Article

Flame Detection Using Appearance-Based Pre-Processing and Convolutional Neural Network

Jinkyu Ryu and Dongkurl Kwak *

Graduate School of Disaster Prevention, Kangwon National University, Samcheok-si 25913, Gangwon-do, Korea; jkryu@kangwon.ac.kr
* Correspondence: dkkwak@kangwon.ac.kr; Tel.: +82-33-570-6823

Abstract: It is important for fire detectors to operate quickly in the event of a fire, but existing conventional fire detectors sometimes do not work properly or there are problems where non-fire or false reporting occurs frequently. Therefore, in this study, HSV color conversion and Harris Corner Detection were used in the image pre-processing step to reduce the incidence of false detections. In addition, among the detected corners, the vicinity of the corner point facing the upper direction was extracted as a region of interest (ROI), and the fire was determined using a convolutional neural network (CNN). These methods were designed to detect the appearance of flames based on top-pointing properties, which resulted in higher accuracy and higher precision than when input images were still used in conventional object detection algorithms. This also reduced the false detection rate for non-fires, enabling high-precision fire detection.

Keywords: deep learning; convolutional neural network; computer vision; video recognition

1. Introduction

In the case of fire, death is more often caused by the inhalation of toxic substances such as carbon monoxide than by direct injury caused by burns. Therefore, it is important to detect and respond to the occurrence of a fire in the early stages. Additionally, because the precise operation of detectors is directly related to the saving of human life, the study of new fire detection methods with higher performance than conventional sensor-based detection is urgently needed. Existing sensor-based fire detectors include flame detectors that detect infrared (IR) and ultraviolet (UV) energy, and heat detectors that detect heat sources.

However, these sensor-based fire detection methods are limited in indoor environments, and the more sensitive such detectors are to IR, UV, and heat, the more easily they react to other factors, resulting in unnecessary manpower consumption due to the malfunction of alarms. In addition, there are limitations, such as the inability to provide information about the location and size of fires, and frequent false fire alarms if the physical sensor is close to the source of the fire, or in contrast, if there are factors that make the operation of the fire detector too sensitive. Moreover, if actual fires occur in a wide range of areas such as large factories and mountains, early fire detection is difficult with existing sensor-based fire detection systems.

Therefore, to address these existing problems, this study aimed to present a supplemented image preprocessing method to detect fire hazards quickly and automatically, and a flame detection method that reduced the misdetection rate via CNN. Machine learning is a branch of artificial intelligence in which computers train on their own to develop predictive models, and deep learning is a machine learning method using deep neural network theory [1–3].

Such deep learning has shown excellent performance in various fields such as pattern recognition, computer vision, speech recognition, and translation. In particular, the CNN used in this work is an artificial neural network built on the basis of deep learning technology, which has the advantage of being able to train while maintaining the spatial structure...
of the image. This exacerbates the problem of losing features to the original data in the process of training by treating it with one-dimensional flat data when using the existing fully connected layer [4–8].

In the field of artificial intelligence, object recognition or region-based object detection using deep learning is an active field of research, and image recognition algorithms through deep learning are ranked at the top in competitions such as ILSVRC (Image Large Scale Visual Recognition Challenge). In the case of deep learning-based models, detection algorithms based on regions that exist within images as well as the classification of images, such as Single Shot Multibox Detector (SSD) or Region-based Convolutional Neural Network (R-CNN) algorithms, have recently emerged [9–14].

However, even if the latest deep learning-based image recognition or object detection algorithms are used, without separate robust image pre-processing, the achieved results may be less accurate than expected in areas requiring high reliability, such as fire detection. Therefore, good results can be obtained if unnecessary background regions are removed as much as possible through image preprocessing and detection is attempted through a deep learning model [15–18]. For example, Zhong et al. [19] proposed that the predicted area where the flame exists in the input image should be filtered through the RGB model corresponding to the flame color, and the flame was detected through the CNN for the corresponding area. Cai et al. [20] similarly performed color-based pre-processing filtering from the input images, pre-processing of flame regions via HSV color transformation, and YCbCr and Canny edge detection. Additionally, the flame was detected in the form of minimizing overfitting by removing the fully connected layer, which is traditionally used in the last layer of convolutional neural networks, and global interval pooling was applied.

