Threshold Value Optimization to Improve Fire Performance Classification Using HOG and SVM

by Yuslena Sari

Submission date: 24-Nov-2021 07:11PM (UTC+0700)
Submission ID: 1711964810
File name: to_Improve_Fire_Performance_Classification_Using_HOG_and_SVM.pdf (1.48M)
Word count: 3621
Character count: 19286

Threshold Value Optimization to Improve Fire Performance Classification Using HOG and SVM

Yuslena Sari
Department of Information Technology,
Faculty of Engineering
Universitas Lambung Mangkurat, Banjarmasin, Indonesia
yuzlena@ulm.ac.id

Reza Ramadhan
Department of Information Technology,
Faculty of Engineering
Universitas Lambung Mangkurat, Banjarmasin, Indonesia
31610817210015@mhs.ulm.ac.id

Puguh Budi Prakoso
Department of Civil Engineering,
Faculty of Engineering
Universitas Lambung Mangkurat, Banjarmasin, Indonesia
puguh.prakoso@ulm.ac.id

Ricardus Anggi Pramunendar
Department of Informatics Engineering,
Faculty of Computer Science,
Universitas Dian Nuswantoro, Semarang, Indonesia
ricardus.anggi@dsn.dinus.ac.id

2021 International Seminar on Application for Technology of Information and Communication (iSemantic) | 978-1-6654-2804-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISEMANTIC52711.2021.9573187

***Abstract*— Fire is a disaster caused by accidental fire, unexpected or unwanted, difficult to control, and detrimental. Many factors cause fires. Humans may directly or indirectly cause these factors, or they may also be caused by nature. Total 913 residential fires were recorded in Indonesia from 2013 to 2015. There are still many incidents of a fire occurring. This research is carried out to reduce the incident to detect fire objects by applying the histogram of oriented gradient (HOG) method and then classified using the support vector machine (SVM) method. The proposal is expected to help in the early identification of fires to reduce the number of fires that occur. This study has several stages: inserting images, converting images to grayscale, thresholding methods, object detection with HOG, and classification using SVM. The system implementation is carried out using the Python programming language with the Django web framework. The final result will be tested using a confusion matrix with four matrices: accuracy, precision, recall, and F1-Score. The results showed that the HOG and SVM methods using the RBF kernel and the sharing of training-test data 70:30 obtained accuracy of 88.33%, precision of 85.71%, recall of 88.89%, and F1-Score of 87. 27%.**

Keywords*— Fire, Computer Vision, Histogram of Oriented Gradient and Vector Machine Support

I. INTRODUCTION

Fire is a disaster caused by an accidental, unexpected/unwanted flame, difficult to control, and detrimental. Many factors cause fires. Humans may directly or indirectly cause these factors, or they may also be caused by nature. The flame that can cause fires also has various ignition sources, not only from direct sources of fire but also from various human activities that can indirectly cause the fires [1]–[6]. Based on data from the National Disaster Management Agency (BNPB), 913 residential fires were recorded in Indonesia from 2013 to 2015. In 2020 BNPB recorded 256 fire incidents that occurred from 1 January 2020 to 30 August 2020. In the city of Banjarmasin in particular, in 2013, it was recorded as 100 residential buildings that caught fire. In 2014 it is decreased to 99 incidents, and in 2015 there were 60 incidents, then increased at the end of 2015 to 92 incidents. A residential fire occurred in Alalak Selatan Sub-District, Banjarmasin City, in 2019. The fire destroyed 65 houses and killed 222 people [7], [8]. In general, the causes of residential fires include defective or unsafe electrical installations, cooking equipment such as the inappropriate use of LPG (liquified petroleum gas), and the occupants' misbehavior, such as lighting candles near flammable items [7].

Based on this, this research proposes a prototype for a fire detection system. The detection is applied to the image inputted into the system. Using the support vector machine (SVM) classification method modeling based on the histogram of oriented gradients (HOG) extraction method, it is hoped that the prototype can recognize images experiencing fires.

II. RESEARCH METHODOLOGY

A. Research Materials

The material or object used in this study is a collection of data collected by Sulenn [9]. The dataset contains fire images divided into burning vehicles, burning houses or buildings, and burning rooms. The pictures are an incident where the fire has spread about the objects around it. Some sample images in the form of fires are shown in Fig. 1.

This dataset also contains pictures of candle flames and pictures of rooms in normal conditions (non-fire). The dataset contains flames or objects that resemble the color of fire but do not spread as large as a fire image. Some examples of pictures with normal good condition containing fire are shown in Fig. 2.

