IJACSA
WHERE WISDOM SHARES

International Journal of Advanced Computer Science and Applications

Volume 13 Issue 9

September 2022

ISSN 2156-5570(Online)
ISSN 2158-107X(Print)

SAI

www.ijacsa.thesai.org

Editorial Preface

From the Desk of Managing Editor…

It may be difficult to imagine that almost half a century ago we used computers far less sophisticated than current home desktop computers to put a man on the moon. In that 50 year span, the field of computer science has exploded.

Computer science has opened new avenues for thought and experimentation. What began as a way to simplify the calculation process has given birth to technology once only imagined by the human mind. The ability to communicate and share ideas even though collaborators are half a world away and exploration of not just the stars above but the internal workings of the human genome are some of the ways that this field has moved at an exponential pace.

At the International Journal of Advanced Computer Science and Applications it is our mission to provide an outlet for quality research. We want to promote universal access and opportunities for the international scientific community to share and disseminate scientific and technical information.

We believe in spreading knowledge of computer science and its applications to all classes of audiences. That is why we deliver up-to-date, authoritative coverage and offer open access of all our articles. Our archives have served as a place to provoke philosophical, theoretical, and empirical ideas from some of the finest minds in the field.

We utilize the talents and experience of editor and reviewers working at Universities and Institutions from around the world. We would like to express our gratitude to all authors, whose research results have been published in our journal, as well as our referees for their in-depth evaluations. Our high standards are maintained through a double blind review process.

We hope that this edition of IJACSA inspires and entices you to submit your own contributions in upcoming issues. Thank you for sharing wisdom.

Thank you for Sharing Wisdom!

Kohei Arai
Editor-in-Chief
IJACSA
Volume 13 Issue 9 September 2022
ISSN 2156-5570 (Online)
ISSN 2158-107X (Print)

Editorial Board

CONTENTS

Real Time Fire Detection using Color Probability Segmentation and DenseNet Model for Classifier

Faisal Dharma Adhinata[1]
Faculty of Informatics
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia

Nur Ghaniaviyanto Ramadhan[2]
Faculty of Informatics
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia

***Abstract*—The forest is an outdoor environment not touched by the surrounding community, so it is not immediately handled when a fire occurs. Therefore, surveillance using cameras is needed to see the presence of fire hotspots in the forest. This study aims to detect hotspots through video data. As is known, fire has a variety of colors, ranging from yellow to reddish. The segmentation process requires a method that can recognize various fire colors to get a candidate fire object area in the video frame. The methods used for the color segmentation process are Gaussian Mixture Model (GMM) and Expectation–maximization (EM). The segmentation results are candidates for fire areas, which in the experiment used the value of K=4. This fire object candidate needs to be ascertained whether the segmented object is a fire object or another object. In the feature extraction stage, this research uses the DenseNet-169 or DenseNet-201 models. In this study, various color tests were carried out, namely RGB, HSV, and YCbCr. The test results show that RGB color produces the most optimal training accuracy. This RGB color configuration is used to test using video data. The test results show that the true positive and false negative values are quite good, 98.69% and 1.305%. This video data processing produces fps with an average of 14.43. So, it can be said that this combination of methods can be used to process real time data in case studies of fire detection.**

Keywords*—Fire detection; color segmentation; GMM-EM; DenseNet; real time

I. Introduction

Human daily life cannot be separated from the heat energy produced by a fire. The heat energy from this fire is often used for cooking, lighting candles as a light source, and burning garbage. However, fires can be catastrophic if they are not controlled and burn a large area. Fires can occur in indoor and outdoor environments such as forests. In Indonesia, forest fires often occur in Sumatra and Kalimantan because forest areas are still common [1]. Natural factors and human error can cause the emergence of fire hotspots. Some natural factors are hot weather, wind, and chemical reactions [2]. Then human error can occur due to forgetfulness in activities with fire, especially in rural areas that still use firewood for daily life [3]. Currently, the government has made efforts to mitigate fire disaster management [4], but the efforts made have not used Artificial Intelligence technology for automation. Therefore, the need for prevention efforts by detecting hotspots as early as possible before the fire spreads. This hotspot detection process can be done by installing an intelligent camera programmed using Artificial Intelligence to identify hotspots.

