International Journal of Advanced Smart Convergence Vol.8 No.2 116-125 (2019)
http://dx.doi.org/10.7236/IJASC.2019.8.2.116

IJASC 19-2-14

Deep Learning and Color Histogram based Fire and Smoke Detection Research

Yeunghak Lee, Jaechang Shim*

Department of Computer Engineering, Andong National University, Korea
*yhsjh.yi@gmail.com, *jcshim@andong.ac.kr*

Abstract

The fire should extinguish as soon as possible because it causes economic loss and loses precious life. In this study, we propose a new atypical fire and smoke detection algorithm using deep learning and color histogram of fire and smoke. First, input frame images obtain from the ONVIF surveillance camera mounted in factory search motion candidate frame by motion detection algorithm and mean square error (MSE). Second deep learning (Faster R-CNN) is used to extract the fire and smoke candidate area of motion frame. Third, we apply a novel algorithm to detect the fire and smoke using color histogram algorithm with local area motion, similarity, and MSE. In this study, we developed a novel fire and smoke detection algorithm applied the local motion and color histogram method. Experimental results show that the surveillance camera with the proposed algorithm showed good fire and smoke detection results with very few false positives.

Keywords: *Fire Detection, Deep Learning, Structure Similarity, Frame Difference. Color Histogram.*

1. Introduction

Fire is one of the greatest disaster in the world, whether artificially or naturally. Because it causes many people to die and lots of property to be lost, many researchers have been studied the automatic fire detection or monitoring system over the past several years to protect casualties and property damage caused by the fire. There are two kinds of fire and smoke detection systems based on sensors and image processing. Existing fire detection system used in buildings, factories and interior spaces uses a number of optical and ionic sensors (temperature, smoke, sparks, etc.) in a limited area. In case of sensors, it will be limited the performance of detection ability according to the around environment. For example, the smoke detection sensors are difficult to detect, if the air diffusion occurred around the sensor, due to ventilation at the outbreak of fire. And ultraviolet detectors for use with flame detection devices are dropped the sensitivity by the smoke and other factors absorbing the ultraviolet radiation. Sensors using the chemical response by ionization detect the existence of particle which is generated by smoke and fire.

As IT technology evolves, computers have advanced performance, and video analytics has improved

Manuscript Received: Apr. 27, 2019 / Revised: May. 4, 2019 / Accepted: May. 9, 2019
Corresponding Author: jcshim@andong.ac.kr
Tel: +82-54-820-6164, Fax: +82-54-820-5645
Department of Computer Engineering, Andong National University, Korea

significantly over the past decade. Computer vision-based smoke detection systems provide several advantages than conventional methods [1]. Because smoke is preliminary symptom of fire, earlier smoke detection provides to prevent spread of fire flame. Many researchers studied for the smoke and fire flame detection algorithms in the literature based on video image. The majority of these algorithms focuses on the colour and shape characteristics together combined to the temporal behavior of smoke and fire flames. Gubbi [2] used discrete cosine transform and wavelet transform to extract features and support vector machine to detect smoke. Ko [3] extracted several features: colour, wavelet coefficient, motion vector, and 9 feature vectors and used random forest to classify blocked smoke. Frizzi [4] used neural network method using deep convolutional neural network by Tensorflow to detect smoke or smoke and fire. Chen [5] and ÇeliK [6] established a color model to recognize the flames and smoke of fire. This method is to build a variety of feature extraction algorithm or multi-dimensional characteristic vector used as input to the existing algorithm for the classification of neural networks, Adaboost, SVM, etc. [7]. Another approach for fire and smoke detection use deep learning algorithm without special algorithms for feature extraction. Deep learning algorithm train the shape or expression of object data to detect fire and smoke in an image. High recognition Convolutional Neural Networks is a variation of the deep learning training, which can extract phase characteristics. However, it cannot be provided the information about the initial fire position, smoke propagation direction, fire size, fire incidence, etc. LeCun [8] suggested new CNN that was reduced the attenuation of the error, Krizhevsky [9] was applied to CNN to object recognition. CNN uses the ReLU (Rectified Linear Unit), activation function, and dropout to obtain a normalized value [10]. These advanced methods have dramatically improved the existing object recognition methods. General deep learning methods are hard to detect for moving objects and atypical objects which is affected by the environment, especially reflecting sunlight, similar shapes, etc. In this paper, we suggest a novel algorithm which is rigid change of natural environment to reduce the false positive smoke detection based on advanced deep learning, as shown in Figure 1

