which are evaluated based on the time required for flame recognition and the accuracy rate. Finally, Section 5 provides the conclusions and plans for future study.

## 2. Related Work

Various sensor- and vision-based fire detection systems have been proposed by researchers. Their advantages and disadvantages have also been presented in several review papers [18–21]. In this section, we present state-of-the-art fire detection and segmentation methods that are divided into two categories: DL-based and vision-transformer-based methods.

### 2.1. Fire Detection and Segmentation Method Based on DL

In the last decade, AI and DL approaches have facilitated extensive advancements in several computer vision problems, such as medical diagnosis [22], object detection [23], autonomous driving [24], road monitoring [24], object segmentation [25], and satellite image analysis [26,27]. The comprehensive feature map produced using convolutional layers was primarily responsible for this. DL algorithms were able to identify each pixel in the image efficiently and define the appearance of objects accurately during object segmentation. These algorithms performed more reliably than traditional machine learning models [25].

Numerous studies have been conducted over the years using DL approaches to address fire detection tasks. We have analyzed some of these studies; Muhammad et al. [28] proposed a novel energy-friendly and computationally effective CNN design, motivated by the SqueezeNet [29] structure for the fire detection, localization, and semantic perception of the appearance of fires observed via CCTV surveillance networks. Gonzalez et al. [30] employed CNNs to identify fires in images with a high accuracy and performance, allowing the system to operate in real-time. This system was part of a new unmanned autonomous vehicle detection system for wildfire monitoring and the calculation of location and range. Two networks, AlexNet [31] and a pure CNN, were used to identify the fire features from images. In 2020, novel VFD approaches based on sophisticated CNN models for object identification were proposed by Li et al. [32]. These VFD techniques, which used Faster-RCNN, SSD, and YOLOv3, were compared in terms of their false alarm rates and accuracy. The experimental results revealed that the YOLOv3 algorithm yielded the most robust result with an accuracy of 83.7%.

Encoder–decoder architectures have recently emerged as one of the most popular designs for semantic segmentation. First, the encoder, which contains convolution and pooling layers, produces a high-dimensional feature map from the input images. Second, the decoder, which consists of unpooling and deconvolution layers, decodes the given features and defines the object mask [33]. Owing to the successful performance of various tasks, such as image segmentation, numerous studies have been conducted based on the encoder–decoder structure to segment objects. For example, Zhang et al. [34] presented an efficient DL model based on U-Net and SqueezeNet for forest fire detection and recognition. The proposed framework is divided into two stages: a segmentation module that extracts the shape of a fire and a classification module that determines whether the detected fire area is correct. Owing to its high performance when used in the forest fire detection and segmentation tasks, Akhloufi et al. [35] combined this encoder–decoder with a deep-fire model. This model achieved an FM-score of 97.09% in training and 91% in testing using a small dataset (419 images from the CorsicanFire wildfire dataset as training data) and Dice loss as the loss function [35]. The experimental results confirmed the high effectiveness of the model in segmenting wildfires in the unstructured surroundings wherein uncontrolled forest fires occur. As a refinement of U-Net, Bochkov et al. [36] proposed the UUNet

concatenative model—a new DL model. Binary and multiclass U-Net techniques were combined in this approach. It allows multiclass (color-based) segmentation of signals acquired from the binary part of single-nature objects obtained (fire regions). In addition, a custom fire-image dataset consisting of 6250 samples with a size of  $224 \times 224$  pixels was created, which outperformed the original U-Net by 3% and 2% in multiclass segmentation and binary segmentation, respectively.

Xu et al. [37] introduced a forest fire detection technique based on ensemble learning using CNNs. Two individual object detectors, YOLOv5 [38] and EfficientDet [39], and a classifier based on EfficientNet [40] are combined to localize and detect fires in various scenarios with high accuracies. The pipeline proposed therein achieved a more reliable detection precision of 99.4% using a dataset including 2976 forest-fire images and 7605 forest images without fires. Furthermore, the capacity to identify forest fires in a variety of settings while lowering the rate of false positive alarms was emphasized. The final detection results will be determined using a decision method based on the outcomes of these three learners, which will improve detection accuracy and reduce false positives.

## 2.2. Fire Detection and Segmentation Method Based on Transformers

Many CNN-based methods rely mainly on the convolution operator, which extracts the local features of an image. However, they are restricted in their ability to simulate the global context and have a high computational cost. Transformers have recently been presented as a method to overcome these restrictions by creating long-range interactions between input patches by applying the self-attention method, which constitutes the foundation of transformers. These approaches simulate the relationships between each input feature and other features. They define the global contextual data of each item by considering its interaction with other details. Researchers initially applied transformers to natural language processing problems and obtained robust results. Because of their excellent performance, transformers have since been adopted in computer vision problems, such as object detection and recognition, text segmentation, image classification, and video processing. There are mainly two types of transformers, pure transformers and hybrid transformers (which combine CNNs with a transformer). The quantity of learning data is the most important factor influencing the transformer performance. Dosovitskiy et al. [41] introduced a vision transformer for image classification tasks. When pre-trained at a sufficiently large scale and applied to jobs with fewer data points, the vision transformer produced outstanding results. The vision transformer advances or outperforms the state-of-the-art methods on several image recognition benchmarks when pre-trained on the public ImageNet-21k dataset or the in-house JFT-300M dataset, which contains 300 million labelled images. The top model results, in particular, obtained 88.55% accuracy on ImageNet, 90.72% on ImageNet-Real, 94.55% on CIFAR-100, and 77.63% on the VTAB suite of 19 problems. More recently, Shahid et al. [42] proposed an accurate analysis of the vision transformer model [41] to determine how the framework may be used for vision-based fire detection. The vision transformer does not use convolution and overcomes the drawbacks of CNN-based techniques, including translation invariance and locality. To construct a series, the images were first assembled into patches, which were subsequently flattened and inserted. Position embedding is used to maintain positional knowledge in these patches. The sequence is then placed into multiple multi-head attention layers, which results in the final representation. In the classification layer, the token sequence is passed as the input to a softmax function. Rafik et al. [14] explored the potential of vision transformers in the context of forest fire segmentation using visible spectrum images. This framework improved upon two vision transformers, Medical Transformer [43] and TransUNet [44], for fire detection. They demonstrated the advantages of transformers for fire detection in a variety of scenarios, including a variety of input sizes and backbone models that comprised either pure or hybrid transformers, and the experimental results were obtained using Dice loss as the loss function and the CorsicanFire dataset for training and evaluation. In

addition, IoT-based fire detection methods are also becoming a more active and attractive research area for scholars [45–47].

