Article

A Study on a Complex Flame and Smoke Detection Method Using Computer Vision Detection and Convolutional Neural Network

Jinkyu Ryu and Dongkurl Kwak *

Graduate School of Disaster Prevention, Kangwon National University, Samcheok 25913, Korea; jkryu@kangwon.ac.kr

* Correspondence: dkkwak@kangwon.ac.kr; Tel.: +82-33-570-6823

Abstract: This study sought an effective detection method not only for flame but also for the smoke generated in the event of a fire. To this end, the flame region was pre-processed using the color conversion and corner detection method, and the smoke region could be detected using the dark channel prior and optical flow. This eliminates unnecessary background regions and allows selection of fire-related regions. Where there was a pre-processed region of interest, inference was conducted using a deep-learning-based convolutional neural network (CNN) to accurately determine whether it was a flame or smoke. Through this approach, the detection accuracy is improved by 5.5% for flame and 6% for smoke compared to when a fire is detected through the object detection model without separate pre-processing.

Keywords: fire safety system; computer vision; image processing; deep learning

Citation: Ryu, J.; Kwak, D. A Study on a Complex Flame and Smoke Detection Method Using Computer Vision Detection and Convolutional Neural Network. *Fire* **2022**, *5*, 108. https://doi.org/10.3390/fire5040108

Academic Editor: James A. Lutz

Received: 24 June 2022
Accepted: 26 July 2022
Published: 27 July 2022

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

Copyright: © 2022 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

In a fire, engineering approaches to reducing the spread of smoke and flames involve compartmentalization, dilution, airflow, and pressurization [1]. These methods are very important for extinguishing a fire in its early stages, but there is a problem in that the fire has already begun to grow or become responsive after full development. Therefore, to solve this problem, this study attempts an image-based fire detection method. In particular, it aims to effectively respond to a fire by detecting all the flame and smoke that may occur in the early stages of the fire. It is very important to detect smoke, not just flames, in a fire, particularly considering that in general, smoke damage to human health occurs more often than direct damage caused by flames. Smoke generated from such a fire can affect the human body due to high temperatures, lack of oxygen, and carbon monoxide. In addition to these direct factors, reduced visibility and subsequent psychological anxiety may adversely affect evacuation behavior [2,3].

To this end, many studies on fire detection based on artificial intelligence have recently been conducted. Existing deep learning computer vision-based flame detection studies have included a method proposed by Shen et al. [4], in which flames were detected using a "you only look once" (YOLO) model based on Tensorflow, without separate filtering of input images. In this case, if additional image pre-processing was added, the accuracy would be improved, as the unnecessary background area would be removed in advance, reducing false negatives significantly. Another study was a fire detection method proposed by Muhammad et al. [5], which classified fire or non-fire from the image in order to efficiently detect fire in resource-constrained environments. However, in this case, the fire is judged not only for the flame but also for the entire unnecessary image area. Nguyen et al. [6] achieved 92.7% accuracy in detecting fire using a UAV-based object detection algorithm, while Jeon et al. [7] achieved a 97.9% F1 score by detecting fire using a CNN based on a multi-scale prediction framework, but both of these previous studies detected only flame.

However, there is a problem of high error detection rate because there is no separate filtering method. Therefore, in this study, the disadvantages of the existing image-based fire detection methods were supplemented through color and dynamic characteristics. In particular, when detection is difficult because the shape is not constant, such as with smoke, detection is facilitated through an appropriate pre-processing method. In addition, both of these previous studies detected only flame. Therefore, to supplement these fire detection methods, this study proposes a method capable of detecting flame and smoke in combination.

Therefore, and as shown in Figure 1, an effective image pre-processing method was designed for each flame, as well as for the smoke from the input image, so that objects that are not related to the fire can be filtered out in advance. In the case of a flame, when a combustible gas generated by pyrolysis of a solid combustible material such as wood is mixed with air and combusted, a flame may be generated. In addition, a flame may be generated by evaporative combustion, in which the combustible liquid evaporates and burns. When such flaming occurs, pre-processing was attempted in order to detect the flame through its appearance-related characteristics. For this purpose, hue, saturation, value (HSV) color conversion and corner detection were used during image pre-processing. First, in the case of HSV color conversion, a color region where a flame is likely to exist is detected. In addition, among the objects that remain after HSV color conversion, the flame is found to possess the characteristic of a sharp texture of the object, resulting in a large number of corners [8,9]. Based on this fact, if the Harris corner detector is performed on the HSV color converted image, corners are generated intensively only in the flame, so that filtering can be performed more precisely. Therefore, following the color conversion and filtering, the region where the corners are gathered is detected as a flame candidate region.

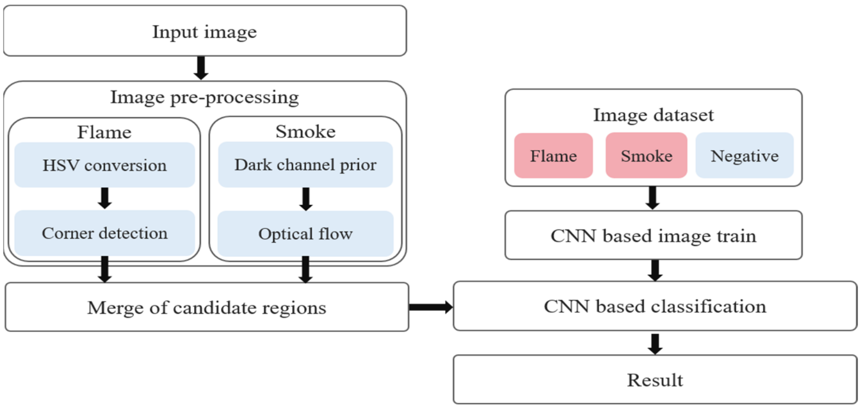


Figure 1. Flow chart of the proposed fire detection.