Fig. 1. Fire image data (a, b, c, and d)

Fig. 2. Standard image data (a, b, c, and d)

This study uses 200 images divided into 100 fire images and 100 typical non-fire images from the data obtained. The usual fire images do not fire as many as 25 images and 75 images in standard rooms that do not experience a fire. The data collected has various resolutions and aspect ratios. However, to facilitate processing, all data is processed to have the same size and aspect ratio by using a 3rd party application in the form of FastStone Viewer to help with this. Several studies [10]–[12] stated that the difference in image resolution and aspect ratio gave difficulty during the classification process. The image size is equalized to 259 x 146 with an aspect ratio of 16: 9 and is an image with the extension 'JPG.'

B. *Research Procedure*

Several procedures in this research were carried out to facilitate the implementation of the research. As a guide so that the objectives of this study were achieved. The procedure begins with a literature review to conduct research testing. The research procedure can be seen in Fig. 3.

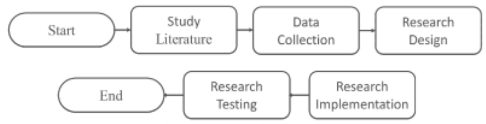

Fig. 3. Research Procedures

1) Literature Study: Literature study is the initial stage of this research, which includes previous research on the recognition or classification of objects in images using the histogram of oriented gradient (HOG) method [13] and support vector machine (SVM) [14].

2) *Data Collection:* The next stage is data collection that will be used in this research. The data used is in the form of fire images divided into images of burning vehicles, images of burning houses or buildings, and images of burning rooms. The data used also include images of candlelight and pictures of a standard room.

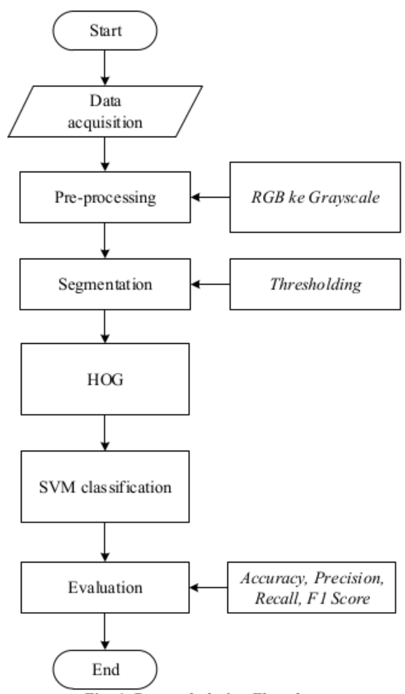

Fig. 4. Research design Flowchart

3) Research Design: The research design is shown in Fig. 4, which contains this research proposal. The proposal is in the form of a new workflow to perform fire detection on images consisting of data acquisition, pre-processing in the form of color conversion, segmentation based on thresholding according to research [15], feature extraction using HOG [13], classification using SVM [14], and evaluation.

The design that is made has several processes in it, starting from inserting the image. The image or picture is processed, namely changing the image to grayscale to avoid interference from the background, and is used to minimize the brightness effect. The resulting images are shown in Fig. 5 and Fig. 6.

Fig. 5. Original image (before processing)

Fig. 6. Image after processing into grayscale)

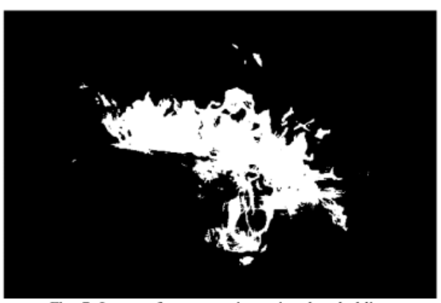

Fig. 7. Image after processing using thresholding

Furthermore, object segmentation includes edge detection and thresholding methods to separate the background and foreground from the image [16]–[20]. Can be seen in Fig. 7.

Then the object is detected using the HOG method by dividing the image into several cells, then grouped into blocks that overlap with each other [19], [21]. The resulting image is shown in Fig. 8.

Fig. 8. Image after extracted using the HOG method

Furthermore, the SVM classification is to classify whether the detected object is a fire or not. Then measure the performance of the HOG and SVM methods in classifying fires by measuring accuracy, precision, recall, and F1-score.

4) *Research Implementation:* The implementation stage will include data processing results and the research design developed and implemented using the HOG and SVM methods. The research design will be built using the Python programming language.