### 3. Fire Detection and Classification

#### 3.1. Dataset

Our preparation database contained 9200 nighttime and daytime flame pictures, as shown in Table 1. We gathered them from freely accessible databases and Google pictures under various conditions (size, color, and shape). In any case, the normal outcome could not be accomplished with these datasets, and we may not achieve the expected fire recognition accuracy in real-time environments. To resolve this issue, we endeavored to utilize the image transformation method to increase the number of fire-scene pictures in the dataset. The subsequent subsection exhaustively describes the production of the dataset.

**Table 1.** Separation of flame images in the database.

Database	Publicly Available Datasets	Video Frames	Total
Flame Frames	4336	4864	9200

Rotating all the images manually and labeling them again requires considerable processing time. Therefore, we created special software for this using the transformation matrix theory in the OpenCV framework. Image transformation can be expressed in the form of matrix multiplication via affine transformation. Affine transformation is widely used for performing rotation operations, and the following  $2 \times 3$  matrices represent it:

$$M = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \quad (1)$$

The representation of the two-dimensional (2D) vector that we want to transform is as follows.

$$X = \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

We can obtain a transformed vector as follows.

$$T = M * [(x, y, 1)]^T \quad (3)$$

$$M = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \quad (4)$$

In the above matrices,  $T$  indicates the transform vector, and  $M$  indicates the rotation matrix.

The transformation of the matrix can be expressed in terms of the angle of rotation and coordinates of the center, as detailed in [48].

$$\begin{bmatrix} \alpha & \beta & (1 - \alpha) * centre.x - \beta * centre.y \\ -\beta & \alpha & \beta * centre.x + (1 - \alpha) * centre.y \end{bmatrix} \quad (5)$$

In this transformation,  $\alpha = scale * \cos \theta$ ,  $\beta = scale * \sin \theta$ , and  $\theta$  is the rotation angle. We calculated at scale = 1. Usually, the image size and reference point changes after rotation, and the transformation matrix has to be modified. Figure 1 shows the calculation of the new dimensions.

Subsequently, the width and height of the new image size were calculated using the following Equation (6):

$$\begin{aligned} \text{new.width} &= h * \sin(\theta) + w * \cos(\theta) \\ \text{new.height} &= h * \cos(\theta) + w * \sin(\theta) \end{aligned} \quad (6)$$

Finally, as the image resizes, the coordinates of the rotation point (center of the image) also change. The newly obtained image center coordinates are calculated using the following transformation matrix:

$$\begin{bmatrix} \alpha & \beta & (1 - \alpha) * \text{centre.x} - \beta * \text{centre.y} + \left( \frac{\text{new.width}}{2} - \text{centre.x} \right) \\ -\beta & \alpha & \beta * \text{centre.x} + (1 - \alpha) * \text{centre.y} + (\text{new.height}/2 - \text{centre.y}) \end{bmatrix} \quad (7)$$

where  $\alpha = \cos \theta$ ,  $\beta = \sin \theta$ , and  $\theta$  is the rotation angle. While rotating an image, we encountered an aliasing effect and holes in the output image. This problem can be easily solved using the interpolation operation. Unlike our previous research, we rotated all fire pictures by  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ , as shown in Figure 2.

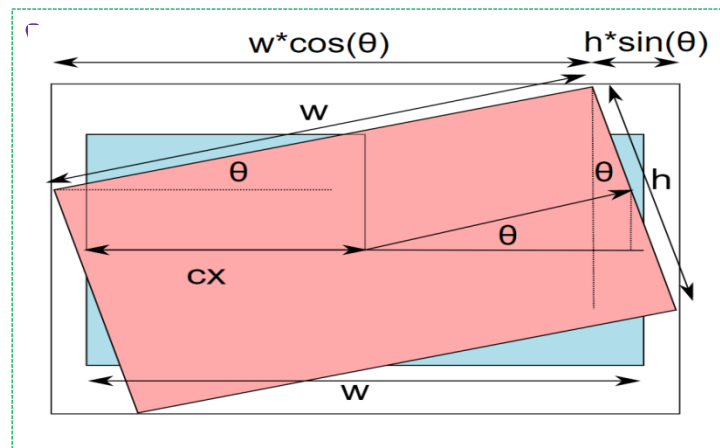


Figure 1. Image rotation schema.

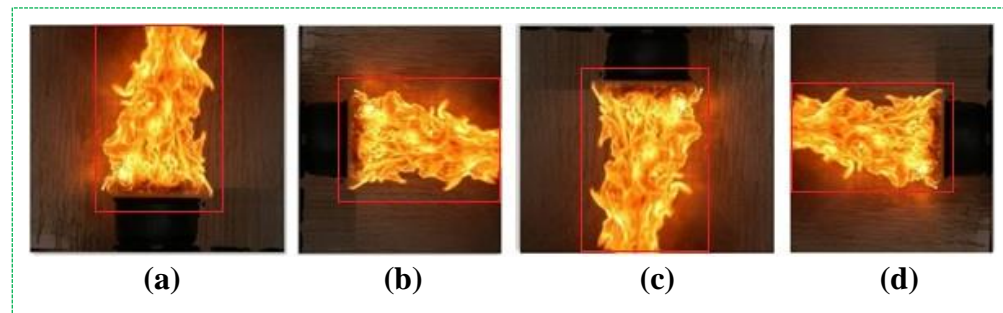


Figure 2. (a) Original image; (b)  $90^\circ$  rotation of (a); (c)  $180^\circ$  rotation of (a), and (d)  $270^\circ$  rotation of (a).

After applying this method, we increased the dataset images by three times the number of original augmented frames. As previously mentioned, our data consisted of 9200 original images. The total size of datasets increased to 27,600 after the effective augmentation method used in this step. Moreover, over 10,000 fire-like images were added to prevent false-positive outcomes, as shown in Table 2. The power of CNN structures fundamentally depends on the number of training dataset images and their resolutions. Consequently, it is necessary to expand the training dataset based on the data augmentation approaches.