In this study, both the optical flow technique, which used the dark channel prior, and the Lucas–Kanade method were used to effectively pre-process smoke. The dark channel prior was originally proposed by He et al. [10] as an algorithm designed to remove haze from the image. However, in this study, the smoke region in the image was detected using these haze detection characteristics. The characteristic of the smoke region that was detected using the dark channel prior was identified in pixels where haze or smoke does not exist; here, at least one color channel among R, G, and B had a value close to 0, and this pixel was defined as a dark pixel. The smoke region detected through these features was additionally filtered using optical flow, based on the Lucas–Kanade method. This allows the smoke to be effectively detected by filtering the background through dynamic characteristics, in which the smoke moves in the upward direction. Optical flow is an important technique for analyzing the motion of an object in computer vision, and includes a differential method, a matching method, and a phase-based method. Although there are various optical flow techniques, in this study the smoke motion characteristics were detected using the Lucas–Kanade method [11].

Finally, a CNN was used to detect fire with higher accuracy and reliability for the pre-processed flame and candidate smoke regions. Among the CNN models, the Inception-V3 model was used for inference, and images related to flames and smoke were collected and configured as a training dataset.

2. Image Pre-Processing for Fire Detection

2.1. Flame Detection

The first image pre-processing step employed for flame detection in this study was HSV color conversion. The HSV color model can be used to identify the color of objects in various applications other than image pre-processing. Hue and saturation components are very useful because they are similar to how humans perceive color, which can be an ideal method for developing image-processing algorithms. Hue represents the distribution of colors based on red, and saturation represents the degree to which white light is included in color. Additionally, value is used to control the intensity of light. The value can be independent of a single component to control its range, thus creating algorithms that are robust to lighting changes [12,13].

In Equation (1), it should be noted that when each pixel value is 1, it indicates a region corresponding to a color space in which a flame can exist at an image location, and pixels in the corresponding range are extracted as a candidate region. A pixel value of 0 means that a pixel is classified as a non-flame.

$$RoI_{HSV(x,y)} \begin{cases} 1, & \begin{matrix} (20 < H(x,y) < 40) \text{ and} \\ (50 < S(x,y) < 255) \text{ and} \\ (50 < V(x,y) < 255) \text{ and} \end{matrix} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Figure 2 shows such HSV color conversion: Figure 2a is the original flame image, and Figure 2b is the image of the result to which the HSV color conversion is applied.

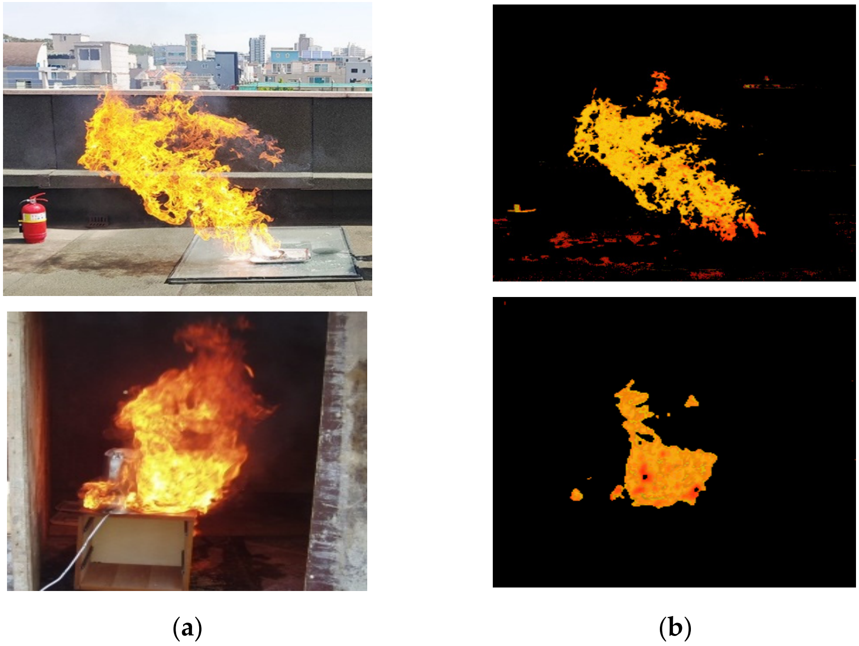


Figure 2. HSV color conversion of flame image. (**a**) original images; (**b**) HSV color conversion in the specified range.

Even after HSV color conversion, the results of objects, including light yellow other than flame, remain. To further filter this, Harris corner detector was used as the second image pre-processing step. Among the remaining objects following HSV color conversion, the flame had a sharp texture, which resulted in a large number of corners. Therefore, in a region where corners are intensively generated, it is highly likely that it is a flame, and such a region is detected as a candidate region.