5) *Research Testing*: In testing this study, the confusion matrix is used to evaluate performance with the results of accuracy, precision, recall, and F1-score.

III. RESULTS AND DISCUSSION

A. System Flow

The system's flow in this study starts from image input, image processing, image segmentation, object detection with HOG, and classification with SVM.

1) Data Acquisition: The first step of the system flow is data acquisition. In this process, the inserted image is the data collected by Sulenn, which is then processed using the FastStone Image Viewer application to equalize the resolution and aspect ratio. This process uses the built-in library in Python, the OpenCV library, and uses the *cv.imread* function.

2) Image Processing: In this process, images that have been inserted before being processed from RGB to grayscale. Image processing is done to avoid distraction from the background and is used to minimize brightness. This process uses the built-in library in Python, namely the OpenCV library, and uses cv's color conversions function.COLOR_BGR2GRAY. The results of the

image processing process are shown in Fig. 9 and Fig. 10.

a b

Fig. 9. Original Image

a b

Fig. 10. Image After processing

3) Image Segmentation: In the image segmentation process, the thresholding method is carried out to separate the background from the foreground from the image. This process uses the built-in library in Python, the OpenCV library, and uses the cv.threshold function.

4) Detection of objects with HOG: After segmenting the image using the thresholding method. Then the object is detected using the feature description method from HOG by dividing the image into several cells, then grouped into blocks that overlap with one another. This process uses an additional library, the skimage library. After the image has been extracted using the HOG method, each image is totaled, and the results are entered into Microsoft Excel.

5) Classification of objects using SVM: After the image data is processed using the HOG method, the results are entered into Microsoft Excel. Furthermore, the SVM method classification determines whether the object is classified as fire or normal (no fire occurred). This classification process will be tested with three types of ratio data combinations divided into training data and test data, namely 70:30, 50:50, and 30:70 [14], [22].

Splitting the data uses the scikit-learn model_selection library with the train_test_split function. The classification code with the SVM method will use the library from the scikit-learn SVM using the RBF kernel [22].

B. *Interface Implementation*

This implementation is a prototype processing stage by applying the HOG and SVM methods. The system interface is needed to make it easier to operate the built prototype. The system will be built using the Django web framework and Python [23], [24]. With a framework, it can help programmers create a website faster, compared to writing code from scratch. The following is the result of implementing the main page interface and the classification page.

1) Main Page Interface Implementation: This main page contains background information on the research in applying the HOG and SVM methods to classify fires. Implementation of the interface or main page display can be seen in Fig. 11.

Fig. 11. Main page interface

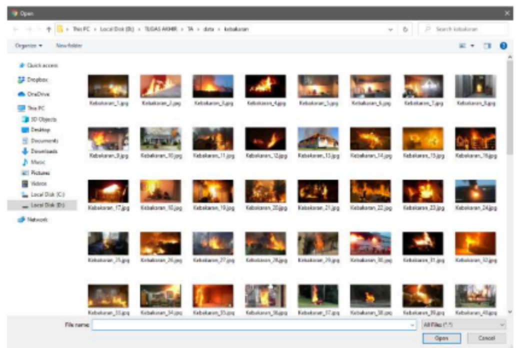

Fig. 12. insert image Interface

Fig. 13. Interface after inserting the image

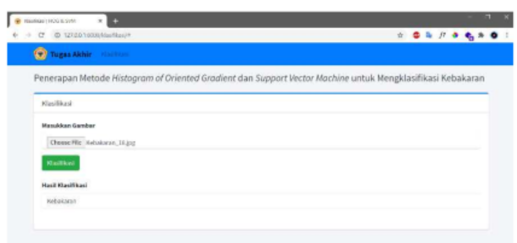

Fig. 14. *Interface after classification*

2) Implementation of Classification Page Interface: On this classification page, selecting images that the prototype will process is carried out using the HOG and SVM methods. The first step in this classification process is to enter an image by clicking on the choose file then selecting the image to be classified. The interface for entering the image can be seen in Fig. 12, and the figure shows the input for image data. After the image has been selected, as in Fig. 13, it will return to the initial view and show a button to classify; besides, it also shows the image address. the next step is to click on the button. Classification to find out the classification results of the image that has been selected, whether the image is classified as fire or normal (non-fire). Implementation of the interface or classification page display can be seen in Fig. 14.

C. Testing and Discussion

This testing phase will be divided into several test scenarios that measure the threshold value by paying attention to the performance results in a confusion matrix (accuracy, precision, recall, and F1-score).