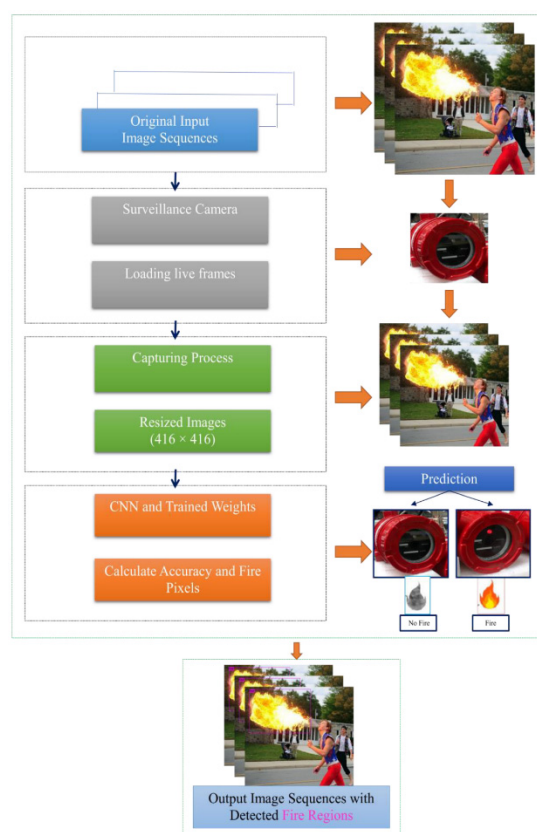
Next, we set the flame in each image according to the YOLO training using the Labelling tool. The tag folder is a TXT document that preserves the flame locations in the frame coordinates. Furthermore, it was utilized in the convolutional neural network as part of the learning procedure. Additionally, we used flame-like pictures to the training set; however, the tag folder comprised free TXT documents. The reason for adding these non-flammable frame sequences during the learning process is to reduce the number of false-positive detections.

**Table 2.** Separation of flame and flame-like images in the database.

Database	Training Images	Testing Images	Total
Flame frames	24,385	3215	27,600
Flame-like images	10,000	0	10,000

### 3.2. System Overview

In this subsection, we provide a short outline of the proposed fire detector for the rapid and accurate identification of fire candidate areas regardless of the size, color, movement, shape, speed, or appearance of the fire. In our methodology, several techniques were developed to accomplish the intended purpose. As shown in Figure 3, we first obtained the original input video sequences from a static camera in real-time. Second, we downsized the original images to resolution of  $416 \times 416$  using the OpenCV framework to avoid complexity within the training and classification task. Smaller images train significantly faster, and may even converge faster, and we will be able to train larger batches using such images. Additionally, using low-dimensional images can help our network generalize better, as there are fewer data to overfit. In our study, we also employed images with  $608 \times 608$  dimensions; they decreased the processing time during training and testing. Notably, large-scale images may occasionally help increase the fire detection accuracy, but processing them can be time consuming.

**Figure 3.** Flowchart outlining the proposed method.

Third, the image enhancement approach was applied to resized images to improve the image quality and increase the number of augmented flame pictures for training the CNN network. Fourth, we operated the network based on pretrained weights by initializing the model. Finally, fire classification accuracy evaluated and predicted the occurrence of a fire. YOLOv4 utilizes the cross-stage partial (CSP) connection with Darknet-53 at the backbone

for feature extraction. At the fire detection and classification stage, the extracted features are compressed and passed through a CNN backbone for predicting fire scores. Based on the surveillance control system, fire accidents were monitored until the fire was recognized. When fire areas are recognized, the red light located on the camera is turned on, and it transmits an emergency alarm to the system operator and firefighting office. The proposed fire detector can be employed for smart city surveillance system applications to enhance the fire safety of the environment. Our method accurately detects and classifies fire regions, even identifying background color intensities that closely resemble the pixel values of the color of the fire.

### 3.3. Fire Detection Process

Three main tasks contribute to the detection of any type of object in real-time scenes with a high speed and accuracy: object classification (type of object), object localization (object located in the image), and object location in the image detection (object position). The YOLO detector is a DL-based approach for object classification. Thus far, five different versions of YOLO have been released: YOLOv1 to YOLOv5. Almost all types of YOLO versions are currently used to identify objects, but all types may not be equally effective in detecting fires. In this study, we used YOLOv4 for fire detection, which was an evolution of the YOLOv3 model. We started our research by examining YOLO networks with the currently available  $9200\ 608 \times 608$  and  $416 \times 416$  fire images. In our previously published paper [16], we achieved a high accuracy using the YOLOv3 network with images of  $608 \times 608$  resolution. However, the training and testing process takes more time than expected because of the large input images and low frames per second (FPS) values. The AP and FPS of YOLOv4 have increased by 10% and 12% (45 FPS) respectively, compared to those of YOLOv3. In addition, several new features were added to build a suitable algorithm for the efficient training and detection of moving objects, which are detailed in the following subsections. First, we evaluated the fire identification performance of YOLOv4 to test the correctness of the prediction of flames with a limited database. As shown in Table 3, the detector was examined using standard variants without any revisions in the calculation stage with 50,000 cycles. The default network employs automatic color augmentation as hue = 0.1, saturation = 1.5, and exposure = 1.5 values to increase the accuracy during the training.

**Table 3.** Constructing pretrained weights using a limited dataset.

Algorithm	Input Size	Training Accuracy (AP50)	Testing Accuracy (AP50)	Weight Size	Number of Iterations	Training Time (h)
YOLOv4	$608 \times 608$	81.1%	74.3%	245 MB	50,000	67 h
YOLOv4	$416 \times 416$	84.5%	79.1%	245 MB		45 h

The results were obtained in terms of the training and testing accuracy with different parameters, as can be seen in Table 3. YOLOv4 yielded the highest result accuracy with an input image size of  $416 \times 416$ . Although the YOLOv4 results are similar, they are not widely used to eliminate fire detection cases; therefore, we chose YOLOv4 from YOLO standard versions to use our research and decided to improve its results using data augmentation. DL methods rely heavily on large datasets; otherwise, the method faces an overfitting problem during the training process. Therefore, to avoid overfitting challenges, dropout and data augmentation methods are usually employed.