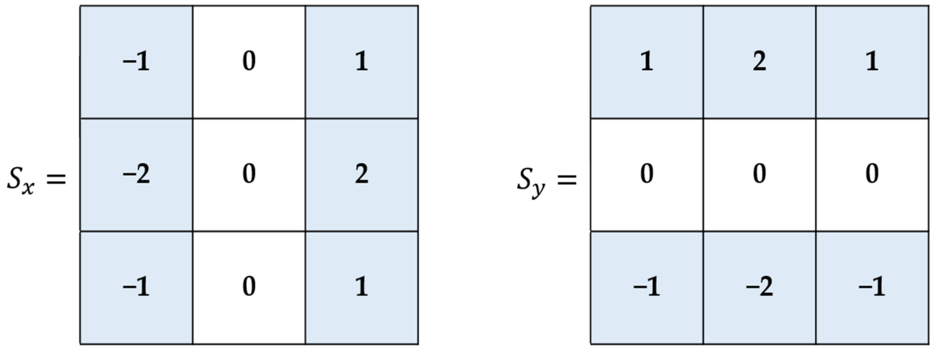
$$E(x,y) = \sum_W [I(x_i,y_i) - I(x_i+u,y_i+v)]^2 \tag{2}$$

First, when there is a reference point (x,y) in the image, it can be expressed as Equation (2). When the amount of change shifts by (u,v) from the reference point. I represents brightness, and (x,y) represents points inside Gaussian window W. The region moved by (u,v) can be organized as shown in Equation (3) below, using the Taylor series.

$$I(x_i+u,y_i+v) \approx I(x_i,y_i) + \left[I_x(x_i,y_i)I_y(x_i,y_i)\right]\begin{bmatrix}u\\v\end{bmatrix} \tag{3}$$

The first-order derivative in the x and y directions, I_x and I_y, could be obtained via convolution arithmetic, using S_x, the Sobel x kernel, and S_y, the Sobel y kernel, as shown in Figure 3. If Equation (3) is substituted for Equation (2), it can be expressed as Equation (4).

$$E(u,v) = [u\ v]\begin{bmatrix}\sum_W (I_x(x_i,y_i))^2 & \sum_W I_x(x_i,y_i)I_y(x_i,y_i)\\ \sum_W I_x(x_i,y_i)I_y(x_i,y_i) & \sum_W \left(I_y(x_i,y_i)\right)^2\end{bmatrix}\begin{bmatrix}u\\v\end{bmatrix} = [u\ v]M\begin{bmatrix}u\\v\end{bmatrix} \tag{4}$$



$$S_x = \begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \qquad S_y = \begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

Figure 3. Mask used by Sobel filter.

If M is defined $M = \begin{bmatrix}A\ C\\ C\ B\end{bmatrix}$, properties such as Equations (5) and (6) are satisfied. Finally, Equation (7) allows us to determine the edge, corner, and flat. k is an empirical constant, and a value of 0.04 was used in this paper.

$$det(M) = AB - C^2 = \lambda_1\lambda_2 \tag{5}$$

$$trace(M) = A + B = \lambda_1 + \lambda_2 \tag{6}$$

$$R(x,y) = det(M) - k(trace(M))^2 \tag{7}$$

Each pixel's location will have a different value, and the final calculated $R(x,y)$ will be compared to the following conditions to distinguish between the edge, corner, and flat [14–17].

- When $|R|$ is small, which happens when λ_1 and λ_2 are small, these points belong to flat regions;
- When $R < 0$, if only one eigenvalue of λ_1 and λ_2 is bigger than the other eigenvalue, the region belongs to edges;
- If R has a large value, the region is a corner.

In these conditions, R has a large value and corresponds to a corner, and Figure 4 is the result of visualizing pixels that satisfy this condition.

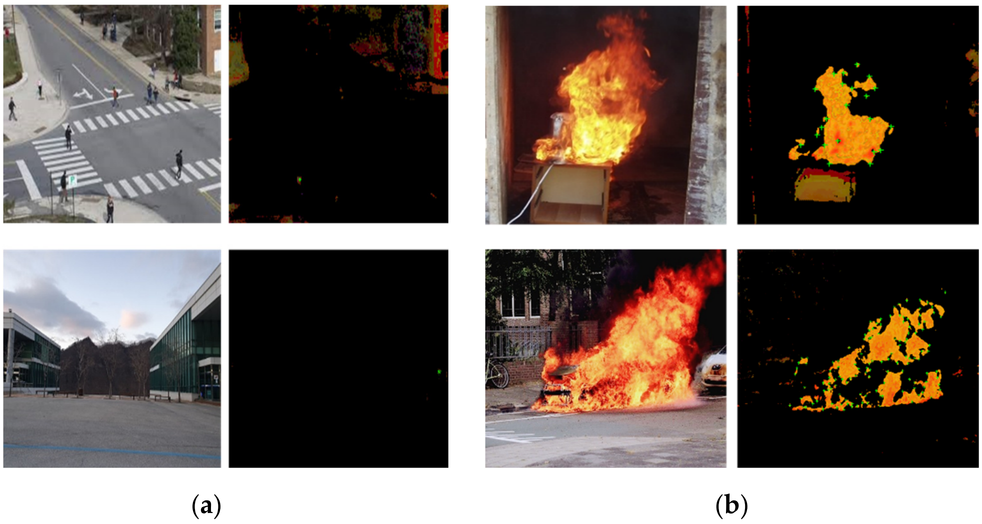


Figure 4. Corner detection images. (**a**) the result of detecting the corner of the flame image; (**b**) the result of detecting the corner of the non-flame image.

Figure 4a is the pre-processing result of the image without a flame, and Figure 4b is the pre-processing result of the image where the flame exists; the pixels that satisfy the corner condition in the HSV color conversion image are marked with green dots. In the case of non-flame images, there are many pixels that have not been filtered even via HSV color conversion, but when corner detection is performed, it can be confirmed that most corners do not exist. In the case of the flame image, one result of intensively detecting a corner in a region where the flame exists may be confirmed. Therefore, through this result, when various objects exist in the image, only the flame region may be effectively pre-processed. In addition, the region where these corners are clustered is used as a candidate region that can be inferred through a deep-learning-based CNN.

