International Journal of Electrical and Computer Engineering (IJECE)
Vol. 12, No. 3, June 2022, pp. 3118~3128
ISSN: 2088-8708, DOI: 10.11591/ijece.v12i3.pp3118-3128

Comparison of two deep learning methods for detecting fire hotspots

Dewi Putrie Lestari, Rifki Kosasih
Centre for Computational Mathematics Studies, Department of Informatics, Gunadarma University, Depok, Indonesia

Article Info

Article history:

Received Jun 9, 2021
Revised Nov 6, 2021
Accepted Dec 2, 2021

Keywords:

CCTV videos
Deep learning
Faster R-CNN method
Fire hotspots
YOLO method

ABSTRACT

Every high-rise building must meet construction requirements, i.e. it must have good safety to prevent unexpected events such as fire incident. To avoid the occurrence of a bigger fire, surveillance using closed circuit television (CCTV) videos is necessary. However, it is impossible for security forces to monitor for a full day. One of the methods that can be used to help security forces is deep learning method. In this study, we use two deep learning methods to detect fire hotspots, i.e. you only look once (YOLO) method and faster region-based convolutional neural network (faster R-CNN) method. The first stage, we collected 100 image data (70 training data and 30 test data). The next stage is model training which aims to make the model can recognize fire. Later, we calculate precision, recall, accuracy, and F1 score to measure performance of model. If the F1 score is close to 1, then the balance is optimal. In our experiment results, we found that YOLO has a precision is 100%, recall is 54.54%, accuracy is 66.67%, and F1 score is 0.70583667. While faster R-CNN has a precision is 87.5%, recall is 95.45%, accuracy is 86.67%, and F1 score is 0.913022.

Corresponding Author:

Dewi Putrie Lestari
Centre for Computational Mathematics Studies, Department of Informatics, Gunadarma University
Margonda Raya Road, Pondok Cina, Depok, West Java 16424, Indonesia
Email: dewi_putrie@staff.gunadarma.ac.id

1. INTRODUCTION

An area with a very large population causes building to be built vertically, i.e. high-rise building due to decrease empty land [1]. However, a high-rise building has been problems related to safe evacuate of occupant during emergencies such as fire. Fire is an unexpected disaster and must be handled quickly so that it does not spread. If not handled quickly, the fire will cross from one floor to another, making the evacuation process more difficult. Therefore, a fire disaster has become a very serious problem that must be handled quickly and in a timely manner to avoid loss of life and loss of property [2]. In the construction of a high-rise building, each building must meet technical requirements regarding the readiness of a building in the face of a fire disaster, be it infrastructure or facilities. One example that must be prepared is a fire detection system such as a sensor that can be used as fire protection which can provide an early warning of a fire in the building, so that the fire to be resolved quickly. However, this system has a weakness, such as when the fire gets bigger it will damage the sensors installed in the building [3]. Currently, several studies have been developed fire detection system using computer vision to overcome the weakness of the fire alarm sensor. Technology of computer vision can be used to monitor fires remotely using closed circuit television (CCTV) videos. However, it is impossible for security personnel to monitor CCTV videos for a full day. Therefore, in this study, we use artificial intelligence deep learning to find out if there are fire hotspots recorded on CCTV videos.

Deep learning is a subset of machine learning that has a concept similar to how the human brain works, therefore it is also called an artificial neural network [4]. Currently, deep learning is widely used for research, i.e. making decisions, speech recognition, and object detection. One of the methods used for fire hotspots detection is the you only look once (YOLO) method which is a modification of the convolutional neural network (CNN). This method uses a single neural network to analyze objects in the frame. YOLO uses a single neural network for localization of an object in the frame and classification [5]. The network contained in this method is 24 convolutional layers [6]–[8]. Previous research was conducted by Lestari *et al.* [7] to detect fire hotspots using the YOLO method which 45 data on fire object images were used divided by 30 training dataset and 15 testing dataset. Based on the results of the study, an accuracy rate of 90% was obtained. However, the image data used is not much and still uses central processing unit (CPU). Therefore, in our study the data used is reproduced and made more diverse and graphics processing unit (GPU) used. In this study, we also compare the YOLO method with the faster region-based convolutional neural network (faster R-CNN) method. Comparison of the two methods is done by evaluating the performance of the method, i.e. measuring the level of precision, recall, accuracy, and F1 score. Faster R-CNN method is a development of the fast region convolutional neural network (fast R-CNN) [9]. This method has an architecture consisting of 2 parts. The first part, region proposal network is used to decide the location to reduce computation from the whole inference process so that it can scan quickly and efficiently at each location [10]. The second part is Fast R-CNN which is used to sort proposals. Faster R-CNN has 9 anchors consisting of 3 scales and 3 ratios that make this method can detect objects more accurately [11]–[13]. When we use R-CNN, the bounding boxes (BBs) are generated [14].