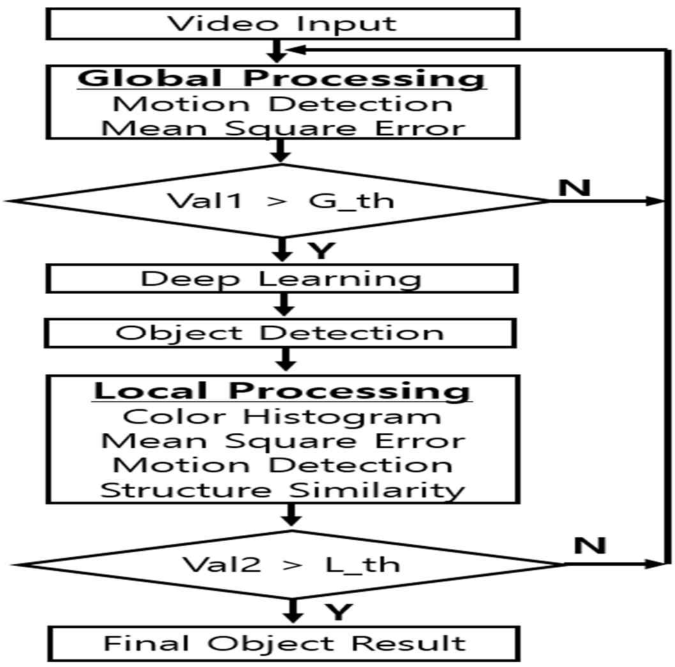

Figure 1. Flowchart of proposed algorithm.

2. Deep learning (Faster R-CNN)

It is often difficult to distinguish objects within an image than to classify images. Basically, deep learning using the R-CNN method goes through several stages. First, the R-CNN creates a region proposal or a bounding box for an area where an object exists. The second is to unify the size of the extracted bounding box to use as input to CNN. The third is to classify the selected region using the SVM. Finally, it uses a linear regression model so that the bounding box of the categorized object sets the exact coordinates. CNN for training data is divided into 3 parts. Figure 2 depicts the full flow of the proposed system.

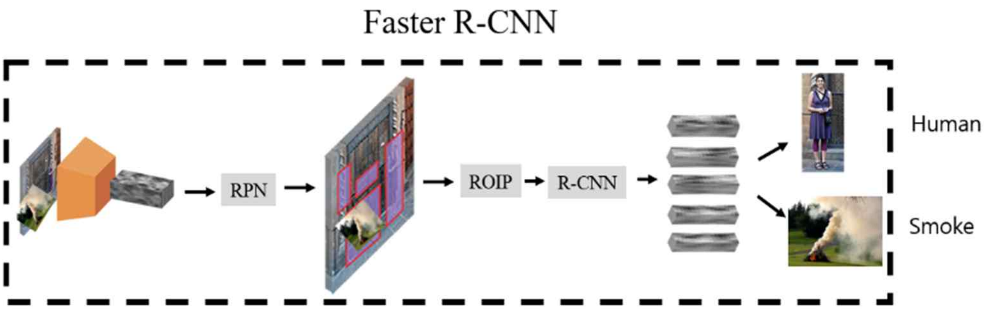

Figure 2. Faster R-CNN system flow.