The rest of the manuscript resembles the structure of [16], which was published recently. The key difference is that we changed the main network from YOLOv3 to YOLOv4 to use new features and apply a fire detection area with a down sampled image size. To improve accuracy, we modified the dataset and algorithm. We removed low-quality images smaller than  $416 \times 416$  pixels from the database. Detection tasks require better visual



information to increase the classification accuracy. In addition to the data, we duplicated the unlabeled data, which led to errors. During the training, we decided not to use the color, saturation, or exposure parameters in the algorithms automatically: color = 0, saturation = 0, and exposure = 0. In addition, we increased the quantity of data by changing the contrast and brightness by certain values before training the images. Szeliski et al. [49] stated that pixel transformation and local operators can be utilized to deal with pictures. In pixel transformations, the value of each output pixel depends only on the values of the corresponding input pixels. Brightness and contrast are good examples of pixel changes and improved image quality.

$$g(x) = \alpha f(x) + \beta \quad (8)$$

In the above formula,  $\alpha > 0$  and  $\beta$  are commonly referred to as gain and bias parameters, respectively, and these parameters affect the contrast and brightness.  $f(x)$  denotes the source pixel of the image, and  $g(x)$  denotes the output pixel of the image. The following equation can be used to simplify Equation (9):

$$g(i, j) = \alpha f(i, j) + \beta \quad (9)$$

where  $i$  and  $j$  indicate that the pixel is located in the  $i$ -th row and  $j$ -th column. By changing the values of  $\alpha$  (contrast [1–3]) and  $\beta$  (brightness [0–100]), we generated new augmented data in the database. Brightness enhancement is one of the most significant techniques for image improvement during the pre-processing stages. In [50–52], researchers employed global color contrast enhancement and combined local and global contrast enhancement approaches.

We mention in Section 3.1 that we had 27,600 fired picture sequences and 10,000 fire similar scene images. Removing low-quality and low-resolution frames from our final database, we gathered a total of 20,100 good images to use in implementation. After modifying the luminance and chromaticity of the flame frames, we increased the total number of learning pictures around to 80,400, as shown in Table 4. First, we doubled the contrast between the first inserted images. Eventually, we decreased the brightness of the original input image by half.

**Table 4.** Separation of all flame images in the database.

Before	After Filtering	After Contrast Increase (Double)	After Contrast Decrease (Half)
20,100	20,100	20,100	20,100

Using our final dataset, we tested the YOLOv4 network with homogeneous image sizes and iterations. Second experiment results are significantly different from the previous experiment outputs. The improved YOLOv4 detector achieved a better accuracy than previous detectors. However, the training process in this approach takes more time because the number of dataset images increases considerably, as summarized in Table 5.

**Table 5.** Constructing pretrained weights using YOLOv4.

Algorithm	Input Size	Training Accuracy (AP50)	Testing Accuracy (AP50)	Weight Size	Number of Iterations	Training Time
YOLOv4	608 × 608	95.1%	94.7%	245 MB	50,000	87 h
YOLOv4	416 × 416	98.8%	96.3%	245 MB		72 h

Furthermore, we tested all YOLO network versions on the augmented fire dataset (80,400 images) and compared the final accuracies; Table 6 reveals that YOLOv4 ranked the highest in training with 98.8% accuracy. Furthermore, YOLOv3 achieved 97.1% (a difference

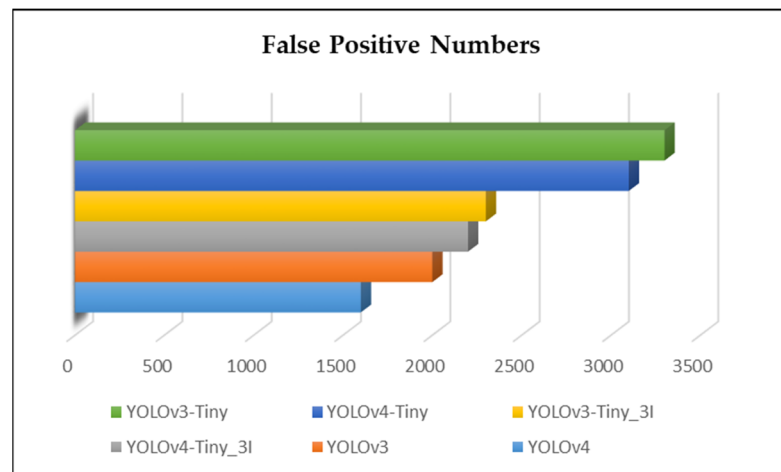
of 1.7% from YOLOv4) and marginally trailed YOLOv4 in terms of the testing accuracy. This was followed by YOLOv4tiny and YOLOv4tiny\_3l. These algorithms consumed more time than those in previous experiments because of the increased number of dataset images. YOLOv4 was considered as an efficient and powerful fire detection model with the highest prediction accuracy, even when the processing time was less than that of the YOLOv4\_tiny algorithm.

**Table 6.** Comparison of all YOLO networks based on a large dataset.

Algorithms	Input Size	Training Accuracy (AP50)	Testing Accuracy (AP50)	Weight Size	Number of Iterations	Training Time
YOLOv4	416 × 416	98.8%	96.3%	245 MB	50,000	72 h
YOLOv4-tiny_3l		89.9%	85.7%	57 MB		37 h
YOLOv4-tiny		93.8%	91.2%	85 MB		43 h
YOLOv3		97.1%	94.6%	236 MB		82 h
YOLOv3-tiny_3l		92.6%	90.4%	63.7 MB		39 h
YOLOv3-tiny		85.3%	82.7%	53 MB		37.5 h

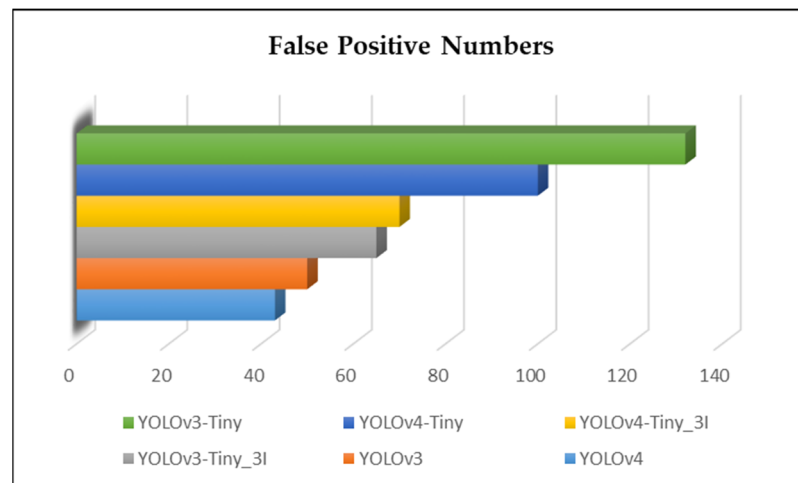
As mentioned above, owing to fire scenes, several errors occur in real-time fire detection at the wrong signal. In this research, we used 3000 sunrise and sunset images in the training and not the testing steps of the model. This is because sunlight pixel values are very close to fire color intensities, even though they are not real fires.