The pre-processing method used in this study first converted the input image to HSV color channels and filtered only the color distribution of the flame. Moreover, when detecting the corner point using the Harris corner detector, pre-processing was performed based on the appearance characteristics of finding the sharp part of the flame. In other words, only the top-direction corner points, not the side or bottom corner points, were detected and used as a ROI; finally, a performance evaluation of flames from real fire images via a CNN was conducted.

2. Proposed Method

2.1. Overview of Proposed Approach

The most basic flame detection method using CNN is based on iterative learning through a learning dataset for flames, and then the image is classified as being a flame or not through a new input image. However, there is a limitation in classification using CNN, and if the object is simply predicted, the accuracy may decrease when multiple objects exist in one image. In addition, repeated learning through many datasets cannot significantly reduce the ratio of false negatives or false positives, resulting in reduced accuracy. Therefore, through the application of an appropriate image pre-processing process, these problems were offset, and the detection accuracy of the flame was improved.

In this study, the proposed method to increase the detection accuracy of a flame was divided into two main procedures. First, the flame and non-flame image datasets were collected and trained using Inception-V3 among the CNN models, as shown on the left in Figure 1. When classifying a flame from direct input images without any pre-processing for models learning about a flame, it is difficult to reliably classify flame or non-flame images because they contain non-flame objects or unnecessary background areas. Therefore, the first image pre-processing separates objects from the input image filters the HSV color regions where flames exist. Subsequently, second image pre-processing uses Harris corner detector to detect corner points. Additionally, corner point detection was performed only on areas where corner points existed in a 45 to 135 degree direction, with sharp characteristics at the top of the detected corner points, bounding boxes for dense points, and the area was extracted as a ROI.
corner detector to detect corner points. Additionally, corner point detection was performed only on areas where corner points existed in a 45 to 135 degree direction, with sharp characteristics at the top of the detected corner points, bounding boxes for dense points, and the area was extracted as a ROI.

Figure 1. The flame detection algorithms proposed in this paper.

2.2. Image Pre-Processing

In this study, HSV color transformation was performed with the first image pre-processing. The HSV color model can handle color features in a similar way to how humans perceive colors, so it can be used to identify colors of objects in many applications, in addition to image pre-processing. These properties of HSV color models make them ideal tools for developing image processing algorithms based on color sensing properties [21,22].

In HSV color models, hue represents the distribution of colors based on the longest wavelength of red, and saturation represents the degree to which pure colors contain white light.

Value is also used to measure the intensity of light. The value can be independent of a single component to control the range, thus creating algorithms that are robust to lighting changes.

\[
ROI_{HSV(x,y)} = \begin{cases} 
1, & (20 < H(x,y) < 40) \text{ and } (50 < S(x,y) < 255) \text{ and } (50 < V(x,y) < 255) \\
0, & \text{otherwise} 
\end{cases} 
\] (1)

In Equation (1), the pixel values 1 for \( H \), \( S \), and \( V \), respectively refer to the regions corresponding to the color space where flames could exist at the image location, and those in that range were extracted as the ROI of the first image pre-processing. A pixel value of 0 means a pixel is classified as a non-flame area.

Figure 2 shows the HSV color conversion: Figure 2a is the original flame image, and Figure 2b is the resultant image from applying the HSV color conversion. However, even after HSV color conversion, results from items other than flames or objects containing light-yellow remained, such as leaves. Therefore, to filter this additionally, a Harris corner detector was used for the second image preprocessing.
Figure 2. Conversion of flame image using the HSV color model. (a) Original image before color conversion; and (b) HSV color-converted image within the set range.