Several studies have been performed in fire detection, i.e. detect smoke using synthetic smoke images. In this study, a synthesis pipe is built and simulates using a variety of smoke conditions. The data used are categorized into two, i.e. smoke and not smoke. In the test, not smoke category has a strong interference in detecting smoke [15]. Other research was also conducted by Appana *et al.* [16] to detect a smoke on video using the pattern of smoke flow in the alarm system. In this study, he used three attributes in building a smoke detection system, i.e. color, blur, and diffusion behavior. The first stage is analyCze color, then extract the features using the Gabor filtering method to get a feature vector. The final stage of this research is to classify the types of smoke by using a support vector machine (SVM) [16]. Further research was conducted by Hendri [17] on forest fire detection using the CNN method. This method uses reclassification to detect hotspots. To detect an object, the previous system would take an object's classifier and evaluate it at various locations and various scales in the frame. In his research, detection of fire object using CNN method has an accuracy about 54%. The next study was carried out by Mohammed *et al.* [18] to detect forest fires using machine learning methods such as SVM and k-nearest neighbors (KNN) on geodata. In this research, it was obtained accuracy rate of the SVM model was 74% and the KNN was 58%. Furthermore, the research was conducted by Kadir *et al.* [19] used a wireless sensor network (WSN) to detect forest fires. WSN technology is used in sensor systems to collect environmental data. Hotspot detection training data is conducted in the data center to determine and infer fire hotspots that have the potential to become major fire hotspots. However, if a large fire occurs it can damage the sensor device. Based on this description, previous researchers detected fire hotspots using conventional methods, i.e. SVM, KNN, and CNN. Therefore, in this study, fire hotspots were detected using the latest methods such as the YOLO method and the faster R-CNN method.

Other studies have also been performed in detecting fire, i.e. Li and Zhao [20] used the SSD method, Gagliardi *et al.* [21] used the Kalman filter and CNN algorithm, Saponara *et al.* [22] used the YOLOv2 method, Park and Ko [23] used the YOLOv3 method, and Zhong *et al.* [24] used the CNN method. However, previous researchers only perform detection of a fire outdoors i.e. surrounding environment and forest fires and have not detected indoors such as in buildings. So in this study, we propose to detect fires hotspots that appear in the room using the YOLO method and the faster R-CNN method.

2. RESEARCH METHOD

In this study, we want to compare two methods of detecting fire hotspots by using YOLO method and the faster R-CNN method. General framework of this research is shown in Figure 1. The first stage in this research is collecting data. The data used are image data containing 100 random images of fire objects. The data is categorized into two with a composition of 70 training data (70%) and 30 testing data (30%). After obtain the training data and testing data, we perform the labeling image on training data.

2.1. Labeling image

The next stage in this research is to label the training dataset by creating a bounding box around the object to be recognized. The labeling results contain information on the position of the object you want to detect and store in .xml form which shown in Figure 2. After that we perform transfer learning by using the YOLO method and the faster R-CNN method.