2.2. Smoke Detection

If smoke occurs in a fire, it can cause negative physiological effects, such as poisoning or asphyxiation, leading to problems in evacuation or extinguishing activities. In addition, when smoke is generated, visibility is poor, the range of action for evacuation is narrowed, and adverse effects such as the malfunction of fire alarm equipment can be caused. Therefore, detecting smoke early in the event of a fire is important. To this end, in this study the smoke region was detected using the dark channel prior and optical flow.

The dark channel refers to a case in which at least one channel with a low intensity value among R, G, and B color channels exists in the case where an image has no haze. Dark channels are algorithms that remove haze based on these characteristics, as proposed by He et al. When haze or smoke exists in the atmosphere, some of the light reflected from the object is lost in the process of being transmitted to the observer or camera, causing the object to appear blurred. This can be expressed as Equation (8), based on pixel x [10].

$$(x) = J(x)t(x) + A(1 - t(x)) \tag{8}$$

$J(x)$ represents an undistorted pixel, $I(x)$ represents a pixel that has reached the actual camera, and $t(x)$ has a value of 1 when it reaches the camera completely without haze or smoke with medium transmission. A is air light, and it can be assumed that all pixels in the image have the same value. The operation of the dark channel existing in the pixel x in the image can be expressed as Equation (9).

$$J^{dark}(x) = \min_{C \in r,g,b} (\min_{y \in \Omega(x)} (J^C(y))) \tag{9}$$

Here, $\Omega(x)$ is the kernel centered on pixel x, and $C \in r, g, b$ represents the value of each channel. Therefore, a case where the brightness value of at least one channel among the R, G, and B values of $\Omega(x)$ is very low is defined as J^{dark} [10]. Using the characteristics of these dark channels, it is possible not only to effectively remove haze or fog from the image, but also to pre-process smoke regions that may occur during a fire. Figure 5 shows the results of a thresholding set through the dark channel characteristics that exist in the pixel.

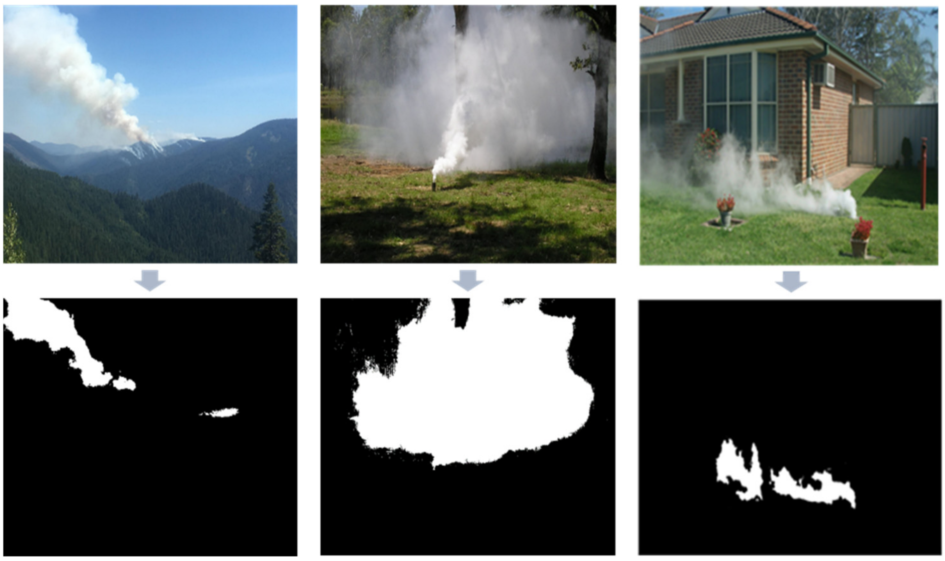


Figure 5. Smoke image thresholded through a dark channel.

The smoke region detected through the dark channel prior is filtered once more through the dynamic characteristics of the smoke. Combustion of solid fuels in a fire usually entails heat in the adjacent material that burns or in the fuel itself. As a result, hot volatile or flammable vapors are emitted, and when a fire column and gas accompanying hot smoke are generated, they rise above the surrounding cold air due to the lowered gas density [18–20]. Therefore, in order to pre-process the image with the smoke flow characteristics, the motion of the smoke rising upward was detected using the optical flow algorithm. Estimating the motion of an object through optical flow uses the change in contrast from two adjacent images with a time difference [21,22].

The optical flow based on the Lucas–Kanade method makes appropriate assumptions within the range that does not significantly deviate from the actual reality. Among them, brightness constancy is the most important assumption with regard to the optical flow estimation algorithm. According to the brightness constancy, the same part of two scenes with a time difference from the video have the same or almost the same contrast values. This brightness constancy is not always correct in reality, but it is based on the principle whereby it can be assumed that the change in contrast of an object is not large in the short time difference between image frames [23,24].

If the time difference between two adjacent images is sufficiently small, the following Equation (10) is established according to the Taylor series.

$$f(y + dy, x + dx, t + dt) = f(y, x, t) + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial t}dt + \ldots \tag{10}$$

Assuming that is small, dy and dx, which represent the movement of an object, are also small, so there is no significant error even if the quadratic term or higher are ignored.