Several researchers have developed early detection of hotspots, including fire detection using video [5] and sensors [6]. Limitations in using sensors, especially gas sensors, can occur when there is other smoke, for example, people smoking. Then the heat sensor can also go wrong when the weather is hot. Using fire detectors through sensors also costs a lot when used in an outdoor environment because they must replicate the tool at many points. So, the proposed research focuses on using video to detect hotspots. Video data can be obtained by installing a Closed-Circuit Television (CCTV) camera. CCTV camera can detect fires using digital image processing and computer vision technologies, known as image-based fire detection. The advantages of image-based fire detection compared to conventional fire detectors can be installed in a large, open area to reduce expenses. The use of video data requires a method that can run in real-time [7]. Besides, the video resolution also affects the detection accuracy results. In a previous study [2], the fire detection system produced a reasonably accurate accuracy, but the processing time of each frame could not be done in real time. It is also a limitation of previous research. The main steps that affect the speed and accuracy are image segmentation, feature extraction and classification.

The segmentation process is carried out to take the fire area in the video frame. The segmentation stage of searching candidate fire object is very important to separate the fire candidate object from the background, which should not enter the feature extraction stage. The color of fire is a combination of various colors, ranging from reddish to yellow [8]. Previous studies conducted experiments using fire color segmentation, including RGB, HSV, and YCbCr color features, have not produced optimal accuracy [9]. The lack of this feature is because the color of the fire changes due to the wind. Therefore, this research uses a segmentation method that can overcome the quick color change of the fire using probability. This probability makes several color combinations of fire. The proposed research uses color probabilities to perform segmentation. In other case study research, the segmentation process was carried out on the image using a combination of the Gaussian Mixture Model (GMM) and Expectation–maximization (EM) methods [10][11]. The segmentation results show that combining these methods can detect multi-colored objects. Therefore, the proposed research uses of GMM-EM for the segmentation of candidate fire objects contained in video frames.

After the candidate fire object is obtained, the fire object must be sure that the segmented one is a fire object. Several studies of feature extraction and classification use the transfer learning method. The transfer learning of DenseNet201 model is used for image classification [12]. The results showed good accuracy for the feature extraction of corn disease. In another research, the DenseNet model was also used for feature extraction of the lungs affected by Covid-19 [13]. The results showed good accuracy using the DenseNet model. This research will also use the DenseNet model at the feature extraction and classification stage. The result of this research is a real-time fire early detection system using video data.

This research aims to build a real-time fire point detection system using video data for early warning of fires. Speed is an important thing in this study to be evaluated. The color of fire that is not only yellow requires a precise segmentation process, so the proposed method uses a combination of various colors of fire. The segmentation results are then extracted and classified to ensure that the object is a fire. Overall, the contributions of this research are:

- The use of various color combinations of fire to perform the segmentation process for searching fire object candidates.
- Evaluation of the segmentation process on each video frame to minimize non-fire object detection errors.
- The use of transfer learning as feature extraction and classification to achieve optimal accuracy and real-time processing.

II. Related Work

The development of fire detection applications often uses sensors [14]. The downside of using this sensor depends on the surrounding weather. When using a heat sensor during the dry season, the sensor may experience error detection. Then the gas sensor can also experience an error when there is other smoke, for example, cigarette smoke, smoke from burning garbage, etc. Even detecting hotspots in open areas, such as forests, is very difficult. Therefore, the fire detection uses video data.

Several researchers who process fire video data, including Khan et al. [15], used a fire's color, perimeter, area, and roundness for an indoor fire case study. The method used does not consider small fires, so it cannot carry out early detection of hotspots. Then, research by Thepade et al. [9] used a color combination of HSV and YCbCr to detect hotspots. The method used is still static, so the use of dynamic video data cannot be done. The segmentation process can also use the deep feature [16]. This deep feature is suitable for high-resolution images such as satellite images. Several segmented objects produce relatively good accuracy. However, the disadvantage of using deep features is that the processing time is quite long, so it is unsuitable for real-time processing. The color component of fire is not only red, but a combination of various colors, including yellow, orange, red, white, and blue. Previous research by Dong Keun Kim [10] used Gaussian Mixture Model (GMM) and Expectation–maximization (EM) to detect color combinations on objects. The proposed research will segment fire objects with fire color data training using the GMM-EM method. Video resolution also affects the detection accuracy results. In a previous study [2], the fire detection system produced a fairly accurate accuracy of 99.7%, but the processing time of each frame took 0.23 seconds or four fps. Therefore, this research proposes a new approach to obtain optimal accuracy and can run in real-time.