2.1 Labeling dataset

The labeling of the vehicle in the image was carried out by using the LabelImg program. This paper has used a variety of fire and smoke such as the small, medium and large size for training process, as shown in Figure 3. The labeling result is stored in the .xml file with the four-point coordinates of each rectangle along with the image name. For labeling dataset, there are two things to be considered. First, a list of class is necessary for the dataset. Second, bounding boxes (Xmin, Ymin, Xmax, Ymax) will be generated by the labeling program according to the classes for images

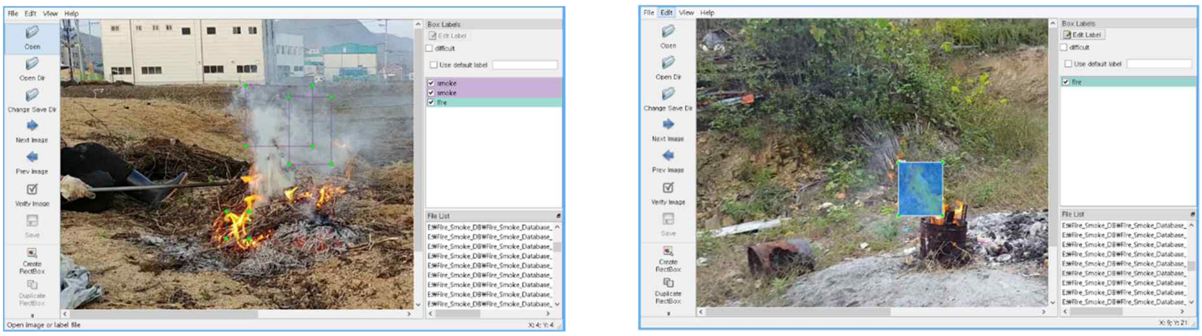

Figure 3. Example of labeling area for fire and smoke dataset image.

2.2 Training dataset with Faster R-CNN

Faster R-CNN is a method of applying a new method called Region Proposal Network (RPN) that merely integrates the part that generates the region proposal within the model. This method is a new application of the RPN network for object detection. The role of RPN is to output the rectangle and object score of the part that proposes the object in the input image. It is a fully connected network and is designed to share a convolutional layer with Faster R-CNN. Trained RPN improves the quality of the proposed area and improves the accuracy of object detection. In general, the Faster R-CNN uses an external slow selection search (calculated by the CPU), but uses an internal fast RPN (calculated by the GPU) to speed up. The RPN is located after the last convolutional layer. Thereafter, ROI pooling, classification, and bounding boxes are located, as shown in Figure 4. RPN extracts 256 or 512 features from input image by convolution calculation using 3x3 window. It is used as a box classifier layer and a box regress layer. The box regression uses predefined reference box names, which are used as bounding box candidates at each position of the sliding window. This extracts features by applying predefined anchor boxes of various ratios/sizes using the center position moving the sliding window of the same size. Nine anchor boxes (three sizes and three proportions) are used. The anchor box is used as a candidate for the bounding box at each position of the sliding window.

2.3 Creating inference graph

Inference graph is also known as a freezing model which is saved for further process. While training dataset with the model, each pair at different time steps, one is holding the weights ".data", and another is holding the graph ".meta". The labeled image data is processed using the Faster R-CNN model described above, and the ".meta" file is generated as a learning result. The final graph file (".pb" file) is generated using the ".meta" file obtained in the previous step. Finally, the object detection results (bounding box and object points) using the ".pb" file are displayed on the monitor of the input image.