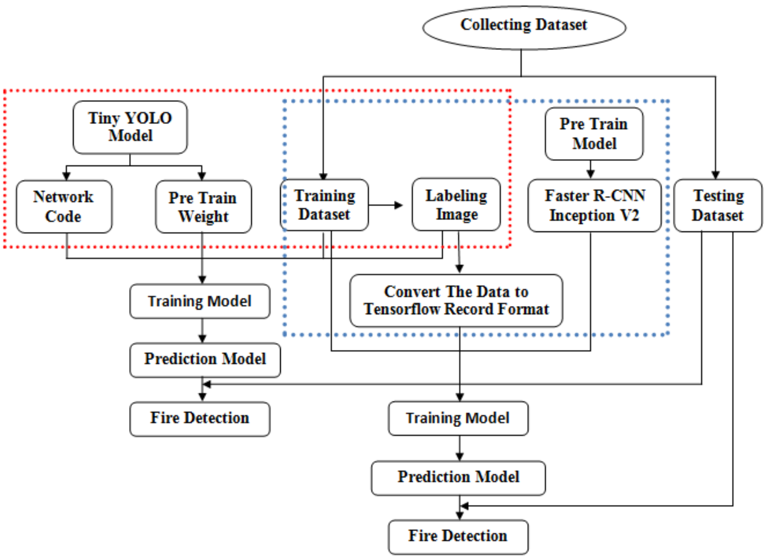


Figure 1. General framework of this research


```
<annotation>
	<folder>images</folder>
	<filename>img6.jpg</filename>
	<path>D:\yolo_code dan manual book\yolo_code dan manual book\darkflow-master\train\images\img6.jpg</path>
	<source>
		<database>Unknown</database>
	</source>
	<size>
		<width>640</width>
		<height>480</height>
		<depth>3</depth>
	</size>
	<segmented>0</segmented>
	<object>
		<name>Fire</name>
		<pose>Unspecified</pose>
		<truncated>0</truncated>
		<difficult>0</difficult>
		<bndbox>
			<xmin>466</xmin>
			<ymin>300</ymin>
			<xmax>493</xmax>
			<ymax>315</ymax>
		</bndbox>
	</object>
</annotation>
```


Figure 2. Example of labeling format in XML

2.2. Transfer learning with YOLO method

In the YOLO method to detect objects, the image will be split into a grid with a size of S×S [25]. The next stage, we make bounding boxes on these grids and have a confidence value. Confidence value is the probability that the object is in the bounding box as in the (1). If the centroid of the fire object is in the grid cell, the grid is tasked with detecting the fire object.

$$\mathrm{CV} = \Pr(\mathrm{Object}) * \mathrm{IOU}_{\mathrm{predict}}^{\mathrm{truth}} \tag{1}$$

IOU is intersection of bounding box predicted by the ground truth divided by the union of bounding box predicted by the ground truth. IOU has value from 0 to 1 and bounding box will approach ground truth if IOU value close to 1 [26]–[28]. We also define probability of class for each grid in (2):

$$Pr(Class_i|Object) * Pr(Object) * \ IOU_{predict}^{truth} = Pr(Class_i) * IOU_{predict}^{truth} \tag{2}$$

In the YOLO method, there are 24 convolution layers with 2 connected layers [29] and has a fast version designed to quickly find the boundary of detected objects [6]. One example of a fast version of the YOLO method is the tiny YOLO model which has 9 convolutional layers [30] is shown in Table 1. The tiny YOLO model contains a network code and pre train weight.

We use the tiny YOLOv3 model as a pre train model to be used in the transfer learning process. Transfer learning is learning carried out by the pre train model to recognize fire objects in training data that have been labeled as in Figure 1 (red box). To make transfer learning, batch size and learning rate are needed. We use batch size=1, because the data used is an image that has a very large size, so that the image sample can pass the training process into the neural network quickly and we use a small learning rate, i.e. 0.0002. The smaller the value of the learning rate, the value of the loss function is guaranteed to decrease after the update. Furthermore, the model training is carried out repeatedly so that the pre train model can recognize the fire object well. In this study, we use loss function to measure performance of model as shown in (3) [7]. The model's performance gets better if the loss value is less than 1 or close to 0.