Currently, deep learning is a method that researchers often use for classification case studies. Deep learning, frequently used to handle picture data, is called Convolutional Neural Network. Deep learning is a technique used by artificial neural networks to manage input data utilizing multiple hidden layers. The output of this process is a non-linear modification of the input data used to determine the output value [17]. Deep learning is typically used for vast amounts of data. However, the data is relatively small in some instances, such as in this fire detection scenario. Transfer learning is a strategy for processing small amounts of data in which the model has been trained using other data [18]. DenseNet is an example of a transfer learning model. In this research, an evaluation of the DenseNet-169 and DenseNet201 models will be carried out.

III. Proposed Method

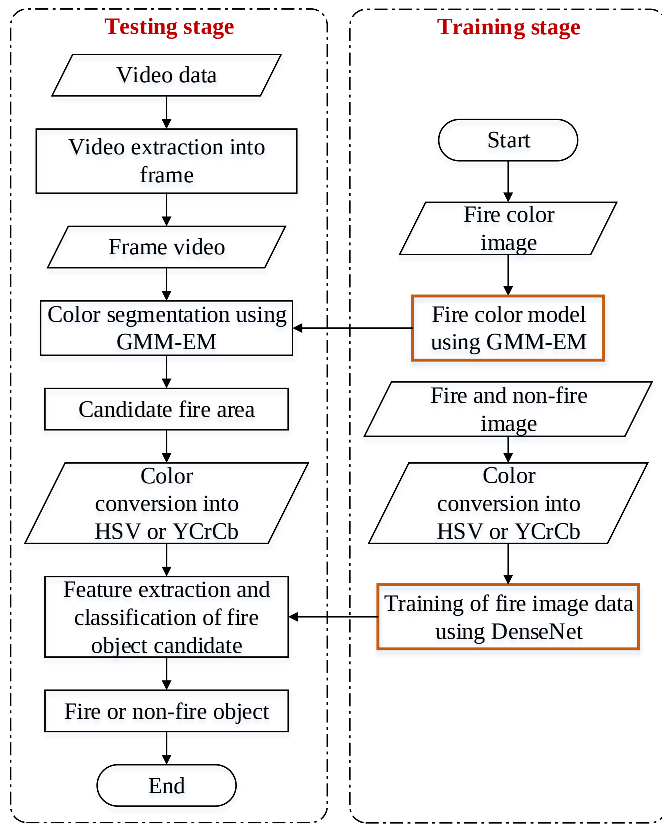


Fig. 1. Proposed System of Fire Detection.

The hotspot detection system starts from the training stage. There are two training processes: training for the segmentation process and feature extraction on fire objects. The training process uses a combination of GMM and EM methods. The fire segmentation process uses image data of fire colors. Then, the feature extraction process uses fire and non-fire image data. Before the training process, the image is converted to HSV or YCbCr color. The training process uses the transfer learning method. The best model is used for matching video data. Then the testing phase begins with real-time video data input. Video data is extracted in the form of frames. Detection of fire candidates in video frames is done by matching the color model. Flame object candidates are converted to HSV or YCbCr color. The conversion results are matched with the feature extraction model. The feature extraction and classification stage use DenseNet model. The results of the classification are fire and non-fire objects. Fig. 1 shows the flowchart of this research's fire point detection.

A. *Acquisition Data*

This research used two datasets: a dataset for segmentation of fire object candidates and a dataset for fire object classification. The dataset for segmentation uses 30 images of fire color images. This dataset uses three color channels: Red, Green, and Blue (RGB), measuring 100 x 100. The features of this dataset were taken from it based on the RGB color model, which was used to show the different colors of fire in the color probability model. The fire color varies so that it can detect various colors of fire when testing using video data. Fig. 2 shows an example of a fire color dataset used for the segmentation process.

Fig. 2. Fire Color Data for Segmentation Stage.

Then at the feature extraction stage, the data uses from Kaggle created by Jadon et al. [19]. This dataset consists of two classes, namely the fire class in various places and the non-fire class like other objects. The number of fire data is 1123 images, while the non-fire data is 1301. It was made by taking images of fire and things that don't fire under challenging situations, like the fire image in the forest and the non-fire image with things that look like fire in the background. At the training stage, the percentage of training data used is 80%, while the testing data is 20%. Fig. 3 shows an example of training data for the feature extraction stage.