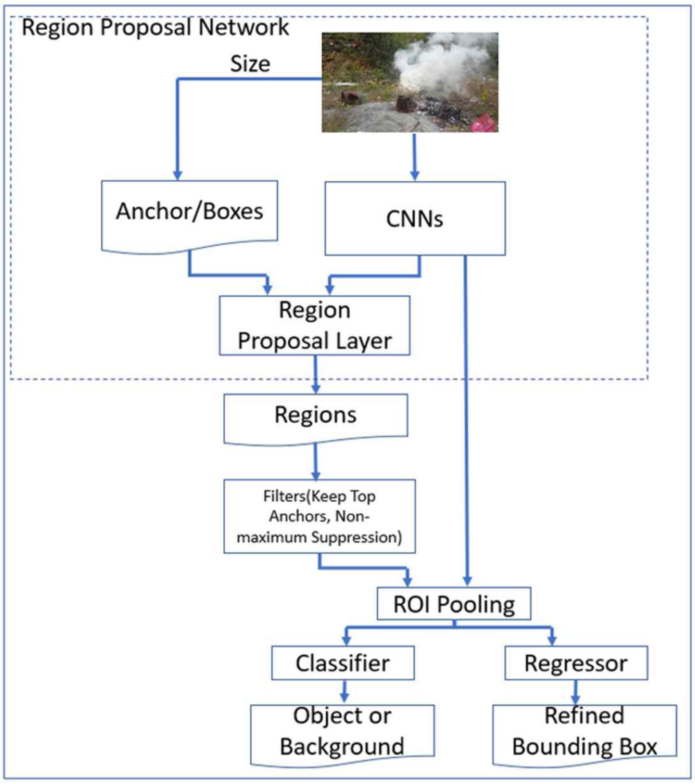

Figure 4. The architecture of Faster R-CNN.

3. Structure of Similarity (SSIM)

Structural Similarity (SSIM) [11] is a method to measure the similarity between the original image and the distortion due to compression and transformation. This shows higher accuracy than the Mean Square Error (MSE) and Peak Signal Noise Ratio (PSNR) methods. This is an evaluation of the test image (X) for the original image (Y) to quantify the visual similarity. A value close to 1.0 means that the test image is similar to the original image, and a value close to 0.0 means that the test image is very different from the original image. The SSIM formulas are defined as follows,

$$l(x,y) = \frac{2\mu_x\mu_y + K1}{\mu_x^2 + \mu_y^2 + K1} \tag{1}$$

$$m(x,y) = \frac{2\sigma_x\sigma_y + K2}{\sigma_x^2 + \sigma_y^2 + K2} \tag{2}$$

$$n(x,y) = \frac{\sigma_{xy} + K3}{\sigma_x\sigma_y + K3} \tag{3}$$

where μ_x and μ_y are mean of pixels, σ_x and σ_y are standard deviation, and σ_{xy} is covariance. *K1*, *K2*, and *K3* are constants for preventing the denominator and numerator from becoming zero. *l*(x, y) is the relationship of the brightness difference, *m*(x, y) is the contrast difference, and *n*(x, y) is the similarity of the structural change between x and y. The structural similarity is shown in Equation (4).

$$SSIM = [l(x,y)]^\alpha [m(x,y)]^\beta [n(x,y)]^\gamma \tag{4}$$

where *α*, *β*, and *γ* represent the importance of each term, 1.0 was used in this paper.

4. Color histogram

The color properties of fire flame and smoke signs are different. Although combustible material affect flame colors, most flame color are reddish. We use HSV color space model instead of RGB to detect fire flame. And smoke can change its brightness in a wide range of values, transparent gray to dark, so we use the analysis of area of contrast.

A color histogram is a representation of the distribution of colors in an image. For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges, that span the image's color space, the set of all possible colors [12]. Please do not write or print outside of the column parameters. Margins are 2cm on the sides, 3cm on the top, and 2cm on the bottom.

Generally, the smoke is grayish (dark gray, gray, light gray and white). And black smoke is generated by unburnt materials or a combustion at high temperatures. That is, a certain time has passed since the fire occurred. This paper focuses on the smoke of the initial generation, and sets the conditions as shown in Equation 5 to use smoke colors ranging from gray to white.

$$C = (R + G + B)/3\,,\ \tau_1 < C_L < \tau_2,\ \tau_3 < C_H < \tau_4 \tag{5}$$

where *C* is output image. This research set the C_L is low range minimum value between 80 and 150 and the C_H is upper range maximum value between 180 and 250. The average image *C* is histogramized into 256 bins (0 to 255) for each pixel. The values stored in each bin of the histogram are normalized using the input image size, and the sum is obtained, as in Equation (6).

$$H_{sum} = \sum_{i=0}^{255} b_i/(h * w) \tag{6}$$

where H_{sum} is the RGB color histogram result value. b_i means the histogram bins from 0 to 255 which is only included equation (5) range. And *h* and *w* is height and width for input image. The grayish color is distributed intensively from 80 to 150 and from 180 to 250.