1) Test Scenario 1: In this test scenario 1, the data used is the same as the amount of data that has been previously collected, namely 100 fire images and 100 standard images, where the standard image data consists of 25 pictures of candle fires and 75 pictures of rooms with normal condition based on [4], the threshold value to be used in the image segmentation process is 100, which is the gray value of the image. Image intensity values greater than or equal to the threshold will be converted to white, and image intensity values less than the threshold will be converted to black. Therefore, the output image from the threshold result is a binary image. Then the classification test is carried out to determine the performance of this test scenario one with data divided into 70:30. After testing, the performance of each confusion matrix is obtained, which can be seen in Table 1.

TABLE I. 1ST TEST SCENARIO PERFORMANCE RESULTS

Matrix	Result using 70 *training* and 30 *test*
Accuracy	80,00%
Precision	85,71%
Recall	75,00%
F1-*Score*	80,00%

2) Scenario Test 2: In the 2nd test scenario, the data used is the same as test scenario 1. In this test, the threshold value used in the image segmentation process is from 70 to 150, intending to know the level of change in the accuracy value. Will be obtained. Classification testing is still the same as testing in scenario one that divided the data into 70:30. After testing, the performance of each confusion matrix is obtained, which can be seen in Table 2. That table it can be seen that there is an increase in the overall performance of the matrix when the threshold value used is 130 and decreases when the threshold value exceeds 130. The highest performance reached 88.22% accuracy, 85.71% precision, 88.89% recall, and an F1 score of 87.27%.

3) Testing Scenario: This step is based on test scenario 2. The threshold value to be used in test scenario three is 130. In this test scenario 3, the amount of data used is still the same as the previous scenario. In testing scenario 3, the amount of data will be carried out in three combinations of data sharing in the classification process to determine test scenario three. Three types of ratio data combinations will be tested in this test: divided into training data and test data, namely 70:30, 50: 50, and 30:70. The test results can be seen in Table 3, which shown the highest performance value than the others and are obtained by applying the amount of training of 70 with testing of 30. The performance obtained from the distribution of these data reaches 88.33% for accuracy, 85.71% for precision, 88, 89% for recall, and 87.27% for F1 score.

TABLE II. 2ND TEST SCENARIO PERFORMANCE RESULTS

Threshold	Result (*training-test split* 70:30)			
	Accuracy	*Precision*	*Recall*	*F1-Score*
70	66,67%	75,00%	61,76%	67,74%
80	73,33%	89,29%	65,79%	75,76%
90	80,00%	89,29%	73,53%	80,65%
100	80,00%	85,71%	75,00%	80,00%
110	83,33%	82,14%	82,14%	82,14%
120	83,33%	82,14%	82,14%	82,14%
130	88,33%	85,71%	88,89%	87,27%
140	86,67%	82,14%	88,46%	85,19%
150	83,33%	82,14%	82,14%	82,14%

TABLE III. 3RD TEST SCENARIO PERFORMANCE RESULTS

Training: Test Split	Results			
	Accuracy	*Precision*	*Recall*	*F1-Score*
70:30	88,33%	85,71%	88,89%	87,27%
50:50	82,00%	88,89%	75,47%	81,63%
30:70	78,57%	84,29%	75,64%	79,73%

IV. CONCLUSION

Based on the results of an experiment, it can be concluded that the histogram of oriented gradient (HOG) and support vector machine (SVM) method with the RBF kernel produces good accuracy. This model is successfully applied using Python programming on the Django web framework for detecting the fire in the images. However, the recommended threshold value in the image segmentation process using the HOG method is 130. The highest performance was obtained using the training-test split 70:30, getting an accuracy result of 88.33%, precision of 85.71%, recall of 88,89%, and an F1-score of 87.27%. These results show that the prototype system created can provide good performance and can be applied runtime to the running system.