Fire image

Non-fire image

Fig. 3. Fire and Non-Fire Image for Feature Extraction Stage.

B. *Fire Object Segmentation*

Multiple clusters can describe a dataset's distribution. Modelling a dataset with a single mean (one Gaussian) and estimated parameters is not optimal. For example, if a dataset contains two means of 218 and 250, the average may be close to 221. It is not a precise estimate. Multiple Gaussians with means of 218 and 250 provide a more accurate representation of the distribution of the data set.

In situations where multiple data sets with varying numbers of clusters describe the same feature, it is preferable to model the data across the three sets using a multivariate Gaussian [20]. Equation (1) represents the multivariate Gaussian equation. It allows for a more precise evaluation of the distribution of clusters across the provided data.

$$N(x|\mu,\Sigma) = \frac{1}{(2\pi|\Sigma|)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\} \quad (1)$$

This research employs a multivariate Gaussian with three color channels: Red, Green, and Blue. It has detected fire-based objects, so the number of clusters in each color channel will be examined. This research uses the Expectation-Maximization algorithm to estimate the means and covariances and determine the probability of a pixel belonging to a cluster. The total image is then modelled with a three-dimensional Gaussian.

The following sections outline the stages involved in performing the EM algorithm:

1) *Using some random numbers*: initialize the means and covariances. The covariance matrix must have the shape (dim, dim), where dim is the Gaussian's dimension number. These values are stored in a dictionary data structure called 'parameters.'

2) *E Stage*: Gaussians are combined in (2).

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \quad (2)$$

These are the probabilities associated with a given value x. It can accomplish this by applying the Bayes rule as (3).

$$= \frac{\pi_k N(x|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x|\mu_j,\Sigma_j)} \qquad \text{where, } \pi_k = \frac{N_k}{N} \quad (3)$$

These are saved in an array named 'cluster prob' with the dimensions (n_feat,K). n_feat is the number of rows in the dataset in this case.

3) *M Stage*: Then, update the means and covariances. This can be accomplished using the following (4).

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)\,(x_n-\mu_j)(x_n-\mu_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

$$\pi_J = \frac{1}{N}\sum_{n=1}^{N} \gamma_j(x_n) \quad (4)$$

4) *Calculate the Log Likelihood*: The objective is to increase the log likelihood function until the change in likelihood is equal to or less than a specified number. The following (5) is used to get the log likelihood.

$$\ln p(X|\mu,\Sigma,\pi) = \sum_{n=1}^{N} ln\{\sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k)\} \quad (5)$$

The log likelihood is appended to the log likelihoods array. The log probability difference is calculated by subtracting the most recent value from the most recently stored value. This technique is continued iteratively until the difference in log likelihood reaches a predefined value or the maximum number of iterations is reached. The final values produced are the dataset's estimated means and covariances. Fig. 4 shows an example of segmentation results.

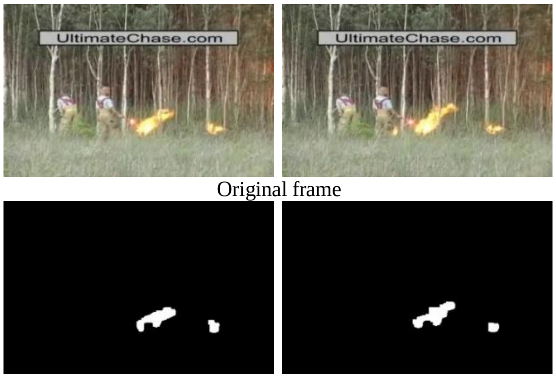


Fig. 4. Fire and Non-Fire Image for Feature Extraction Stage.