Fire flames are usually bright orange or red (red -> orange -> yellow -> white -> mellow). This paper uses HSV color instead of RGB color. The range of HSV color used in the paper is as follows.

- H: 0 to 40
- S: 100 to 255
- V: 80 to 255

As shown in smoke color extraction, HSV color image is also calculated the average value for the filtered range image. And the HSV histogram is obtained by equation (6).

5. Experimental results

We proposed a new algorithm using similarity and color histogram of whole frame and local frame to reduce smoke false positive rate generated by fire detection system using ONVIF camera based on deep learning. In this study, an experiment was carried out with an ordinary user computer environment consisting of an Intel Core i7-7700 (3.5 GHz), memory 16G, Ge-force TITAN-X and OpenCV and Python 3.5 program. The flame and smoke databases used in this study was obtained from the internet, and general ground and factory directly recorded video. The video recording device was a mobile phone camera and Cannon G5 camera. The software used was basically Python, Tensorflow and Opencv.

In order to implement the proposed algorithm, the following process has been performed. The first is labeling dataset. The first working is labeling data using the LabelImg program, as shown figure 4. The labeling categories used in this paper are flame, smoke, Grinder, Welding, and human. The result of labeling data is stored in an .xml file that contains the object type name and the four point coordinates of the object area.

Second, it is a training process for labeled images. The .xml file should be converted to the learning data format of the Tensorflow. In the training process, the input image is a JPEG or PNG file. Since the meta data and labels of these images are stored in a separate file, the code becomes complicated when reading the training data because the image file must be read separately from the meta data and label file. In addition, performance degradation can occur if the image is read in JPEG or PNG format and decoded each time. But the TFRecord file format avoids the above performance degradation and makes it easier to develop. The TFRecord file format stores the height and width of the image, the file name, the encoding format, the image binary, and the labeling data rectangle position value, which indicates the position of the object in the image. Through this process, the entire training data is classified and stored as 70% training data and 30% validation data. The FASTER-CNN ResNet (Deep Residual Network) was selected as the basic model for training, and it is characterized by the smallest number of objects and the highest detection rate. The fire images used in the training are 21,230 pieces.

Third, we processed the extraction of training model. The learning process stores a check pointer indicating a result of learning for each predetermined pointer. Each check pointer has meta information about the model in the Tensorflow model file format and can be learned again. However, because there is a lot of unnecessary information in the ".meta" file, the .meta file needs to be improved to use the actual model. Finally, a ".pb" file is generated that combines the weights except for the unnecessary data in the ".meta" file.

In this paper, we used a factory, office, and natural environment recorded video images as the experimental data. Figure 5 shows an example of a continuous frame of video used in the experiment. Fire detection experiment was performed using ".pb" file based on Fater R-CNN model. Figure 6 shows fire and smoke detection results included true positive and false positive using general deep learning.

Figure 5. Example of the fire and smoke frame sequence of test videos.

(a)

(b)

(c)

Figure 6. The experimental results using the Faster R-CNN, (a) the results of true positive, (b) the results of false positive (fixed object), (c) the results of false positive (moving object).

Figure 6 (a) shows the result of the experiment to detect fire and smoke using various videos. The detection threshold of Faster R-CNN was 30%. Figure 6 (b) and (c) shows the result of false positive detection by applying deep learning training results. Although false positives have appeared in many places, there are three types of false positives. First, smoke or flame is detected by reflection of sunlight for fixed objects. Second, facilities inside and outside the factory show similar shapes and colors like smoke and fire. Third, when objects are moving around, deep learning system recognizes object as fire or smoke when a similar shape of trained fire and smoke appears, as shown in Figure 6 (c). Table 1 shows the fire and smoke detection results for the videos.