As mentioned earlier, according to the assumption with regard to brightness constancy, the new point $f(y + dy, x + dx, t + dt)$ is formed by moving (dx, dy) during the time dt, so that $f(y+dy, x+dx, t+dt)$ of the new point is the same as $f(y,x,t)$ of the original point, $\frac{dy}{dt} = v$, and $\frac{dx}{dt} = u$. Therefore, Equation (10) can be written as Equation (11).

$$\frac{\partial f}{\partial y}v + \frac{\partial f}{\partial x}u + \frac{\partial f}{\partial t} = 0 \tag{11}$$

This equation is a differential equation and is called an optical flow constraint equation or a gradient constraint equation. Although the motion of the object can be estimated through this equation, the resulting value cannot be determined because there are two unknowns, v and u. In order to solve two vectors that could not be obtained in the optical flow estimation algorithm, the Lucas–Kanade algorithm, a local computation method, is used. The Lucas–Kanade method solves the equation using the least squares method as shown in Equation (12) below [25].

$$\begin{pmatrix} v \\ u \end{pmatrix} = \begin{bmatrix} \sum_{i=1}^{n} w_i \left(\frac{\partial(y_i,x_i)}{\partial y}\right)^2 & \sum_{i=1}^{n} w_i \frac{\partial f(y_i,x_i)}{\partial y}\frac{\partial f(y_i,x_i)}{\partial x} \\ \sum_{i=1}^{n} \frac{\partial f(y_i,x_i)}{\partial y}\frac{\partial f(y_i,x_i)}{\partial x} & \sum_{i=1}^{n} w_i \left(\frac{\partial(y_i,x_i)}{\partial y}\right)^2 \end{bmatrix}^{-1} \times \left(\frac{-\sum_{i=1}^{n} w_i \frac{\partial f(y_i,x_i)}{\partial y}\frac{\partial f(y_i,x_i)}{\partial t}}{-\sum_{i=1}^{n} w_i \frac{\partial(y_i,x_i)}{\partial x}\frac{\partial(y_i,x_i)}{\partial t}} \right) \tag{12}$$

i corresponds to the coordinate values of all pixels, and the optical flow is calculated based on the derivative value calculated in each pixel. The change in the direction of optical flow can distinguish the smoke area by manually setting the threshold T.

Figure 6 depicts a scene where the smoke flow is detected using optical flow from the smoke generated images. Using the optical flow algorithm, the vector change of the object is calculated only for the area where the smoke extracted through the dark channel feature is expected to exist, not the entire input image.

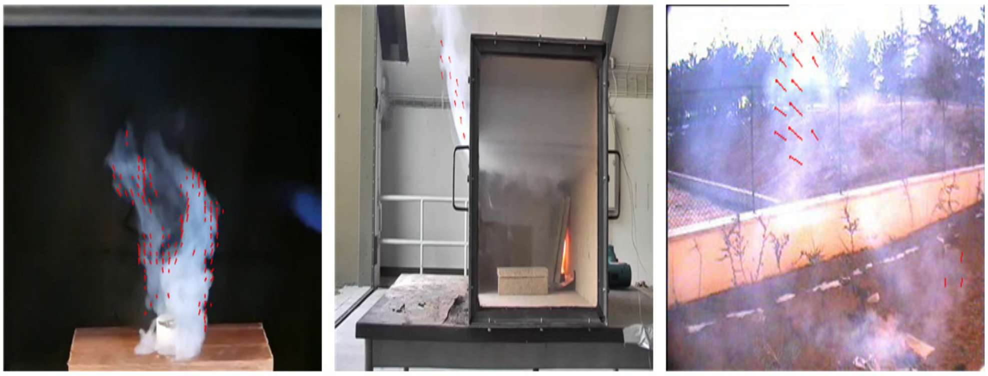


Figure 6. Optical flow detection result of smoke videos.

The information obtained through this includes θ_i, which is the angle, as shown in Equation (13).

$$\theta_i(X,Y) = arctan\left(\frac{Y_{t+1} - Y_t}{X_{t+1} - X_t}\right) \tag{13}$$

$$Candidate\ region = \begin{cases} 1, & 45^\circ < \theta_i(X,Y) < 135^\circ \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

Considering that $\theta_i(X,Y)$ is the direction of the optical flow vector of the pixel at position (X,Y) of the i-th frame according to Equation (13), dx and dy are the motion flow vectors of the row and column, respectively. Among the values of the vectors obtained here, the area that moved from 45 degrees to 135 degrees, as shown in Equation (14), can be filtered. Through these pre-processing processes, a region with a high probability of acting can be used as a candidate region and predictions can be made through a deep-learning-based CNN.

2.3. Inference Using Inception-V3

In order to detect fire with higher accuracy in the region of interest obtained during image pre-processing, in this study, CNN was constructed as the last step toward finally detecting whether a fire occurred. CNN is used in similar computer vision studies, such as in image classification, object detection and recognition, and image matching. In addition, from the simple neural networks of the past, complex and deep network-type models are now being developed.

When training through deep learning, it is common to obtain high precision when using it with a deep layer and a wide node. However, in this case, the amount of parameters increases, and the computational amount increases considerably, and an over-fitting problem or a gradient vanishing problem occurs. Therefore, we made the connections between nodes sparse and the matrix operations dense. Reflecting this, the inception structure in Figure 7 makes the overall network deep but not difficult to operate.

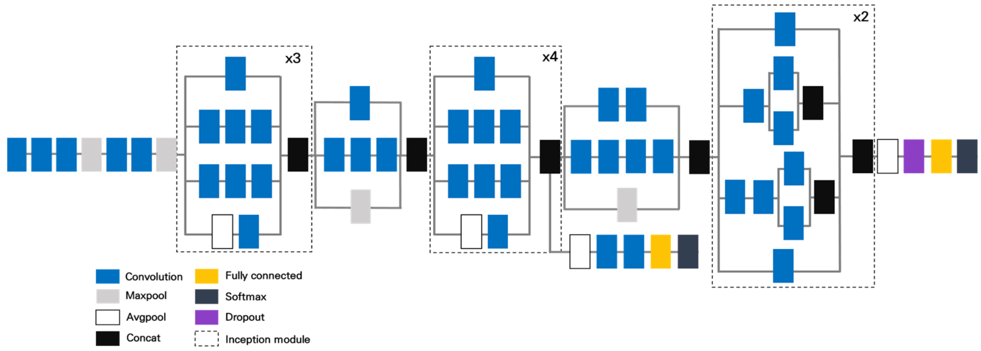


Figure 7. Schematic diagram of Inception-V3.