C. Feature Extraction and Classification on Fire Object Candidate

In this study, the feature extraction phase uses HSV or YCbCr colors. The image of the flame candidate is converted to HSV or YCbCr color space. The results of this conversion are utilized in the process of feature extraction. The process of feature extraction employs the DenseNet transfer learning model [21]. A DenseNet is a convolutional neural network with dense connections between layers using Dense Blocks, where all layers (with matching feature-map sizes) are directly connected. Maintain the feed-forward nature; each layer receives additional inputs from all preceding levels and sends its feature maps to all subsequent levels.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 densenet201 (Functional)    (None, 1, 1, 1920)        18321984

 flatten (Flatten)           (None, 1920)              0

 dense (Dense)               (None, 256)               491776

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 18,814,017
Trainable params: 492,033
Non-trainable params: 18,321,984
_________________________________________________________________
```

Fig. 5. Architecture for Training Stage on Fire and Non-Fire Image.

At the training stage, the preprocessing process such as resizing to 224x224 pixels and normalizing the data after resizing the data. The training data image uses a method with the DenseNet model, which includes feature extraction and classification processes. The DenseNet model has general operations for batch normalization, ReLU activation, and convolution. DenseNet model with 201 layers has dense block 1, transition layer 1, dense block 2, transition layer 2, dense block 3, transition layer 3, dense block four, and classification layer processes that produce the output model with .h5 format. In this research, feature extraction will be carried out using the DenseNet-169 and DenseNet-201 models. Fig. 5 shows the architectural configuration used for the training process.

D. System Evaluation

Using video data in a fire detection system requires evaluation, especially regarding the accuracy and speed of processing video frames. The first test is carried out at the segmentation stage. The segmentation process is used to find the fire object candidate area in the video frame. This research evaluates the value of K used in the GMM method against the segmentation results and the resulting fps. Then at the feature extraction stage, this research assesses the use of RGB, HSV, and YCbCr colors to see the results of training accuracy. The last configuration is used for evaluation using video data to see the true positive and false negative values of the video data matching results. The last is an evaluation of the video data processing speed to see the fps value.

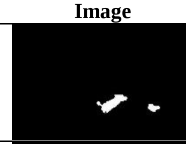
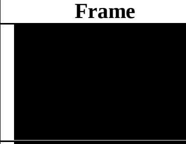
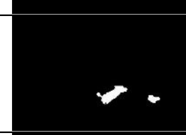
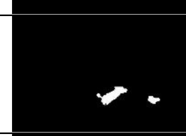
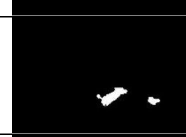
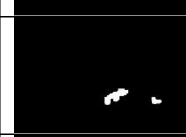
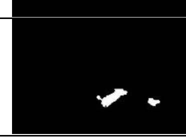
IV. Result and Discussion

This section conducts some experiments at the segmentation stage, feature extraction and classification of fire or non-fire objects. The last is matching using video data. In this experiment, this research used a computer with Core i5 specifications with 8GB of RAM and VGA GTX 1650. Then, the computer program uses Python programming language.

A. Fire Object Segmentation

At the segmentation stage using the GMM and EM methods, the most influential parameter is K value, which functions as a clustering dataset of fire colors. Table I shows the results of the variation of the K value on the segmentation results.

TABLE I. The Effect of K Value on Segmentation Results

K Value of GMM	Ground Truth Image	Segmentation on Frame	Fps
2			31.28
3			29.22
4			20.66
5			17.65
6			15.13

Based on the experiment using the K value in the GMM method, there are no segmented fire objects when the value of $K = 2$. Whereas in the ground truth image, there are two fire objects contained in the image. Then at the value of $K = 3$ to