We examined the training weights in both experiments using a database of images with no fires. YOLO networks “evaluate” regions based on their similarity to predefined classes. Areas with the highest scores were recorded as positive, regardless of which class they were closest to. The results are shown in Figure 4.



**Figure 4.** False-positive results in the weight files of the first experiment.

It can be seen in Figure 4 that the weight files trained through the default algorithms yielded a lot of mistakes. Although the fewest mistakes were exhibited by the YOLOv4 method, they are classified as major false positives. After adding fire-like images to the dataset, the weight files were trained again, and the experiments revealed a 20-fold reduction in errors compared to those in the default algorithms, as shown in Figure 5.



**Figure 5.** False-positive results in the weight files of the second experiment.

Although we achieved the expected 98.8% score at this stage, we attempted to increase the final accuracy using ideas from recently published papers. We deduced from [53] that the detection of small flame regions is not easy and that most approaches cannot detect them in a timely manner. To overcome this difficulty, we produced small-scale fire images to improve the classification accuracy, as detailed in [16]. We used a large-scale feature map to identify small moving objects and combine them with the feature map in the previous layers, which helped preserve the fine-grained features, as mentioned in [32]. This extensive feature map, which contains the location data of the previous layers and the complex properties of the deeper layers, has been used to recognize small-sized fire regions. Another advantage of our method is that it can analyze the intentions of humans as well as the safety of generated fire through AI. Fire detectors can make rapid decisions in emergency situations and ignore alarms when cameras detect fires created by humans.

We increased the fire detection accuracy to 99.8%. With this result, we can detect any type of fire in the early stage, even small-scale ones. Finally, we needed to implement our method on the Banana Pi M3(BPI M3) board to enable the use of the proposed method with a small convolutional neural network to reach a reduction in test process time and loss of accuracy. The large CNN operating on the BPI M3 results in a slow performance and is practically complicated. To accomplish this work, we utilized three layers of the improved YOLOv4. In the next section, we compare the traditional in-depth study frameworks with proposed method for efficiency and effectiveness analyses. To achieve this objective, we trained our network using images depicting artificial fires that people use in their daily lives. Providing a “golden time” for an initial response is the most important element of our approach in dealing with and identifying catastrophic situations.

#### 4. Experimental Results and Discussions

We constructed and examined the proposed system in the Anaconda 2020 Python distribution on a PC with a 3.20 GHz CPU, 32 GB RAM, and two Nvidia GeForce 1080Ti GPUs (Nvidia, Santa Clara, CA, USA). To estimate tkuldoshbay@gachon.ac.kr performance of the fire detection method, the model was tested on various indoor and outdoor fire scenarios. The previous section discusses several experiments that were conducted and implemented using YOLO classifier systems. Here, we focus on the analysis of the execution of conventional fire detection techniques and the proposed approach in terms of qualitative and quantitative results. Figures 6 and 7 show examples of visible experiments in daytime and nighttime environments using the CSPDarknet-53 feature extractor. CSPDarknet-53 is a CNN and backbone for object detection that uses DarkNet-53 [54,55]. It employs a CSPNet strategy to partition the feature map of the base layer into two parts and

subsequently merges them through a cross-stage hierarchy. The use of a split-and-merge strategy allows for more gradient flow through the network.



**Figure 6.** Visible experiment results in daytime fire scenes: (a) input image sequences and (b) output image sequences with detected fire regions.



**Figure 7.** Visible experiment results in nighttime fire scenes: (a) input image sequences and (b) output image sequences with detected fire regions.

Experiments have demonstrated that our improved fire detector can relieve people's anxiety, and it enables the early suppression and rapid response irrespective of the time of day or the shape or size of the fire. It is difficult to prevent fire accidents at initial stages because of the false alarms by traditional fire detectors, such as those raised when moving objects exhibiting similar color and pixel intensity values to those of fire regions in the background area.

To evaluate and analyze the effectiveness of fire recognition cases, we compared the proposed approach with recently published fire detection methods. To perform this task, we employed widely used estimation metrics for object (static or dynamic) detection, as in our previous publications [16,56]. First, we computed the precision and recall metrics to measure the performance of an object detector by determining whether detection is valid. Precision is the ability of a classifier to identify only the relevant objects; it is the proportion of true positives detected. Recall measures the ability of the model to identify all relevant cases; it is the proportion of true positives detected among all ground truths. A

good model is one that can identify most ground-truth objects (exhibits high recall) while identifying only the relevant objects (exhibits high precision). A perfect model is one with a false-negative value = 0 (recall = 1) and a false-positive value = 0 (precision = 1). We used following equations to estimate the average precision and recall measures of the fire recognition systems:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \end{aligned} \quad (10)$$

where  $TP$  means true positives which denote accurately found flame pixels,  $FP$  means false positives, and  $FN$  indicates the number of false negatives.

Evaluating an object detection model using precision and recall can provide valuable insights into how the model performs at various confidence values. The F-measure (FM) score is particularly helpful in determining the optimum confidence that balances the precision and recall values for testing methods and can be evaluated using Equation (11), which considers both precision and recall.

$$\text{FM} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (11)$$

In general, the use of a validation set (a set of data used to adjust the hyper-parameters) and test set (a set of data used to evaluate the performance of a fully trained model) is recommended to analyze the performance of models. The classification function only evaluates the probability that a class object will appear in the image, which is a simple task for the classifier to determine the correct predictions from incorrect predictions. However, the object detection task further localizes the object through a boundary box associated with a corresponding confidence score to report how accurately the object class boundary box is defined [57]. The performance of our approach and other recently published fire detection techniques is provided in Table 7.