First, when there is a reference point \((x, y)\) in the image, when the amount of change is moved by \((u, v)\) from the reference point, it can be expressed as Equation (2). \(I\) is the brightness, and \((x_i, y_i)\) are the points inside the Gaussian window \(W\).

\[
E(x, y) = \sum_W [I(x_i + u, y_i + v)]^2
\] (2)

The Taylor series allows the area that has moved as much as \((u, v)\) to be organized as in the following Equation (3).

\[
I(x_i + u, y_i + v) \approx I(x_i, y_i) + [I_x(x_i, y_i) I_y(x_i, y_i)] \begin{bmatrix} u \\ v \end{bmatrix}
\] (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.

\[
S_x = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{bmatrix} \quad S_y = \begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1
\end{bmatrix}
\]

Figure 3. Sobel \(x\) kernel and \(y\) kernel.

If Equation (3) is substituted for Equation (2), it can be expressed as Equation (4).

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

where \(K\) is an empirical constant, and a value of 0.04 was used in this paper.
If $M$ is defined as $M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$, 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.

\[
det(M) = AB - C = \lambda_1 \lambda_2, \quad \text{(5)}
\]
\[
\text{trace}(M) = A + B = \lambda_1 + \lambda_2 \quad \text{(6)}
\]
\[
R(x, y) = \det(M) - k(\text{trace}(M))^2 \quad \text{(7)}
\]

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 [23–26].

- When $|R|$ is small, which happens when $\lambda_1$ and $\lambda_2$ are small, these points belong to flat regions;
- When $R < 0$, if only one eigenvalue of $\lambda_1$ and $\lambda_2$ is bigger than the other eigenvalue, the region belongs to edges;
- If $R$ has a large value, the region is a corner.

Figure 4a shows the input original images, and Figure 4b shows the corner detected by the HSV color conversion images using the Harris corner detector, and the corner points are still detected for non-flame objects. In order to only select the areas that were most likely to be flames among the corner points of the detected objects, this study additionally applied the appearance characteristics of flames in the flame. Therefore, this paper proposes a method to further filter only the corners with the top of the detected corners.

\[
\theta(x, y) = \begin{cases} 
(180^\circ / \pi \times \arctan(I_x(x, y)/I_y(x, y))) & \text{if } (I_x \geq 0) \\
180 - (180^\circ / \pi \times \arctan(I_y(x, y)/I_x(x, y))) & \text{if } (I_x < 0)
\end{cases} \quad \text{(8)}
\]

In order to only select the corner facing the top direction among the detected corners, the angle of the corner-facing direction was calculated using Equation (8). $I_x$ and $I_y$ are first-order derivatives in the $x$ and $y$ directions, respectively, and can be obtained via a convolution arithmetic using $S_x$, the Sobel x kernel, and $S_y$, the Sobel y kernel.

\[
\text{ROI} = \begin{cases} 
1, & 45^\circ < \theta(x, y) < 135^\circ \\
0, & \text{otherwise}
\end{cases} \quad \text{(9)}
\]
Therefore, each Sobel filter value could be used to calculate the angle in the direction toward the corner point through the arctangent, of which only the corners from 45 to 135 degrees, as shown in Equation (9), are shown (Figure 4c). As a result of the pre-processing of these images, most of the non-flaming objects have been removed, but there are cases where the corners of some non-flaming objects remain. Therefore, the area where the corners are concentrated is designated as the ROI, and finally, the process of classifying flames or non-flames using the Inception-V3 CNN model was undertaken.

2.3. Inception-V3 CNN Model

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 number of parameters increased and the computational amount increased considerably, and an over-fitting problem or a gradient vanishing problem occurred. Therefore, we made the connections between nodes sparse and the matrix operations dense. Reflecting this, the Inception structure makes the overall network deep, but not difficult to operate [27–29].

The left side of Figure 5 shows the structure of the Inception A, Inception B, and Inception C modules, including a $1 \times 1$ convolution filter. The $1 \times 1$ convolution filter has no change in height or width, and even if convolution is performed on a plane, there is no spatial feature loss.

![Figure 5.](image)

**Figure 5.** The three main modules used in Inception-V3: (a) Inception A module; (b) Inception B module; and (c) Inception C module.