6, the segmentation results show two fire objects with the same ground truth. However, if it looks closely, the more K values are added, the closer the segmentation results get to the ground truth shape. In processing video data also need to pay attention to the resulting speed. In this experiment, it was tested with the resulting fps value. As the value of K increases, the resulting fps also decreases. Because the number of clusters is increasing, it takes time to match each cluster. Therefore, this research choses a value of $K = 4$, which still produces an average of 20 fps. This configuration will be used to test using multiple videos containing fire objects.

B. Feature Extraction and Classification on Candidate Fire Object

This transfer learning model uses to process features into the feature extraction layer before the classification layer. The feature extraction used in this study is the DenseNet-169 or DenseNet201 model. The difference between DenseNet-169 and DenseNet-201 is the number of parameters. In DenseNet-169 it is 14.3M, while in DenseNet-201, it is 20.2M [22]. This study's training process configuration uses an image input size of 50 x 50. Then the distribution of training data and testing data is 80% training data and 20% testing data. Then the optimizer used is Adam with a loss configuration using binary_crossentropy because the number of classes used is two, namely fire and non-fire. Table II shows training results using two DenseNet models by monitoring validation accuracy. This research experimented with three colors, namely RGB, HSV, and YCbCr.

TABLE II. THE TRAINING RESULT FOR FEATURE EXTRACTION STAGE

Epoch	**Transfer Learning Model**					
	DenseNet-169			**DenseNet-201**		
	RGB	**HSV**	**YCbCr**	**RGB**	**HSV**	**YCbCr**
1	0.8438	0.6484	0.6562	0.8672	0.6484	0.6406
2	0.9766	0.8750	0.8906	0.9531	0.8984	0.8047
3	0.9844	0.9453	0.9531	0.9766	0.8984	0.8750
4	0.9844	0.9531	0.9375	0.9844	0.9375	0.9219
5	1.0000	0.9688	0.9609	0.9922	0.9375	0.9453
6	1.0000	0.9844	0.9609	1.0000	0.9688	0.9531
7	1.0000	-	0.9766	1.0000	0.9844	0.9766
8	1.0000	-	0.9844	1.0000	0.9922	0.9766

Based on the experimental results in Table II, the best results are obtained using RGB colors. It is because the pre-trained model uses images with RGB colors in the transfer learning model. So, when tested using other colors such as HSV and YCbCr, the accuracy results obtained have not reached 100% in epoch 8. This training process uses an early stop with a maximum of no change of 5 epochs. In this experiment, all models stopped at the eighth epoch. Therefore, this research used RGB color as the color configuration in the video data experiment. From the DenseNet-169 and DenseNet-201 models, the best results are obtained using the DenseNet-169 model because in the fourth epoch, the accuracy is 100%, and the DenseNet-169 model is lighter, which affects faster data processing. Therefore, in the experiment using video data, this research used the DenseNet-169 model.

C. Matching with Video Data

The segmentation and feature extraction models were obtained for video data testing. In this test, the data used is a fire video obtained from the VisiFire fire detection software [23]. All video datasets have a resolution of $400 \; x \; 256$ at 15 fps. The number of video frames varies, Controlled1 260 frames video, Controlled2 246 frames, Controlled3 208 frames, Forest1 200 frames, Forest2 245 frames, and Forest3 255 frames. It will check whether a fire object is detected in the video frame. It will evaluate true positive (TP), and false negative (FN) results in each video experiment. Table III shows the results of the evaluation of video data processing.

TABLE III. RESULT OF MATCHING WITH VIDEO DATA

Video	**Proposed Method**		**Color + SVM** [24]		**Tempo-spatial + SVM** [25]	
	TP	**FN**	**TP**	**FN**	**TP**	**FN**
Controlled1	100	0	55.2	44.8	94.98	5.02
Controlled2	100	0	77.7	22.3	-	-
Controlled3	100	0	97.9	2.1	95	5
Forest1	100	0	-	-	-	-
Forest2	100	0	-	-	-	-
Forest3	92.17	7.83	-	-	-	-
Average	98.69	1.305	76.93	23.067	94.99	5.01