Therefore, the Inception-v3 model has the advantage of having a deeper layer than other CNN models, but not having a relatively large parameter. Table 1 shows the configuration of the CNN layer configured using Inception modules. The size of the input image was set to 299 × 299, and a reduction layer was added between the inception modules [26–28]. With most CNN, the pooling layer is used, but it is constructed to solve the representational bottleneck problem. Finally, softmax was used as the activation function of the final layer is a classification problem for flame, smoke, and non-fire.

Table 1. Inception-V3 CNN parameter.

Layer	Kernel Size	Input Size
Convolution	3 × 3	299 × 299 × 3
Convolution	3 × 3	149 × 149 × 32
Convolution	3 × 3	147 × 147 × 32
Max pooling	3 × 3	147 × 147 × 64
Convolution	3 × 3	73 × 73 × 64
Convolution	3 × 3	73 × 73 × 80
Max pooling	3 × 3	71 × 71 × 192
Inception module	-	35 × 35 × 192
Reduction	-	35 × 35 × 228
Inception module	-	17 × 17 × 768
Reduction	-	17 × 17 × 768
Inception module	-	8 × 8 × 1280
Average pooling	-	8 × 8 × 2048
Fully connected	-	1 × 2048
Softmax	-	3

In addition, the dataset used for training this CNN model is shown in Table 2. Here, the train dataset used for training and the test dataset for evaluating and reflecting the learning understanding of the training intermediate model were divided into about 8 to 2 ratios, and the training was conducted, and accuracy and loss did not change significantly. Learning was ended at the converging 5000 steps. The train and test image dataset used was obtained from Kaggle and CVonline as public materials for use in research.

Table 2. Composition of datasets for fire detection.

	Classes of Image Datasets		
	Flame	**Smoke**	**Non-Fire**
Train set	6200	6200	6200
Test set	1800	1800	1800

3. Experimental Results

Images of flame and smoke that may occur in fires were pre-processed using appearance characteristics and classified through an Inception-V3 model based on a CNN. Figure 8 visualizes the final detection of the flame region from the test images.

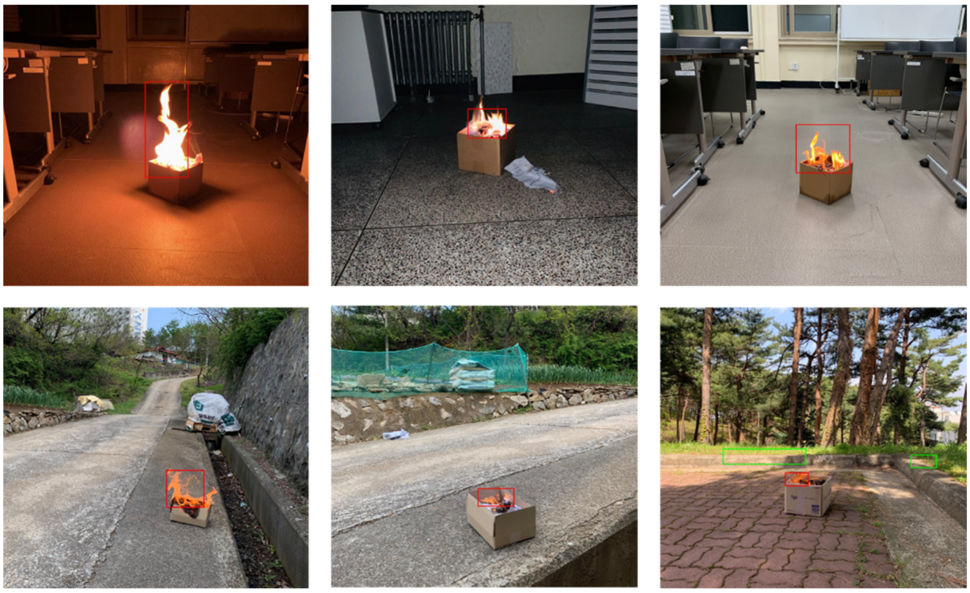


Figure 8. Flame detection results of input videos.

If the candidate region detected through pre-processing is judged to be a flame, it is visualized as a red bounding box, and if it is an object not related to fire, it is visualized as a green bounding box. In addition, Figure 9 visualizes the detection of smoke from the input images, and similarly, in the case of the red bounding box, it is inferred to be the smoke region; in the case of the green bounding box, it is visualized as an object unrelated to fire.