Therefore, the role of this filter is to increase the number of channels due to the performance of convolution in several layers, which functions to control the number of channels. This can reduce the number of parameters on $3 \times 3$ or $5 \times 5$ filters followed by $1 \times 1$ filters. Thus, the Inception-V3 model has the advantage of a deeper layer than other CNN models, but not having a relatively large parameter. Table 1 shows the configuration of CNN layers configured using the Inception modules. The input image size of $299 \times 299$ was set, and the first five general convolutional layers are called stems. The layers are more effective than Inception module. Furthermore, the nine Inception modules had a size of $1 \times 2048$ through a fully connected layer. For conventional convolutional neural networks, pooling was used between modules or layers to reduce the size of the parameters, but to solve the problem of representational bottleneck with increasing feature loss, a dimensional reduction method was used, as shown in Figure 6. If the stride was set to 1 and convolution was performed, the operation cost was equal to $2d^2k^2$. In order to reduce the amount of cost that needed to be calculated, if the stride was set to 2 and convolution was performed
with pooling, the operation cost was equal to $2(d/2)^2k^2$. However, when the stride was set to 2, a representational bottleneck occurred. Therefore, the two forms were properly mixed to compensate for the above shortcomings. Finally, because the activation function in the final layer was a classification problem for both flame and non-flame, sigmoid was used.

Table 1. Inception-V3 CNN parameter.

| Layer                   | Kernel Size | Input Size          |
|-------------------------|-------------|---------------------|
| Convolution             | $3 \times 3$ | $299 \times 299 \times 3$ |
| Convolution             | $3 \times 3$ | $149 \times 149 \times 32$ |
| Convolution (Padded)    | $3 \times 3$ | $147 \times 147 \times 32$ |
| MaxPooling              | $3 \times 3$ | $73 \times 73 \times 64$ |
| Convolution             | $3 \times 3$ | $73 \times 73 \times 80$ |
| MaxPooling              | $3 \times 3$ | $71 \times 71 \times 192$ |
| Inception A $\times 3$ | As in Figure 5a | $35 \times 35 \times 192$ |
| Reduction               | As in Figure 6 | $35 \times 35 \times 228$ |
| Inception B $\times 3$ | As in Figure 5b | $17 \times 17 \times 768$ |
| Reduction               | As in Figure 6 | $17 \times 17 \times 768$ |
| Inception C $\times 3$ | As in Figure 5c | $8 \times 8 \times 1280$ |
| AveragePooling          | $3 \times 3$ | $8 \times 8 \times 2048$ |
| FC                      | -           | $1 \times 2048$     |
| Sigmoid                 | -           | -                   |

Figure 6. Structure of the reduction module used to reduce feature size.

3. Experiment and Performance Analysis

3.1. Dataset Configuration for Training

In order to detect objects through deep learning-based convolutional neural networks, the sufficient collection of image datasets is essential for training the model. The number of datasets used in this study is shown in Table 2, divided into two classes: flame or non-flame. The total number of images used in the flame dataset was 10,153, the number of images in the non-flame dataset was 10,024, and the training dataset and test dataset were used in a ratio of 8:2, respectively.
Table 2. Number of images in the dataset.

|                  | Training Dataset | Test Dataset |
|------------------|------------------|--------------|
|                  | Flame Non-Flame  | Flame Non-Flame |
| Flame            | 8152             | 2001         |
| Non-Flame        | 8024             | 2000         |

3.2. Experimental Setup and Training

In the experimental environment, the CPU used the Intel i7-8700 processor, the GPU used the Geforce RTX 3070, and the OS was tested on Linux Ubuntu 18.04. Figure 7 shows the flame dataset training results of CNN, where Figure 7a is accuracy and Figure 7b is loss. The red curve is the dataset used for training among flame images collected by the training dataset, and shows the accuracy detected through this dataset. The blue curve is the test dataset which, unlike the training dataset, has not been used in the training process, and is a dataset used only for accuracy evaluation purposes. Therefore, if high accuracy and low loss are shown in the training dataset and relatively low accuracy and high loss in the test dataset, overfitting can be seen as occurring, and training has not progressed properly. However, the training results of this work show that both datasets have similarly high accuracy and low loss, which can be judged to be well-trained against flames. Additionally, the training was ended at 3000 steps, where the accuracy did not increase over a certain level and converged to some extent.