$$\mathrm{Loss} = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{D} I_{ij}^{obj} [(r_i - \hat{r}_i)^2 + (s_i - \hat{s}_i)^2] + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{D} I_{ij}^{obj} \left[\left(\sqrt{t_i} - \sqrt{\hat{t}_i}\right)^2 + \left(\sqrt{v_i} - \sqrt{\hat{v}_i}\right)^2 \right] + \sum_{i=0}^{s^2} \sum_{j=0}^{D} I_{ij}^{obj} \left(CV_i - \widehat{CV}_i\right)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{D} I_{ij}^{obj} \left(CV_i - \widehat{CV}_i\right)^2 + \lambda_{coord} \sum_{i=0}^{s^2} I_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \quad (3)$$

Where S is the measure of the grid, r and s variables are the centers of each prediction, t and v variables are dimensions of bounding box. The λ_{coord} variable is used to increase probability value of bounding box that has a fire object and λ_{noobj} variable is used to decrease probability value of bounding box that has no fire object. CV is a confidence value and $p_i(c)$ is prediction of class.

The loss value is used to see the performance of the pre train model (tiny YOLO model) in learning to recognize fire objects. After the learning process is complete, a new model from the training results will be formed that can recognize object of fire. The new model will be used to predict an image whether it contains fire objects or not. In this research, we use python programming to run the YOLO method.

Table 1. The architecture of tiny YOLO model

Layer	Shape	Stride	Kernel
Input	(416, 416, 3)		
Conv	(416, 416, 16)	1	3 × 3
MaxPool	(208, 208, 16)	2	2 × 2
Conv	(208, 208, 32)	1	3 × 3
MaxPool	(104, 104, 32)	2	2 × 2
Conv	(104, 104, 64)	1	3 × 3
MaxPool	(52, 52, 64)	2	2 × 2
Conv	(52, 52, 128)	1	3 × 3
MaxPool	(26, 26, 128)	2	2 × 2
Conv	(26, 26, 256)	1	3 × 3
MaxPool	(13, 13, 256)	2	2 × 2
Conv	(13, 13, 512)	1	3 × 3
MaxPool	(13, 13, 512)	1	2 × 2
Conv	(13, 13, 1024)	1	3 × 3
Conv	(13, 13, 1024)	1	3 × 3
Conv	(13, 13, 125)	1	1 × 1

2.3. Transfer learning with faster R-CNN method

The faster R-CNN method uses the region proposal network (RPN) to increase speed when perform objects recognition [31]–[34]. The RPN will receive input in the form of a feature map that has been processed by convolution. The convolution process is carried out using an architecture that is on CNN. In this research, we used inception V2 architecture. The inception V2 architecture is designed to reduce CNN complexity [35]. The inception V2 uses pre train model to transfer learning process which can be seen in Figure 1 (blue box).

To make transfer learning, batch size and learning rate are needed. We use batch size and learning rate same as size and learning rate in the YOLO method, i.e. batch size is 1 and learning rate is 0.0002. The batch size is a term used in transfer learning. The learning rate is the number of changing to the model during each step of this search process [35]. The learning rate can control a model learn a fire detection [36]. After that, we use the loss function to determine performance of the model as in (4) [37]:

$$L(\{pr_i\},\{tr_i\}) = \frac{1}{N_{clas}}\sum_i L_{clas}(pr_i, pr_i^*) + \gamma\frac{1}{N_{regr}}\sum_i pr_i^* L_{regr}(tr_i, tr_i^*) \quad (4)$$

where i is the index of anchor, pr_i is probability value of anchor, pr_i^* is label of ground truth, i.e. if the positive label then pr_i^*=1 and if the negative label then pr_i^*=0, pr_i^* is a coordinate of the bounding box of the anchor, tr_i^* is the ground truth box, L_{clas} is log loss, N_{clas} is normalization classifier value with value 256, and N_{regr} is normalization regression value with value 2,400. However, to balance regression and classifier can be done by multiplying γ [37].

If the loss value less than 1 or close to 0, then transfer learning process will finish [7]. This process produces a model that can recognize fire hotspots. Based on the description of the YOLO method and the faster R-CNN method, the loss value can be used to obtain a good model for detecting fire hotspots. The final stage, we use the new model to predict testing data.