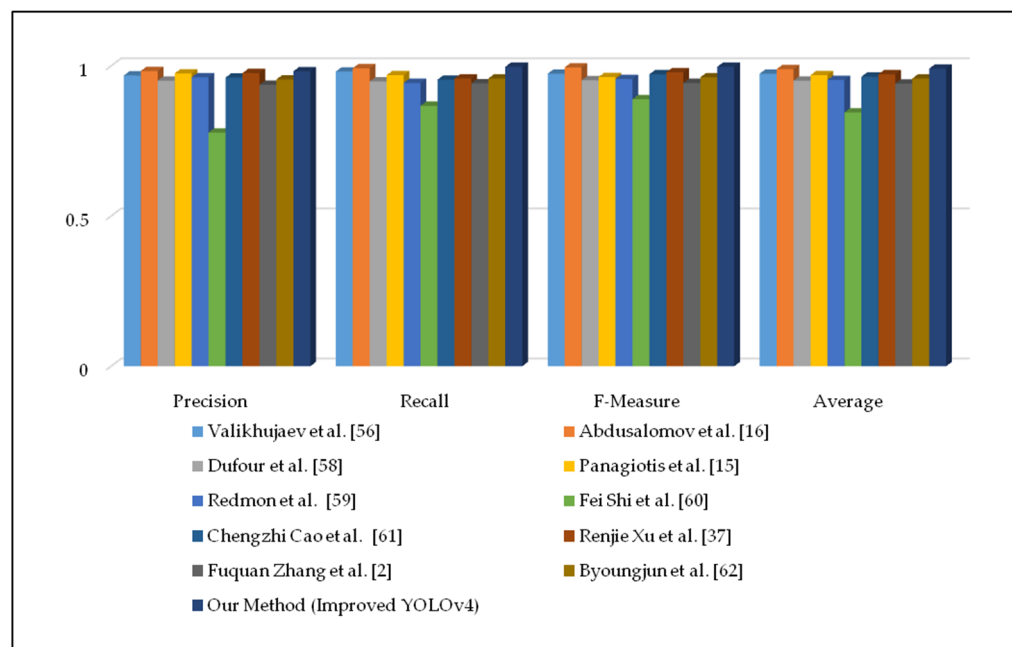
**Table 7.** Quantitative results of fire detection approaches.

<i>Approach</i>	<i>Precision</i>	<i>Recall</i>	<i>FM</i>	<i>Average</i>
Valikhujaev et al. [56]	0.968	0.981	0.974	<b>0.974</b>
Abdusalomov et al. [16]	0.983	0.992	0.995	<b>0.989</b>
Dufour et al. [58]	0.951	0.948	0.952	<b>0.951</b>
Panagiotis et al. [15]	0.975	0.969	0.963	<b>0.969</b>
Redmon et al. [59]	0.962	0.943	0.956	<b>0.953</b>
Fei Shi et al. [60]	0.778	0.867	0.889	<b>0.845</b>
Chengzhi Cao et al. [61]	0.961	0.954	0.972	<b>0.964</b>
Renjie Xu et al. [37]	0.976	0.958	0.979	<b>0.972</b>
Fuquan Zhang et al. [2]	0.937	0.942	0.943	<b>0.941</b>
Byoungjun et al. [62]	0.955	0.958	0.962	<b>0.958</b>
<b>Our Method (Improved YOLOv4)</b>	0.982	0.997	0.997	<b>0.991</b>

In Table 7, we compare the experimental performance of our method with those of existing approaches in terms of three metrics: precision, recall, and FM. Based on the results of the analyzed methods, our methods perform the best in terms of FPs and FNs.

In addition, based on Table 7, we finalized the average accuracy using the precision, recall, and FM metrics, as shown in Figure 8. The improved fire detector yielded an approximate 99.1% accuracy, while other approaches yielded accuracies approximating 95%. We used the results in their papers for comparison; however, the accuracy of these values is not easily verifiable because many articles are written on a project basis, and the source codes and datasets of the papers are not publicly uploaded to the Internet. In the

case of normal scenes, our method has proven the high accuracy of early fire detection by reducing the computational time with correct final decisions even when the weather is cloudy or rainy.



**Figure 8.** Quantitative results of fire detection approaches using vertical graphs.

## 5. Conclusions

This manuscript presents a new approach that uses the YOLO4 network to detect and classify suspected regions of fire without delayed final decisions. We employed several techniques to improve the accuracy of the default YOLOv4 method in fire detection and achieved a high precision. A technique was created to consequently move labeled bounded boxes at each instance of turning fire dataset images by  $15^\circ$ . In addition, we created large fire dataset images that depicted different fire and fire-like scenarios (daytime and nighttime images) that will be freely available to everyone for research purposes. Important features in the in-depth CNN were explored using large databases to make accurate predictions and control overfitting issues. Another achievement of our method is that we implemented it on a BPI M3 board which allowed us to reduce the processing time of CPU and GPU tools compared to conventional flame recognition approaches. Current research shows that it is important to detect fire accidents rapidly at their initial stages to increase safety in our daily lives.

Therefore, we aim to continue our research in this area and improve our results further. Our objective is to use the YOLOv5 and YOLACT models to identify fire incidents in real-time with a reduced number of false positives.

**Author Contributions:** This manuscript was designed and written by K.A. K.A. conceived the main idea of this study. M.M. wrote the program in Python and conducted all experiments. F.M. and Y.I.C. supervised the study and contributed to the analysis and discussion of the algorithm and the experimental results. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2021-2017-0-01630) supervised by the IITP and 2018R1D1A1A09084151 by NRF, and K\_G012000 supported by Korea Agency for Technology and Standards.