Accuracy, precision, recall, and F1 score were calculated to determine the objective performance of the experimental results obtained through this study, where *TP* is the number of true positives, *FP* the number of false positives, *FN* the number of false negatives and *TN* the number of true negatives. The relationships among them are listed as shown below. First, accuracy and precision were obtained via Equations (15) and (16), and recall was obtained using Equation (17), while F1 score, which is the harmonic mean of the precision and detection rate, was obtained using Equation (18).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{18}$$

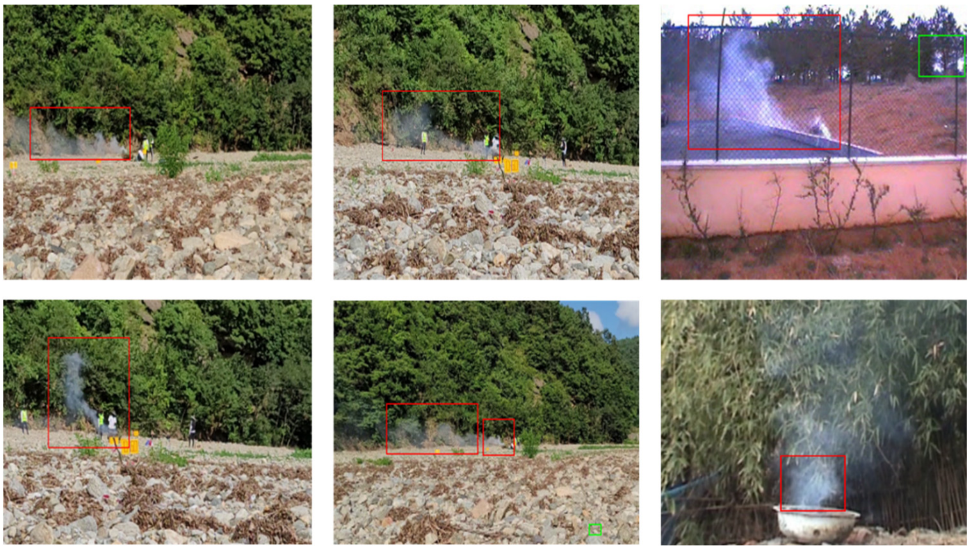


Figure 9. Smoke detection results of input videos.

The performance evaluation was conducted using five videos featuring flames, five videos with smoke, and five videos not related to fire. When carrying out the detection method using optical flow, the performance should be evaluated through continuous images—that is, by using videos, rather than a single still image. Therefore, 50 frames in which inference was performed on the object were extracted from each test video, and the result was calculated. In addition, a performance evaluation was performed in the same way in the flame and non-fire videos. Moreover, to compare the results with the model presented in this study, the deep-learning-based object detection model Single Shot Multibox Detector (SSD) [29]–Faster R-CNN (Region proposal Convolutional Neural Network) [30] was used.

The flame detection results are shown in Table 3, and for the model presented in this study, the accuracy was 96.0%, precision was 94.2%, and recall and F1 score were 98.0% and 96.1, respectively. In the case of SSD, the accuracy was 89.0% and that of Faster R-CNN was 92.0%. The accuracy of the flame detection algorithm presented in this study was relatively high. The frequency of false positives, in which non-flame objects are mis-detected as flames, was decreased by more than 10% compared to other studies, which greatly affected the overall increase in precision.

Table 3. Evaluation of flame detection results from each model.

	Evaluation Indicator			
	Accuracy	**Precision**	**Recall**	**F1 Score**
Our proposed model	96.0	94.2	98.0	96.1
SSD	89.0	86.8	92.0	89.3
Faster R-CNN	92.0	88.9	96.0	92.3

The smoke detection results were similar, and in the model presented in this study, the accuracy was 93.0%, precision was 93.9%, and detection rate and F1 score were 92.0% and 92.9%, respectively. In the case of SSD, the accuracy was 85.0% and Faster R-CNN was 89.0%, as shown in Table 4.

Table 4. Evaluation of smoke detection results from each model.

	Evaluation Indicator			
	Accuracy	**Precision**	**Recall**	**F1 Score**
Our proposed model	93.0	93.9	92.0	92.9
SSD	85.0	84.3	86.0	85.1
Faster R-CNN	89.0	89.8	88.0	88.9

4. Conclusions

In this study, an appropriate pre-processing method was presented to detect both flames and smoke that may occur during a fire in its early stages. To this end, color-based and optical flow methods were used, and in order to make a precise judgment on the detected candidate region, inferences were made using a deep-learning-based CNN. Through this approach, it was possible to reduce false detections due to unnecessary background regions while improving accuracy when detecting fire. Our tests of the proposed flame detection method found that the accuracy was improved by 5.5% compared to the object detection models without separate pre-processing. For the smoke detection method proposed in this study, dark channel feature and optical flow were utilized, and accuracy was improved by 6% compared to other object detection models. In future studies, a CNN that can accurately detect objects with irregular shapes, such as flames and smoke, will be developed or improved for future applications. In addition, research will pursue the development of an intelligent fire detector that can be applied to low-specification systems, and which can easily perform real-time detection by supplementing pre-processing methods. In addition, we will study a method that can accurately detect fire even from small characteristics in images and develop a fire detection model with higher reliability than can be achieved by human observation.

Author Contributions: Conceptualization, J.R. and D.K.; Methodology, J.R.; Software, J.R.; Supervision, D.K.; Validation, D.K.; Writing—original draft preparation, J.R.; Writing—review and editing, D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant (20010162) of Regional Customized Disaster-Safety R&D Program funded by Ministry of Interior and Safety (MOIS, Korea).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Dubinin, D.; Cherkashyn, O.; Maksymov, A.; Beliuchenko, D.; Hovalenkov, S.; Shevchenko, S.; Avetisyan, V. Investigation of the effect of carbon monoxide on people in case of fire in a building. *Sigurnost* **2020**, *62*, 347–357. [CrossRef]
2. Hadano, H.; Nagawa, Y.; Doi, T.; Mizuno, M. Study of effectiveness of CO and Smoke Alarm in smoldering fire. *ECS Trans.* **2020**, *98*, 75–79. [CrossRef]
3. Gałaj, J.; Saleta, D. Impact of apartment tightness on the concentrations of toxic gases emitted during a fire. *Sustainability* **2019**, *12*, 223. [CrossRef]
4. Shen, D.; Chen, X.; Nguyen, M.; Yan, W. Flame detection using deep learning. In Proceedings of the 2018 4th International Conference on Control, Automation and Robotics (ICCAR), Auckland, New Zealand, 20–23 April 2018; pp. 416–420.
5. Muhammad, K.; Khan, S.; Elhoseny, M.; Ahmed, S.; Baik, S. Efficient Fire Detection for Uncertain Surveillance Environment. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3113–3122. [CrossRef]
6. 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), Phu Quoc Island, Vietnam, 13–15 January 2021.
7. Jeon, M.; Choi, H.-S.; Lee, J.; Kang, M. Multi-scale prediction for fire detection using Convolutional Neural Network. *Fire Technol.* **2021**, *57*, 2533–2551. [CrossRef]