![Graph](image1)

(a) Accuracy curve of learning dataset and test dataset; and (b) loss curve of learning dataset and test dataset.

Figure 7. Accuracy and loss curve of learning outcomes using CNN. (a) Accuracy curve of learning dataset and test dataset; and (b) loss curve of learning dataset and test dataset.

Figure 8 shows the equipment used for the actual fire test. Figure 9 shows the results of the flame detection evaluation. For specific performance evaluation of the proposed flame detection algorithm, a fire test picture was taken, and pictures of the actual fire site in different environments were used: indoors, outdoors night, day, etc. This was composed of various evaluation pictures.

The presented photos are some of the results of correct flame detection for the flame image, with the green-bound boxes indicating the areas finally determined to be flames in the CNN model. From the left, Figure 9a is the result of detection through the method presented in this study, Figure 9b is the result of detection using the Faster R-CNN, and Figure 9c is the result of detection using SSD. In the case of the model presented in this study, most of the flames were detected even when the size of the object occupied by the flame was small; the Faster R-CNN also correctly judged most flames, but in some cases, it incorrectly judged an object other than a flame. In the case of the SSD, it took less
computation time than the Faster R-CNN, but there were many cases where the object size was not detected in the picture.

The test images comprised 100 fire and non-fire photographs, to evaluate the detection results with the following expressions for a specific and objective accuracy evaluation [30].

\[
\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{10}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{11}
\]

\[
F1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{13}
\]

where \(TP\) is the correct detection of flames, \(FN\) is not detected from the image of flames, \(TN\) is the correct detection of non-flame objects, and \(FP\) is the number of incorrect detections of non-flame objects.

Equation (10) is the result of dividing the whole case when the flame and the non-flame are correctly classified in a manner corresponding to accuracy. Equation (11) is a calculation of precision, with \(TP\) divided by the addition of \(FP\) to \(TP\). However, it is not appropriate to evaluate performance only with accuracy and precision in object detection artificial intelligence models. Therefore, the F1 score was additionally calculated using the detection rates in Equations (12) and (13).

The calculations showed 97.5% accuracy, 98.5% precision, and 98.5% F1 score. This result is shown by comparing with Faster R-CNN and SSD in Table 3, and the compared object detection models were trained through the same image dataset.

![Figure 8. Equipment used for performance evaluation.](image)

**Table 3.** Performance comparison for object detection algorithms with the same dataset.

| Model Name   | Accuracy | Precision | Recall | F1-Score |
|--------------|----------|-----------|--------|----------|
| Our Proposed | 97.5%    | 98.9%     | 96.0%  | 97.4%    |
| Faster R-CNN | 89.0%    | 89.7%     | 88.0%  | 88.8%    |
| SSD          | 79.5%    | 74.7%     | 89.0%  | 81.2%    |
Figure 9. Flame detection results of the model presented in this study and two-object detection: (a) Flame detection results of the model presented in this study; (b) flame detection results using Faster R-CNN; and (c) flame detection results using SSD.
Figures 10–12 show the receiver operating characteristic (ROC) curves and precision recall (PR) curves for specific accuracy comparisons of the three detection models. The ROC curves were expressed through two parameters, TPR (true positive rate) and FPR (false positive rate), and showed that the ROC curve changed by changing the threshold criteria. At this time, if the classification threshold was lowered, the TPR and FPR classified as positive in the general classification model increased together. Therefore, a curve with a higher TPR and a lower FPR on the graph can be judged as a better classification model. Likewise, PR curves are a method of evaluating the performance of classification models by changing thresholds, and the higher the precision and recall values, the better the model [31,32].