For Video 1 (fire video), the precision rate [13] is 98%, and the calculation formula is as shown in equation (7). The false negative is about 2%. The Video 2 is smoke test. It showed good result. False positive frames were 106, 166, 10, and 9 for Video 3, Video 4 and Video 5. From Video 3 to Video 5 are the inside video of factory, like as figure 6. In Table 1, Ground Truth represents the total number of frames in the video True Positive (TP) indicates when a fire and smoke is detected as fire and smoke. True Negative (TN) indicates that a non-fire objects are not detected as fire and smoke. False Positive (FP) is a case where non-fire objects are detected as a fire.

$$Precision = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP}) \tag{7}$$

Table 1. The results of video test using general Faster R-CNN

Videos	Ground Truth	True Positive	True Negative	False Positive
Video 1	1183	1158	0	25
Video 2	102	102	0	0
Video 3	14997	0	14891	106
Video 4	14658	0	14492	166
Video 5	12111	0	12101	10

In the case of Video 1 and Video2, these are not generated in the continuous frame. Since the video is 30 fps, it can be sufficiently compensated. However, in the case of Video 3 to Video 6, the alarm continues to ring and the stress of worker becomes higher. In order to reduce false positives generated in Video 2, we use the following characteristics. First, it is global check. We checked the motion characteristics before performing deep learning using equation mean square error (8) and frame difference (9) [14]. Since there is a motion when a fire occurs, if a block of moving pixels are generated, it is registered as a fire candidate state. If the fire candidate frame status (V_G) is 1, a deep learning process is performed, as shown Figure 1.

$$S_k = SSIM\ (f_i, f_j),\ M_k = MSE(f_i, f_j),\ A_k = diff\ (f_i, f_j) \tag{8}$$

$$V_G = \begin{cases} 1 & if\ \ S_k < th1, M_k > th2, A_k > th3 \\ 0 & else \end{cases} \tag{9}$$

Second, it is local check. If there is a trained class in the input frame image, a bounding box is created and stored as a local area of interest. The next step is to verify the local area of interest again. In this paper, we determine the final fire and smoke region using the color histogram H, SSIM, frame difference, and mean square error (MSE) with other frames as below equation,

$$V_G = \begin{cases} 1 & if\ \ S_k < th4, M_k > th5, A_k > th6,\ H_{sum\_F} > th7, H_{sum\_S} > th8 \\ 0 & else \end{cases} \tag{10}$$

where k means frames, from *th1* to *th8* are threshold value by experiment. We compared the local region (bounding box area) of interest using the 3 frame difference algorithm (first, middle, and last frames) from the stored 10 frame images.

Table 2 shows the experimental results using the proposed algorithm. In Video 3, the false positive rate dropped to 0% and the fire detection of Video 1 (fire video) persisted. The false positive rate of Video 4 and Video 5 were removed to almost 99.9%. False positives are very important in fire detection systems. This is a very confusing situation in large systems because it is not a fire but is recognized as a fire. Additionally, we tested other video (Video 6). It also marked zero false positive rate for the proposed method. The false positive rate for the additional 8 videos were 0%, and the image examples used in the video experiment are shown in Figure 7.

Table 2. The results of video test using prosed algorithm

Videos	Ground Truth	True Positive	True Negative	False Positive
Video 1	1183	1158	0	0
Video 2	102	102	0	0
Video 3	14997	0	14891	0
Video 4	14658	0	14492	3
Video 5	12111	0	12111	3
Video 6	13454	0	13454	0

Figure 7. The example of other videos test for the proposed algorithm.

The reason why the false positive did not disappear in the Video 4 and Video 5 experiment is that object is detected as the fire and smoke. This is the case where the shape of this part is deformed and detected as fire. It is necessary to adjust the final threshold considering various factors, or an algorithm using additional color and feature analysis for the fire.