**Acknowledgments:** The first author K.A. would like to express his sincere gratitude and appreciation to the supervisor, Young Im Cho (Gachon University) for her support, comments, remarks, and engagement over the period in which this manuscript was written. Moreover, the authors would like to thank the editor and anonymous referees for the constructive comments in improving the contents and presentation of this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Hong, X.; Wang, W.; Liu, Q. Design and Realization of Fire Detection Using Computer Vision Technology. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 5645–5649.
2. Zhang, F.; Zhao, P.; Xu, S.; Wu, Y.; Yang, X.; Zhang, Y. Integrating multiple factors to optimize watchtower deployment for wildfire detection. *Sci. Total Environ.* **2020**, *737*, 139561. [[CrossRef](#)]
3. Zhang, F.; Zhao, P.; Thiyaalingam, J.; Kirubarajan, T. Terrain-influenced incremental watchtower expansion for wildfire detection. *Sci. Total Environ.* **2018**, *654*, 164–176. [[CrossRef](#)]
4. Lee, B.; Kwon, O.; Jung, C.; Park, S. The development of UV/IR combination flame detector. *J. KIIS* **2001**, *16*, 1–8.
5. Kang, D.; Kim, E.; Moon, P.; Sin, W.; Kang, M. Design and analysis of flame signal detection with the combination of UV/IR sensors. *J. Korean Soc. Int. Inf.* **2013**, *14*, 45–51.
6. Fernandes, A.M.; Utkin, A.B.; Lavrov, A.V.; Vilar, R.M. Development of neural network committee machines for automatic forestfire detection using lidar. *Pattern Recognit.* **2004**, *37*, 2039–2047. [[CrossRef](#)]
7. Muhammad, K.; Ahmad, J.; Mehmood, I.; Rho, S.; Baik, S.W. Convolutional Neural Networks Based Fire Detection in Surveillance Videos. *IEEE Access* **2018**, *6*, 18174–18183. [[CrossRef](#)]
8. Tao, C.; Zhang, J.; Wang, P. Smoke detection based on deep convolutional neural networks. In Proceedings of the 2016 International Conference on Industrial Informatics -Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan, China, 3–4 December 2016; pp. 150–153.
9. Filonenko, A.; Kurnianggoro, L.; Jo, K.-H. Comparative study of modern convolutional neural networks for smoke detection on image data. In Proceedings of the 2017 10th International Conference on Human System Interactions (HSI), Ulsan, Korea, 17–19 July 2017; pp. 64–68.
10. Yin, Z.; Wan, B.; Yuan, F.; Xia, X.; Shi, J. A Deep Normalization and Convolutional Neural Network for Image Smoke Detection. *IEEE Access* **2017**, *5*, 18429–18438. [[CrossRef](#)]
11. Dunning, A.J.; Breckon, T.P. Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection. In Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1558–1562.
12. Mao, W.; Wang, W.; Dou, Z.; Li, Y. Fire Recognition Based on Multi-Channel Convolutional Neural Network. *Fire Technol.* **2018**, *54*, 531–554. [[CrossRef](#)]
13. Hu, Y.; Lu, X. Real-time video fire smoke detection by utilizing spatial-temporal ConvNet features. *Multimed. Tools Appl.* **2018**, *77*, 29283–29301. [[CrossRef](#)]
14. Ghali, R.; Akhloufi, M.A.; Jmal, M.; Souidene Mseddi, W.; Attia, R. Wildfire Segmentation Using Deep Vision Transformers. *Remote Sens.* **2021**, *13*, 3527. [[CrossRef](#)]
15. Barmoutis, P.; Dimitropoulos, K.; Kaza, K.; Grammalidis, N. Fire Detection from Images Using Faster R-CNN and Multidimensional Texture Analysis. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 8301–8305. [[CrossRef](#)]
16. Abdusalomov, A.; Baratov, N.; Kutlimuratov, A.; Whangbo, T.K. An Improvement of the Fire Detection and Classification Method Using YOLOv3 for Surveillance Systems. *Sensors* **2021**, *21*, 6519. [[CrossRef](#)]
17. Nguyen, A.Q.; Nguyen, H.T.; Tran, V.C.; Pham, H.X.; Pestana, J. A Visual Real-time Fire Detection using Single Shot MultiBox Detector for UAV-based Fire Surveillance. In Proceedings of the 2020 IEEE Eighth International Conference on Communications and Electronics (ICCE), IEEE, Phu Quoc Island, Vietnam, 13–15 January 2021; pp. 338–343.
18. Gaur, A.; Singh, A.; Kumar, A.; Kumar, A.; Kapoor, K. Video Flame and Smoke Based Fire Detection Algorithms: A Literature Review. *Fire Technol.* **2020**, *56*, 1943–1980. [[CrossRef](#)]
19. Bu, F.; Gharajeh, M.S. Intelligent and vision-based fire detection systems: A survey. *Image Vis. Comput.* **2019**, *91*, 103803. [[CrossRef](#)]
20. Fonollosa, J.; Solórzano, A.; Santiago, M. Chemical sensor systems and associated algorithms for fire detection: A review. *Sensors* **2018**, *18*, 553. [[CrossRef](#)]
21. Mahdipour, E.; Dadkhah, C. Automatic fire detection based on soft computing techniques: Review from 2000 to 2010. *Artif. Intell. Rev.* **2012**, *42*, 895–934. [[CrossRef](#)]
22. Bakator, M.; Radosav, D. Deep Learning and Medical Diagnosis: A Review of Literature. *Multimodal Technol. Interact.* **2018**, *2*, 47. [[CrossRef](#)]
23. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)]