3. RESULTS AND DISCUSSION

This section describes about results of the YOLO method and the faster R-CNN method in detecting fire hotspots. The YOLO method divides image input into grids of S×S size. The pieces of the image will go through a convolution process. In the YOLO architecture, there are 24 convolutions, 4 max pooling, and 2 fully connected layers to get a grid which contain a value that will be used in the classification process. If the number of image grids is very large and the convolution process takes a long time, it will cause a very heavy computational process.

Meanwhile, the faster R-CNN method uses the RPN to propose areas (parts of an image that you want to observe or predict as objects to be detected). The RPN produces several bounding boxes, each box has 2 probability scores whether there are objects at that location or not. The resulting areas will be input in the classification process. The use of the RPN can reduce the computational requirements significantly, because it does not have to go through the process of dividing the image into grids. In this section, we will explain about training dataset, test results on testing dataset using the YOLO method and the faster R-CNN method, and evaluation results using some indicators such as precision, recall, accuracy, and F1 score.

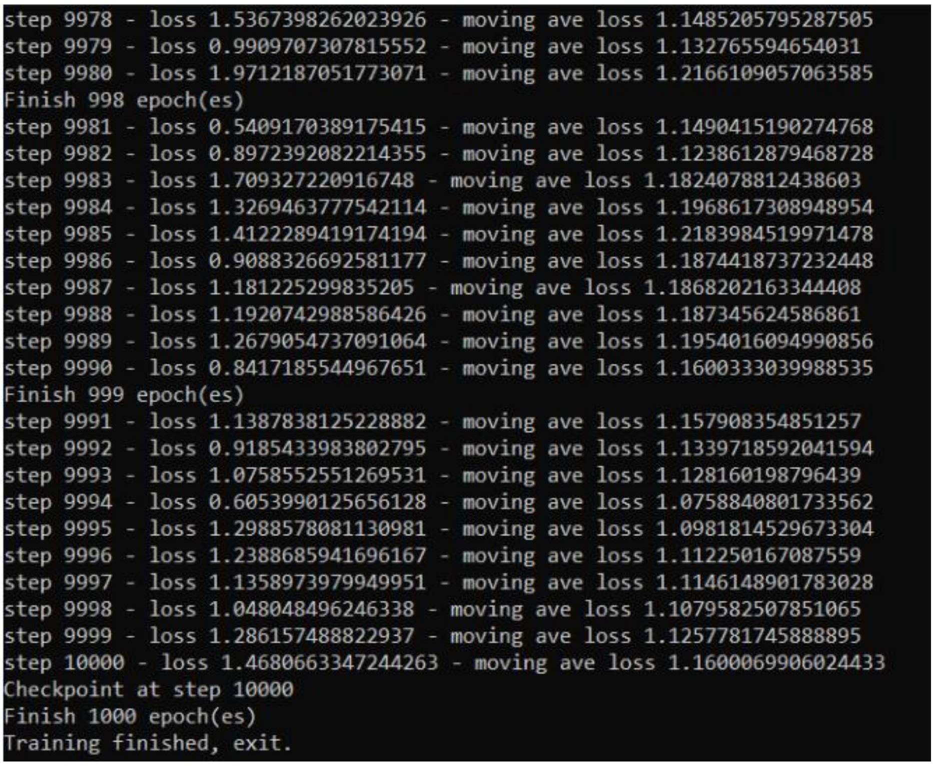
3.1. Training dataset

In this study, we use 70 training dataset obtained from various sites. The training dataset is a collection of images containing fire objects. This data is used by the model to learn about the fire objects that contained in the data. Some of the training dataset that used in this study can be seen in Figure 3.