![Figure 10](image1.png)

**Figure 10.** ROC and PR curves of the model proposed in this study. (a) ROC curve; and (b) PR curve.

![Figure 11](image2.png)

**Figure 11.** ROC and PR curves of the Faster R-CNN: (a) ROC curve; and (b) PR curve.

![Figure 12](image3.png)

**Figure 12.** ROC and PR curves of the SSD: (a) ROC curve; and (b) PR curve.
Both the Faster R-CNN and SSD models, relatively recent deep learning-based object detection algorithms, showed little difference in accuracy, but there was a problem of SSD being superior in response to small objects in the image: detection took about 1.2 s per frame for Faster R-CNN and 0.32 s for SSD. Furthermore, with regard to the flame detection model presented in this study, it took an average of 0.38 s per frame to reach the final CNN-based inference due to low latency in exploring the ROI, and all objects assumed to be flames were extracted as an ROI. In addition, if the proposed pre-processing was applied to Faster R-CNN and SSD, the false detection rate could be decreased. However, due to the large time delay of object detection algorithms using Faster R-CNN and SSD, it is difficult to expect a rapid response when adding the pre-processing.

Thus, for the models presented in this work, the responsiveness was not much different from other models, but also significantly effective in accuracy and detection rates, which is because non-flame objects are filtered in advance through feature points that are likely to be flames. Therefore, both the precision and detection rates were improved due to the high proportion of TP, while the proportion of FP was lower than that of other models presented for comparison.

4. Conclusions

In this paper, in order to accurately detect flames, image preprocessing was performed through the appearance characteristics of flames and it was finally determined whether they were flames through CNN. Through this image pre-processing, object detection and flame classification in a region where flames were expected to exist in the image were accurately performed. In particular, in the case of flame detection in a fire, precision is the highest priority; therefore, frequent erroneous detection should not occur, and there should not be cases where the actual occurrence of fire cannot be properly detected. In order to reflect these characteristics, it was possible to improve accuracy and precision by significantly reducing false positives and false negatives by filtering objects other than flames in advance. In addition, the proposed model showed higher accuracy than the Faster R-CNN model or SSD that performed object detection through a general input image. This will be a more accurate fire detection method than human judgment if the CNN model is developed and applied. In future studies, appropriate video pre-treatment methods to reduce the mis-detection rate of smoke, which is more difficult to detect than flames,
should be applied, and improvements of the computational time generated during the pre-treatment process of images should be pursued.

**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, Seoul, Korea).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

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

**References**

1. Deng, L.; Hinton, G.; Kingsbury, B. New types of deep neural network learning for speech recognition and related applications: An overview. In Proceedings of the 2013 IEEE International Conference on Acoustics Speech and Signal Processing 2013, Vancouver, BC, Canada, 26–31 May 2013; pp. 8599–8603.

2. Patel, H.; Thakkar, A.; Pandya, M.; Makwana, K. Neural network with deep learning architectures. *J. Inf. Optim. Sci.* 2017, 39, 31–38. [CrossRef]

3. Xu, C.; Chai, D.; He, J.; Zhang, X.; Duan, S. InnoHAR: A Deep Neural Network for Complex Human Activity Recognition. *IEEE Access* 2019, 7, 9893–9902. [CrossRef]

4. Lundervold, A.S.; Lundervold, A. An overview of deep learning in medical imaging focusing on MRI. *Z. Med. Phys.* 2019, 29, 102–127. [CrossRef] [PubMed]

5. Atha, D.J.; Jahanshahi, M.R. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Struct. Health Monit.* 2017, 17, 1110–1128. [CrossRef]

6. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Speed Prediction. *Sensors* 2017, 17, 818. [CrossRef] [PubMed]

7. Ghorbanzadeh, O.; Blaschke, T.; Gholamnia, K.; Meena, S.; Tiede, D.; Aryan, J. Evaluation of Different Machine Learning Methods and Deep-Learning Convolutional Neural Networks for Landslide Detection. *Remote Sens.* 2019, 11, 196. [CrossRef]

8. Yasaka, K.; Akai, H.; Abe, O.; Kiryu, S. Deep Learning with Convolutional Neural Network for Differentiation of Liver Masses at Dynamic Contrast-enhanced CT: A Preliminary Study. *Radiology* 2018, 286, 887–896. [CrossRef] [PubMed]

9. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition 2014, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.

10. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.

11. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV) 2017, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

12. Pitaloka, D.A.; Wulandari, A.; Basaruddin, T.; Liliana, D.Y. Enhancing CNN with Preprocessing Stage in Automatic Emotion Recognition. *Procedia Comput. Sci.* 2017, 116, 523–529. [CrossRef] [PubMed]

13. Liu, Y.; Qin, W.; Liu, K.; Zhang, F.; Xiao, Z. A Dual Convolution Network Using Dark Channel Prior for Image Smoke Classification. *IEEE Access* 2017, 5, 60697–60706. [CrossRef]

14. Seebamrungsat, J.; Praising, S.; Riyamongkol, P. Fire detection in the buildings using image processing. In Proceedings of the 2014 Third ICT International Student Project Conference (ICT-ISPC) 2014, Bangkok, Thailand, 26–27 March 2014.

15. Lei, Y.; Cheng, X.; Min, M.; Xie, Y. A semi-supervised Laplacian extreme learning machine and feature fusion with CNN for industrial superheat identification. *Neurocomputing* 2020, 381, 186–195. [CrossRef]

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

17. Cai, Y.; Guo, Y.; Li, Y.; Li, H.; Liu, J. Fire Detection Method Based on Improved Deep Convolution Neural Network. In Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition 2019, Beijing, China, 23–25 October 2019.

18. Yang, Z.; Shi, W.; Huang, Z.; Yin, Z.; Yang, F.; Wang, M. Combining Gaussian Mixture Model and HSV Model with Deep Convolution Neural Network for Detecting Smoke in Videos. In Proceedings of the 2018 IEEE 18th International Conference on Communication Technology (ICCT) 2018, Chongqing, China, 8–11 October 2018.

19. Pranati, R.; Dipanwita, B.; Kriti, B. A Comparative Assessment of the Performances of Different Edge Detection Operator using Harris Corner Detection Method. *Int. J. Comput. Appl.* 2012, 59, 7–13.
20. Hassan, N.; Ming, K.W.; Wah, C.K. A Comparative Study on HSV-based and Deep Learning-based Object Detection Algorithms for Pedestrian Traffic Light Signal Recognition. In Proceedings of the 2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS) 2020, Singapore, 26–29 February 2020.

21. Harris, C.; Stephens, M. A Combined Corner and Edge Detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 147–151.

22. Ye, Z.; Pei, Y.; Shi, J. An Improved Algorithm for Harris Corner Detection. In Proceedings of the 2009 2nd International Congress on Image and Signal Processing 2009, Tianjin, China, 17–19 October 2009.

23. Zhang, J.; Lian, Y.; Jiao, C.; Guo, D.; Liu, J. An Improved Harris Corner Distraction Method Based on B_Spline. In Proceedings of the 2010 2nd IEEE International Conference on Information Management and Engineering, Chengdu, China, 16–18 April 2010; pp. 504–506.

24. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015, Boston, MA, USA, 7–12 June 2015.

25. 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) 2016, Las Vegas, NV, USA, 27–30 June 2016.

26. Guan, Q.; Wan, X.; Lu, H.; Ping, B.; Li, D.; Wang, L.; Zhu, Y.; Wang, Y.; Xiang, J. Deep convolutional neural network Inception-v3 model for differential diagnosing of lymph node in cytological images: A pilot study. Ann. Transl. Med. 2019, 7, 307. [CrossRef] [PubMed]

27. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) 2015, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.

28. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

29. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.

30. Jiao, L.; Bie, R.; Wu, H.; Wei, Y.; Ma, J.; Umek, A.; Kos, A. Golf swing classification with multiple deep convolutional neural networks. Int. J. Distrib. Sens. Netw. 2018, 14, 59–65. [CrossRef]

31. Chouhan, N.; Khan, A.; Khan, H.-R. Network anomaly detection using channel boosted and residual learning based deep convolutional neural network. Appl. Soft Comput. 2019, 83, 1–32. [CrossRef]

32. Son, J.; Park, S.J.; Jung, K.-H. Towards Accurate Segmentation of Retinal Vessels and the Optic Disc in Fundoscopic Images with Generative Adversarial Networks. J. Digit. Imaging 2018, 32, 499–512. [CrossRef] [PubMed]