24. Abdollahi, A.; Pradhan, B.; Shukla, N.; Chakraborty, S.; Alamri, A. Deep Learning Approaches Applied to Remote Sensing Datasets for Road Extraction: A State-Of-The-Art Review. *Remote Sens.* **2020**, *12*, 1444. [CrossRef]
25. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *1*. [CrossRef]
26. Pan, Z.; Xu, J.; Guo, Y.; Hu, Y.; Wang, G. Deep Learning Segmentation and Classification for Urban Village Using a World-view Satellite Image Based on U-Net. *Remote Sens.* **2020**, *12*, 1574. [CrossRef]
27. Bragilevsky, L.; Bajić, I.V. Deep learning for Amazon satellite image analysis. In Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, 21–23 August 2017; pp. 1–5.
28. Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1419–1434. [CrossRef]
29. Iandola, F.N.; Moskewicz, M.W.; Ashraf, K.; Han, S.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <1 MB model size. *arXiv* **2016**, arXiv:1602.07360.
30. Gonzalez, A.; Zuniga, M.D.; Nikulin, C.; Gonzalo, C.; Cardenas, D.G.; Pedraza, M.A.; Fernandez, C.A.; Munoz, R.I.; Castro, N.A.; Rosales, B.F.; et al. Accurate fire detection through fully convolutional network. In Proceedings of the 7th Latin American Conference on Networked and Electronic Media (LACNEM 2017), Valparaiso, Chile, 6–7 November 2017; pp. 1–6.
31. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
32. Li, P.; Wangda, Z. Image fire detection algorithms based on convolutional neural networks. *Case Stud. Therm. Eng.* **2020**, *19*, 100625. [CrossRef]
33. Xing, Y.; Zhong, L.; Zhong, X. An Encoder-Decoder Network Based FCN Architecture for Semantic Segmentation. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8861886. [CrossRef]
34. Zhang, J.; Zhu, H.; Wang, P.; Ling, X. ATT Squeeze U-Net: A Lightweight Network for Forest Fire Detection and Recognition. *IEEE Access* **2021**, *9*, 10858–10870. [CrossRef]
35. Akhloofi, M.A.; Tokime, R.B.; Ellassady, H. Wildland fires detection and segmentation using deep learning. Pattern recognition and tracking xxix. *Int. Soc. Opt. Photonics Proc. SPIE* **2018**, *10649*, 106490B.
36. Bochkov, V.S.; Kataeva, L.Y. wUUNet: Advanced Fully Convolutional Neural Network for Multiclass Fire Segmentation. *Symmetry* **2021**, *13*, 98. [CrossRef]
37. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. *Forests* **2021**, *12*, 217. [CrossRef]
38. Jocher, G.; Stoken, A.; Chaurasia, A.; Borovec, J.; Chanvichet, V.; Kwon, Y.; TaoXie, S.; Changyu, L.; Abhiram, V.; Skalski, P.; et al. YOLOv5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 20 September 2021).
39. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
40. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
41. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
42. Shahid, M.; Hua, K.-L. Fire Detection using Transformer Network. In Proceedings of the Proceedings of the 2021 International Conference on Multimedia Retrieval. *ACM* **2021**, 627–630.
43. Valanarasu, J.M.J.; Oza, P.; Hacihaliloglu, I.; Patel, V.M. Medical Transformer: Gated Axial-Attention for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.10662.
44. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.04306.
45. Marcu, A.-E.; Suci, G.; Olteanu, E.; Miu, D.; Drosu, A.; Marcu, I. IoT system for forest monitoring. In Proceedings of the 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, 1–3 July 2019.
46. Zhou, H.; Taal, A.; Koulouzis, S.; Wang, J.; Hu, Y.; Suci, G.; Poenaru, V.; Laat, C.D.; Zhao, Z. Dynamic real-time infrastructure planning and deployment for disaster early warning systems. In *Proceedings of the International Conference on Computational Science*; Springer: Cham, Switzerland, 2018.
47. Ghosh, S.; Mukherjee, A.; Ghosh, S.K.; Buyya, R. Mobi-iot: Mobility-aware cloud-fog-edge-iot collaborative framework for time-critical applications. *IEEE Trans. Netw. Sci. Eng.* **2019**, *7*, 2271–2285. [CrossRef]
48. Lee, S.J.; Kim, B.H.; Kim, M.Y. Multi-Saliency Map and Machine Learning Based Human Detection for the Embedded Top-View Imaging System. *IEEE Access* **2021**, *9*, 70671–70682. [CrossRef]
49. Szeliski, R. Computer Vision Algorithms and Applications. 2011. Available online: <http://dx.doi.org/10.1007/978-1-84882-935-0> (accessed on 25 October 2021).
50. Abdusalomov, A.; Whangbo, T.K. An improvement for the foreground recognition method using shadow removal technique for indoor environments. *Int. J. Wavelets Multiresolution Inf. Process.* **2017**, *15*, 1750039. [CrossRef]
51. Abdusalomov, A.; Whangbo, T.K. Detection and Removal of Moving Object Shadows Using Geometry and Color Information for Indoor Video Streams. *Appl. Sci.* **2019**, *9*, 5165. [CrossRef]
52. Abdusalomov, A.; Mukhiddinov, M.; Djuraev, O.; Khamdamov, U.; Whangbo, T.K. Automatic Salient Object Extraction Based on Locally Adaptive Thresholding to Generate Tactile Graphics. *Appl. Sci.* **2020**, *10*, 3350. [CrossRef]



53. Barmpoutis, P.; Papaioannou, P.; Dimitropoulos, K.; Grammalidis, N. A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing. *Sensors* **2020**, *20*, 6442. [[CrossRef](#)]
54. Redmon, J. Darknet: Open-Source Neural Networks in C. 2013–2016. Available online: <http://pjreddie.com/darknet/> (accessed on 22 August 2021).
55. Bochkovskiy, A.; Wang, C.-Y.; Mark Liao, H.-Y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
56. Valikhujaev, Y.; Abdusalomov, A.; Cho, Y. Automatic Fire and Smoke Detection Method for Surveillance Systems Based on Dilated CNNs. *Atmosphere* **2020**, *11*, 1241. [[CrossRef](#)]
57. Yolo v5. In GitHub. 2021. Available online: <https://github.com/ultralytics/yolov> (accessed on 31 May 2021).
58. Dufour, D.; Le Noc, L.; Tremblay, B.; Tremblay, M.N.; Généreux, F.; Terroux, M.; Vachon, C.; Wheatley, M.J.; Johnston, J.M.; Wotton, M.; et al. A Bi-Spectral Microbolometer Sensor for Wildfire Measurement. *Sensors* **2021**, *21*, 3690. [[CrossRef](#)] [[PubMed](#)]
59. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
60. Shi, F.; Qian, H.; Chen, W.; Huang, M.; Wan, Z. A Fire Monitoring and Alarm System Based on YOLOv3 with OHEM. In Proceedings of the 2020 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020; pp. 7322–7327. [[CrossRef](#)]
61. Cao, C.; Tan, X.; Huang, X.; Zhang, Y.; Luo, Z. Study of Flame Detection based on Improved YOLOv4. *J. Phys.* **2021**, *1952*. [[CrossRef](#)]
62. Kim, B.; Lee, J. A Video-Based Fire Detection Using Deep Learning Models. *Appl. Sci.* **2019**, *9*, 2862. [[CrossRef](#)]