Figure 3. Sample of training dataset (20 of 70 training dataset)

3.2. Testing results of the YOLO model

After we get the training dataset, the next stage is to label the image by providing a bounding box to the object we want to recognize, i.e. fire. After that, we do transfer learning using a pre train model, i.e. tiny YOLOv3 model, so that the model used can study the fire object. The learning process is done continuously until the loss value is less than 1 or exceeds the desired step limit, which in this study we used 10,000 steps. The loss value of the training model on the YOLO method can be seen in Figure 4.



```
step 9978 - loss 1.5367398262023926 - moving ave loss 1.1485205795287505
step 9979 - loss 0.9909707307815552 - moving ave loss 1.132765594654031
step 9980 - loss 1.9712187051773071 - moving ave loss 1.2166109057063585
Finish 998 epoch(es)
step 9981 - loss 0.5409170389175415 - moving ave loss 1.1490415190274768
step 9982 - loss 0.8972392082214355 - moving ave loss 1.1238612879468728
step 9983 - loss 1.709327220916748 - moving ave loss 1.1824078812438603
step 9984 - loss 1.3269463777542114 - moving ave loss 1.1968617308948954
step 9985 - loss 1.4122289419174194 - moving ave loss 1.2183984519971478
step 9986 - loss 0.9088326692581177 - moving ave loss 1.1874418737232448
step 9987 - loss 1.181225299835205 - moving ave loss 1.1868202163344408
step 9988 - loss 1.1920742988586426 - moving ave loss 1.187345624586861
step 9989 - loss 1.2679054737091064 - moving ave loss 1.1954016094990856
step 9990 - loss 0.8417185544967651 - moving ave loss 1.1600333039988535
Finish 999 epoch(es)
step 9991 - loss 1.1387838125228882 - moving ave loss 1.157908354851257
step 9992 - loss 0.9185433983802795 - moving ave loss 1.1339718592041594
step 9993 - loss 1.0758552551269531 - moving ave loss 1.128160198796439
step 9994 - loss 0.6053990125656128 - moving ave loss 1.0758840801733562
step 9995 - loss 1.2988578081130981 - moving ave loss 1.0981814529673304
step 9996 - loss 1.2388685941696167 - moving ave loss 1.112250167087559
step 9997 - loss 1.1358973979949951 - moving ave loss 1.1146148901783028
step 9998 - loss 1.048048496246338 - moving ave loss 1.1079582507851065
step 9999 - loss 1.286157488822937 - moving ave loss 1.1257781745888895
step 10000 - loss 1.4680663347244263 - moving ave loss 1.1600069906024433
Checkpoint at step 10000
Finish 1000 epoch(es)
Training finished, exit.
```

Figure 4. The loss value of training model on the YOLO method

Based on the Figure 4, it can be seen that the loss value in the 10,000 step is still greater than 1. It indicates that the YOLO model has poor performance. Furthermore, detection of fire hotspots using the testing data is performed. The detection results of fire hotspots using the YOLO method can be seen in Figure 5. In Figure 5, there are 12 images with fire object detected in actual condition and declared as fire in the application (true positive), no fire was detected in actual condition but declared fire in the application (false positive) obtained as many as 0 image, fire detected in actual condition but not stated application (false negative) obtained as many as 10 images, and no fire detected in actual condition and not stated application (true negative) obtained as many as 8 images.

3.3. Testing results of the faster R-CNN model

Next stage, we perform transfer learning using the second method, i.e. the faster R-CNN method. The pre train model used is the faster R-CNN Inception V2. The learning process is done continuously until the loss value is less than 1 or exceeds the desired step limit, which in this study we used 10,000 steps. The loss value of the training model on faster R-CNN method can be seen in Figure 6.

In Figure 6, it can be seen that by using the same number of steps, i.e. 10,000 steps, the loss value close to 0. It indicates that the faster R-CNN model has a very good performance. Furthermore, detection of fire hotspots using the testing data is performed. Detection results of fire hotspots using the faster R-CNN method is shown in Figure 7. In Figure 7, it can be seen that there are 21 images with fire hotspots detected in actual condition and declared as fire in the application (true positive), no fire was detected in actual condition but declared fire in the application (false positive) obtained as many as 3 images, fire was detected in actual condition but not stated application (false negative) obtained as many as 1 image, and no fire detected in actual condition and not stated application (true negative) obtained as many as 5 images.