REFERENCES

[1] K. Chen, Y. Cheng, H. Bai, C. Mou, and Y. Zhang,

"Research on Image Fire Detection Based on Support Vector Machine," 2019, doi: 10.1109/ICFSFPE48751.2019.9055795.
[2] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional Neural Networks Based Fire Detection in Surveillance Videos," *IEEE Access*, 2018, doi: 10.1109/ACCESS.2018.2812835.
[3] H. Xu, W. Yuan, and M. Yu, "Real-time Classification of Ship Cabins Fire Hazard Levels Based on CNN," 2019, doi: 10.1109/CAC48633.2019.8996567.
[4] A. F. Zulkarnain, Y. Sari, and R. Rakhmadani, "Monitoring System for Early Detection of Fire in Wetlands based Internet of Things (IoT) using Fuzzy Methods," *IOP Conference Series: Materials Science and Engineering*, vol. 1115, no. 1, p. 012007, 2021, doi: 10.1088/1757-899x/1115/1/012007.
[5] K. Poobalan and S. C. Liew, "Fire detection based on color filters and Bag-of-Features classification," 2015, doi: 10.1109/SCORED.2015.7449362.
[6] G. F. Shidik, F. N. Adnan, C. Supriyanto, R. A. Pramunendar, and P. N. Andono, "Multi color feature, background subtraction and time frame selection for fire detection," in *2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems*, Nov. 2013, no. November, pp. 115–120, doi: 10.1109/ROBIONETICS.2013.6743589.
[7] Kompas, "BNPB: There were 1,928 disasters throughout 2020," 2020. .
[8] BNPB, "Data on Residential Fire Disasters," 2020. .
[9] Sulenn, "Fire Dataset," 2020. .
[10] S. Battiato, G. M. Farinella, G. Puglisi, and D. Ravi, "Content-aware image resizing with seam selection based on Gradient Vector Flow," 2012, doi: 10.1109/ICIP.2012.6467310.
[11] K. Mishiba, M. Ikehara, and T. Yoshitome, "Content aspect ratio preserving mesh-based image resizing," 2012, doi: 10.1109/ICIP.2012.6466997.
[12] T. Marciniak, A. Chmielewska, R. Weychan, M. Parzych, and A. Dabrowski, "Influence of low resolution of images on reliability of face detection and recognition," *Multimedia Tools and Applications*, 2015, doi: 10.1007/s11042-013-1568-8.
[13] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian Detection using Infrared images and Histograms of Oriented Gradients," in *2006 IEEE Intelligent Vehicles Symposium*, 2006, pp. 206–212, doi: 10.1109/IVS.2006.1689629.
[14] M. Ahmad and S. Aftab, "Analyzing the Performance of SVM for Polarity Detection with Different Datasets," *International Journal of Modern Education and Computer Science*, 2017, doi: 10.5815/ijmecs.2017.10.04.
[15] M. A. Soeleman *et al.*, "Adaptive threshold for moving objects detection using gaussian mixture model," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, pp. 1122–1129, 2020, doi: 10.12928/TELKOMNIKA.V18I2.14878.
[16] S. Koike, "Analysis of adaptive threshold nonlinear correlation algorithm," 2017, doi: 10.1109/ISCIT.2017.8261164.
[17] M. I. Bogachev, O. A. Markelov, and V. Y. Volkov, "Adaptive Object Selection Based on Multi-Threshold Image Processing," 2019, doi: 10.1109/WECONF.2019.8840585.
[18] Y. Sari and P. B. Prakoso, "Detection of Moving Vehicle using Adaptive Threshold Algorithm in Varied Lighting," *2018 5th International Conference on Electric Vehicular Technology (ICEVT)*, pp. 136–141, 2018.
[19] D. Jayashree, S. Pavithra, G. Vaishali, and J. Vidhya, "System to detect fire under surveillanced area," 2017, doi: 10.1109/ICONSTEM.2017.8261285.
[20] Y. Sari, P. B. Prakoso, and A. R. Baskara, "Application of neural network method for road crack detection," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 4, pp. 1962–1967, 2020, doi: 10.12928/TELKOMNIKA.V18I4.14825.
[21] G. Lian, "Pedestrian detection using quaternion histograms of oriented gradients," 2020, doi: 10.1109/ICPICS50287.2020.9202071.
[22] R. Medar, V. S. Rajpurohit, and B. Rashmi, "Impact of Training and Testing Data Splits on Accuracy of Time Series Forecasting in Machine Learning," 2018, doi: 10.1109/ICCUBEA.2017.8463779.
[23] C. Song, R. Huo, S. Wang, and C. Lv, "Transformer Equipment Temperature Monitoring Based on the Network Framework of Django," 2019, doi: 10.1109/CAC48633.2019.8996768.
[24] S. Liawatimena *et al.*, "Django Web Framework Software Metrics Measurement Using Radon and Pylint," 2019, doi: 10.1109/INAPR.2018.8627009.

Threshold Value Optimization to Improve Fire Performance Classification Using HOG and SVM

ORIGINALITY REPORT

13%	7%	13%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

3%

★ "Table of Content", 2021 International Seminar on Application for Technology of Information and Communication (iSemantic), 2021

Publication

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		