6. Conclusion

The fire caused by the small spark is very terrible nature environment which is lose one's fortune and killed human lives. In this paper, we describe a new fire and smoke detection algorithm based on deep learning and color histogram from surveillance camera. Color is a very important factor in fire detection. In this paper, HSV color space is used for fire detection and RGB color space is used for smoke detection. In general, deep learning method using the shape of object frequently generate the false positive which general object is detected as the fire or smoke. To solve this problem, first, we used the motion detection using the three frame difference algorithm. And then we applied the frame similarity using SSIM and MSE. Second, we adapted the Faster R-CNN algorithm to find smoke and fire candidate region for the detected frame. Third, we decided the final fire and smoke area using the local HSV and RGB color histogram, frame similarity, and MSE for the candidate region. Experiments have shown that the probability of false positives of the proposed algorithm is significantly lower than that of conventional deep learning.

As a future work, it is necessary to study the fire color analysis and the experiment using the correlation of the frame and the deep learning model to further reduce false positives.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2018R1D1A1B07051087).

References

[1] B. C. Ko, S. J. Han, and J. Y. Nam, "Modeling and Formalization of Fuzzy Finite Automata for Detection of Irregular fire Flames," *IEEE Transaction on Circuits and System Video Technology*, Vol. 21, No. 12, pp. 1903-1912, December 2011.
DOI: 10.1109/TCSVT.2011.2157190.

[2] J. Gubbi and S. Marimuthu, "Smoke detection in video using wavelets and support vector machines," *Fire Safety Journal*, Vol. 44, No. 8, pp. 1110 - 1115, 2009.
DOI: https://doi.org/10.1016/j.firesaf.2009.08.003

[3] B. C. Ko, J. Y. Kwak, and J. Y. Nam, "Wildfire smoke detection using temporospatial features and random forest classifiers," *Optical Engineering*, Vol. 51, January 2012.
DOI: https://doi.org/10.1117/1.OE.51.1.017208

[4] S. Frizzi, R. Kaabi, M. Bouchouicha, J. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional Neural Network for Video Fire and Smoke Detection," *IECON 2016*, 2016.
DOI: 10.1109/IECON.2016.7793196

[5] T. Chen, P. Wu, and Y. Chiou, "An Early Fire-Detection Method Based on Image processing," *2004 International Conference on Image Processing*, pp. 1707-1707, October 2004.
DOI: 10.1109/ICIP.2004.1421401

[6] B. U. Toreyin and Y. Dedeoglu, and A. E. Cetin, "Wavelet Based Real-time Smoke Detection in Video," *EUSIPO 2005*, 2005.

[7] C. Yu, Z. Mei, and X. Zhang, "A Real-time Video Fire Flame and Smoke Detection Algorithm," *Procedia Engineering*, Vol. 62, pp. 891-898, 2013.
DOI: https://doi.org/10.1016/j.proeng.2013.08.140

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," *Proceeding of IEEE*, Vol. 86, No. 11, pp. 2278 – 2324, November 1998.
DOI: 10.1109/5.726791

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep convolutional neural Network: Advanced in Neural Information Processing System 25," *26th Annual Conference on Neural Information Processing Systems*, pp. 1106-1114, December 2012.
DOI: 10.1145/3065386

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, S. Ilya, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," Clinical Orthopedics and Related Research 2012, pp.1-18, July 2012.
DOI: 10.1.1.258.6363

[11] Y. Lee, I. Ansari, and J. Shim, "Rear-Approaching Vehicle Detection using Frame Similarity base on Faster R-CNN," *International Journal of Engineering and Technology*, Vol. 7, pp. 177-180, 2018.
DOI: 10.14419/ijet.v7i4.44.26979

[12] Wikipedia, Color histogram, *Web URL: https://en.wikipedia.org/wiki/Color_histogram/*

[13] Receiver operating characteristic, *Web URL:https://en.wikipedia.org/ wiki/Receiver_operating_ characteristic/*

[14] Y. Zhang, X. Wang, and B. Qu, "Three-Frame Difference Algorithm Research Based on Mathematical Morphology," *Procedia Engineering*, Vol. 29, pp. 2705-2709, December 2012.
DOI: 10.1016/j.proeng.2012.01.376