The experimental results show that the combination of segmentation and feature extraction models produces a reasonably good true positive, 98.69%. In previous studies, 95% true positive results did not exist in the model using supervised learning. Likewise, with false negative results, in this study, the value was below 2 percent, which means that only a few fire objects were not detected. Previous research also used the handcrafted method, which means that the features obtained are based on the components contained in the fire object. The classification process is also carried out using machine learning. Quantitatively, the average true positive of the proposed method is better than the previous research. The amount of video data tested is also more, so this method passes more test data with various fire object conditions. In this case, it makes qualitative testing of the proposed method better. In addition to testing true positive and false negative values, we also evaluate the resulting fps results for video data processing. Table IV shows the fps results obtained from the tested videos.

TABLE IV. RESULT OF COMPUTATION TIME

Video	**Fps**
Controlled1	16.63
Controlled2	14.75
Controlled3	11.42
Forest1	14.84
Forest2	12.6
Forest3	16.36
Average	14.43

The video used to test the fps is 400x 256 resolutions. The fps results obtained based on Table IV are not the same because the fire objects detected in the video are different. The more fire objects there are in a frame, the fps result also decreases. The average fps produced is quite good, namely 14.43 fps, meaning that for 1 second, it can process around 14 frames. An example of the results of the segmentation and detection processes in this system is shown in Fig. 6. This study has limitations related to the resulting fps that are not optimal. There are still about 12 fps in testing, while CCTV cameras usually produce 15 fps recordings. The challenge in further research is to increase the resulting fps value so that the use of CCTV cameras with high fps can be applied. However, video data processing must also consider the detection results in addition to the resulting fps value. In this study, the number of true positives produced was quite good, 98.69%.

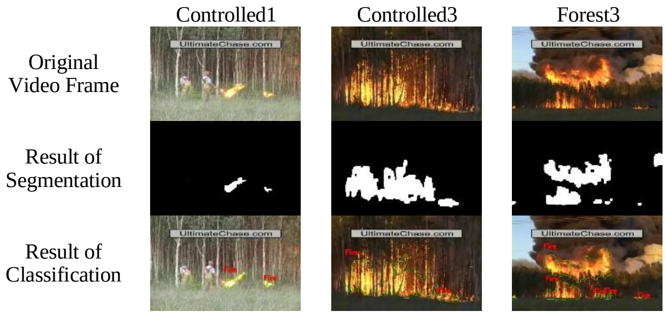


Fig. 6. Example Fire Detection on Video Frame.

V. Conclusion

Fire is a disaster that must be handled immediately so that it does not spread to a broader area. Early hotspot detection is needed, so a fire is directly identified to extinguish the fire. This research proposes a framework for fire detection using video data. The detection process starts with the fire object candidate segmentation. The fire object candidate area was performed by feature extraction and classification using the DenseNet model. It matches results using video data, resulting in true positive values of 98.69% and 14.43 fps. Future research can modify the combination of segmentation and feature extraction methods to produce higher fps. It is because with the development of technology, of course, CCTV cameras will also produce clearer videos with more fps. Therefore, future research is still very open to improving the resulting fps for real-time processing. In addition, if the method produces a high enough fps, it can be applied for implementation with configurations through embedded system devices with CCTV cameras.

Acknowledgment

We would like to thank the Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Indonesia through the Penelitian Dosen Pemula (PDP) grant at the number of IT Tel3701/LPPM-000/Ka.LPPM/VI/2022.

References

[1] R. B. Edwards, R. L. Naylor, M. M. Higgins, and W. P. Falcon, "Causes of Indonesia's forest fires," World Dev., vol. 127, 2020, doi: 10.1016/j.worlddev.2019.104717.

[2] A. Abdusalomov, N. Baratov, A. Kutlimuratov, and T. K. Whangbo, "An improvement of the fire detection and classification method using YOLOv3 for surveillance systems," Sensors, vol. 21, no. 19, 2021, doi: 10.3390/s21196519.

[3] R. L. Naylor, M. M. Higgins, R. B. Edwards, and W. P. Falcon, "Decentralization and the environment: Assessing smallholder oil palm development in Indonesia," Ambio, vol. 48, no. 10, pp. 1195–1208, 2019, doi: 10.1007/s13280-018-1135-7.

[4] A. A. Fitriany, P. J. Flatau, K. Khoirunurrofik, and N. F. Riama, "Assessment on the use of meteorological and social media information for forest fire detection and prediction in riau, indonesia," Sustain., vol. 13, no. 20, 2021, doi: 10.3390/su132011188.