8. Lai, T.Y.; Kuo, J.Y.; Fanjiang, Y.-Y.; Ma, S.-P.; Liao, Y.H. Robust little flame detection on real-time video surveillance system. In Proceedings of the 2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications, Kaohsiung, Taiwan, 26–28 September 2012.
9. Ryu, J.; Kwak, D. Flame detection using appearance-based pre-processing and Convolutional Neural Network. *Appl. Sci.* **2021**, *11*, 5138. [CrossRef]
10. Kaiming, H.; Sun, J.; Tang, X. Single Image Haze Removal Using Dark Channel prior. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.
11. Kwak, D.-K.; Ryu, J.-K. A study on the dynamic image-based dark channel prior and smoke detection using Deep Learning. *J. Electr. Eng. Technol.* **2021**, *17*, 581–589. [CrossRef]
12. Kang, H.-C.; Han, H.-N.; Bae, H.-C.; Kim, M.-G.; Son, J.-Y.; Kim, Y.-K. HSV color-space-based automated object localization for robot grasping without prior knowledge. *Appl. Sci.* **2021**, *11*, 7593. [CrossRef]
13. Chen, W.; Chen, S.; Guo, H.; Ni, X. Welding flame detection based on color recognition and progressive probabilistic Hough Transform. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5815. [CrossRef]
14. Gao, Q.-J.; Xu, P.; Yang, L. Breakage detection for grid images based on improved Harris Corner. *J. Comput. Appl.* **2013**, *32*, 766–769. [CrossRef]
15. Chen, L.; Lu, W.; Ni, J.; Sun, W.; Huang, J. Region duplication detection based on Harris Corner Points and step sector statistics. *J. Vis. Commun. Image Represent.* **2013**, *24*, 244–254. [CrossRef]
16. Sánchez, J.; Monzón, N.; Salgado, A. An analysis and implementation of the Harris Corner Detector. *Image Process. Line* **2018**, *8*, 305–328. [CrossRef]
17. Semma, A.; Hannad, Y.; Siddiqi, I.; Djeddi, C.; El Youssfi El Kettani, M. Writer identification using deep learning with fast keypoints and Harris Corner Detector. *Expert Syst. Appl.* **2021**, *184*, 115473. [CrossRef]
18. Appana, D.K.; Islam, R.; Khan, S.A.; Kim, J.-M. A video-based smoke detection using smoke flow pattern and spatial-temporal energy analyses for alarm systems. *Inf. Sci.* **2017**, *418–419*, 91–101. [CrossRef]
19. Wang, Y.; Wu, A.; Zhang, J.; Zhao, M.; Li, W.; Dong, N. Fire smoke detection based on texture features and optical flow vector of contour. In Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016.
20. Bilyaz, S.; Buffington, T.; Ezekoye, O.A. The effect of fire location and the reverse stack on fire smoke transport in high-rise buildings. *Fire Saf. J.* **2021**, *126*, 103446. [CrossRef]
21. Plyer, A.; Le Besnerais, G.; Champagnat, F. Massively parallel lucas kanade optical flow for real-time video processing applications. *J. Real-Time Image Process.* **2014**, *11*, 713–730. [CrossRef]
22. Sharmin, N.; Brad, R. Optimal filter estimation for Lucas-Kanade Optical Flow. *Sensors* **2012**, *12*, 12694–12709. [CrossRef]
23. Liu, Y.; Xi, D.-G.; Li, Z.-L.; Hong, Y. A new methodology for pixel-quantitative precipitation nowcasting using a pyramid Lucas Kanade Optical Flow Approach. *J. Hydrol.* **2015**, *529*, 354–364. [CrossRef]
24. Douini, Y.; Riffi, J.; Mahraz, M.A.; Tairi, H. Solving sub-pixel image registration problems using phase correlation and Lucas-Kanade Optical Flow Method. In Proceedings of the 2017 Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 17–19 April 2017.
25. Hambali, R.; Legono, D.; Jayadi, R. The application of pyramid Lucas-Kanade Optical Flow Method for tracking rain motion using high-resolution radar images. *J. Teknol.* **2020**, *83*, 105–115. [CrossRef]
26. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
27. Demir, A.; Yilmaz, F.; Kose, O. Early detection of skin cancer using deep learning architectures: Resnet-101 and inception-V3. In Proceedings of the 2019 Medical Technologies Congress (TIPTEKNO), Izmir, Turkey, 3–5 October 2019.
28. Kristiani, E.; Yang, C.-T.; Huang, C.-Y. ISEC: An optimized deep learning model for image classification on Edge Computing. *IEEE Access* **2020**, *8*, 27267–27276. [CrossRef]
29. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot multibox detector. In *Computer Vision—ECCV 2016*; Springer: Cham, Switzerland, 2016; pp. 21–37.
30. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]