[5] F. Gong et al., "A real-time fire detection method from video with multifeature fusion," Comput. Intell. Neurosci., vol. 2019, 2019, doi: 10.1155/2019/1939171.

[6] R. S. Kharisma and A. Setiyansah, "Fire early warning system using fire sensors, microcontroller, and SMS gateway," J. Robot. Control, vol. 2, no. 3, pp. 165–169, 2021, doi: 10.18196/jrc.2372.

[7] A. Nurhopipah and A. Harjoko, "Motion Detection and Face Recognition for CCTV Surveillance System," IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 12, no. 2, p. 107, 2018, doi: 10.22146/ijccs.18198.

[8] R. Sadek et al., "Novel colored flames via chromaticity of essential colors," Def. Technol., vol. 15, no. 2, pp. 210–215, 2019, doi: 10.1016/j.dt.2018.05.002.

[9] S. D. Thepade, J. H. Dewan, D. Pritam, and R. Chaturvedi, "Fire Detection System Using Color and Flickering Behaviour of Fire with Kekre's LUV Color Space," Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2018, pp. 1–6, 2018, doi: 10.1109/ICCUBEA.2018.8697454.

[10] D. K. Kim, "Color detection using Gaussian mixture model," J. Theor. Appl. Inf. Technol., vol. 95, no. 17, pp. 4313–4320, 2017.

[11] Y. Li, J. Zhang, R. He, L. Tian, and H. Wei, "Hybrid DE-EM Algorithm for Gaussian Mixture Model-Based Wireless Channel Multipath Clustering," Int. J. Antennas Propag., vol. 2019, pp. 1–10, 2019, doi: 10.1155/2019/4639612.

[12] F. D. Adhinata, G. F. Fitriana, A. Wijayanto, M. Pajar, and K. Putra, "Corn Disease Classification using Transfer Learning and Convolutional Neural Network," vol. 9, no. 2, pp. 1–7, 2021.

[13] M. K. Bohmrah and H. Kaur, "Classification of Covid-19 patients using efficient fine-tuned deep learning DenseNet model," Glob. Transitions Proc., vol. 2, no. 2, pp. 476–483, 2021, doi: 10.1016/j.gltp.2021.08.003.

[14] F. Khan, Z. Xu, J. Sun, F. M. Khan, A. Ahmed, and Y. Zhao, "Recent Advances in Sensors for Fire Detection," Sensors, vol. 22, no. 9, 2022, doi: 10.3390/s22093310.

[15] R. A. Khan, J. Uddin, S. Corraya, and J.-M. Kim, "Machine vision-based indoor fire detection using static and dynamic features," Int. J. Control Autom., vol. 11, no. 6, 2018, doi: 10.14257/ijca.2018.11.6.09.

[16] S. D. Khan, L. Alarabi, and S. Basalamah, "Deep hybrid network for land cover semantic segmentation in high-spatial resolution satellite images," Inf., vol. 12, no. 6, pp. 1–16, 2021, doi: 10.3390/info12060230.

[17] A. Anton, N. F. Nissa, A. Janiati, N. Cahya, and P. Astuti, "Application of Deep Learning Using Convolutional Neural Network (CNN) Method For Women's Skin Classification," Sci. J. Informatics, vol. 8, no. 1, pp. 144–153, 2021, doi: 10.15294/sji.v8i1.26888.

[18] Y. Gultom, A. M. Arymurthy, and R. J. Masikome, "Batik Classification using Deep Convolutional Network Transfer Learning," J. Ilmu Komput. dan Inf., vol. 11, no. 2, p. 59, 2018, doi: 10.21609/jiki.v11i2.507.

[19] A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "FireNet: A Specialized Lightweight Fire & Smoke Detection Model for Real-Time IoT Applications," 2019, [Online]. Available: http://arxiv.org/abs/1905.11922

[20] A. Ruseckaite, D. Fok, and P. Goos, "Flexible Mixture-Amount Models Using Multivariate Gaussian Processes," J. Bus. Econ. Stat., vol. 38, no. 2, pp. 257–271, Apr. 2020, doi: 10.1080/07350015.2018.1497506.

[21] T. Zhou, X. Ye, H. Lu, X. Zheng, S. Qiu, and Y. Liu, "Dense Convolutional Network and Its Application in Medical Image Analysis," Biomed Res. Int., vol. 2022, p. 2384830, 2022, doi: 10.1155/2022/2384830.

[22] [22] I. Kousis, I. Perikos, I. Hatzilygeroudis, and M. Virvou, "Deep Learning Methods for Accurate Skin Cancer Recognition and Mobile Application," Electron., vol. 11, no. 9, pp. 1–19, 2022, doi: 10.3390/electronics11091294.

[23] A. E. Cetin, "Computer Vision Based Fire Detection Software," VisiFire, 2014. http://signal.ee.bilkent.edu.tr/VisiFire.

[24] B. C. Ko, K.-H. Cheong, and J.-Y. Nam, "Fire detection based on vision sensor and support vector machines," Fire Saf. J., vol. 44, no. 3, pp. 322–329, 2009, doi: https://doi.org/10.1016/j.firesaf.2008.07.006.

[25] T. Xuan Truong and J.-M. Kim, "Fire flame detection in video sequences using multi-stage pattern recognition techniques," Eng. Appl. Artif. Intell., vol. 25, no. 7, pp. 1365–1372, 2012, doi: https://doi.org/10.1016/j.engappai.2012.05.007.