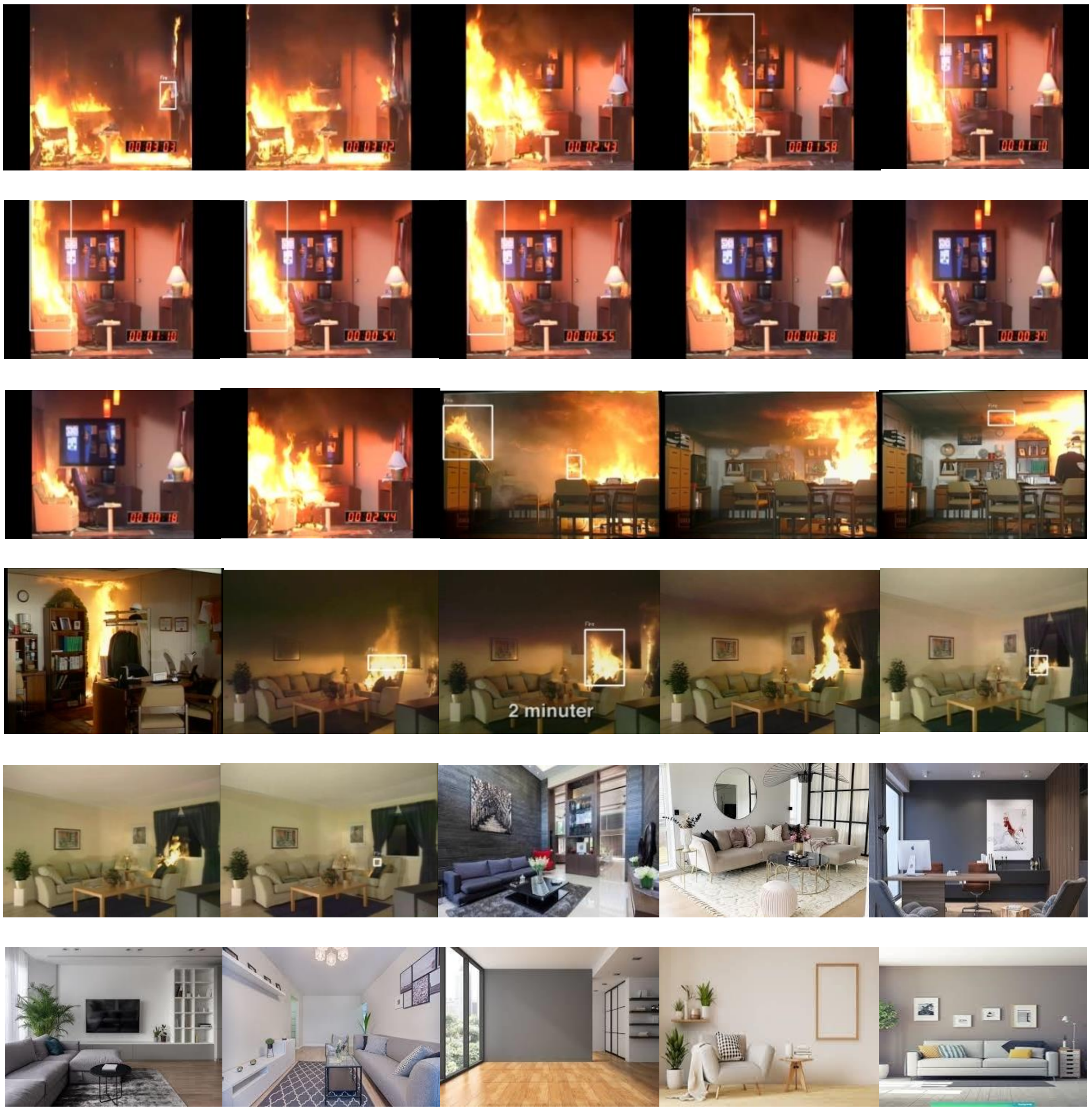


Figure 5. The prediction results of fire objects using the YOLO method

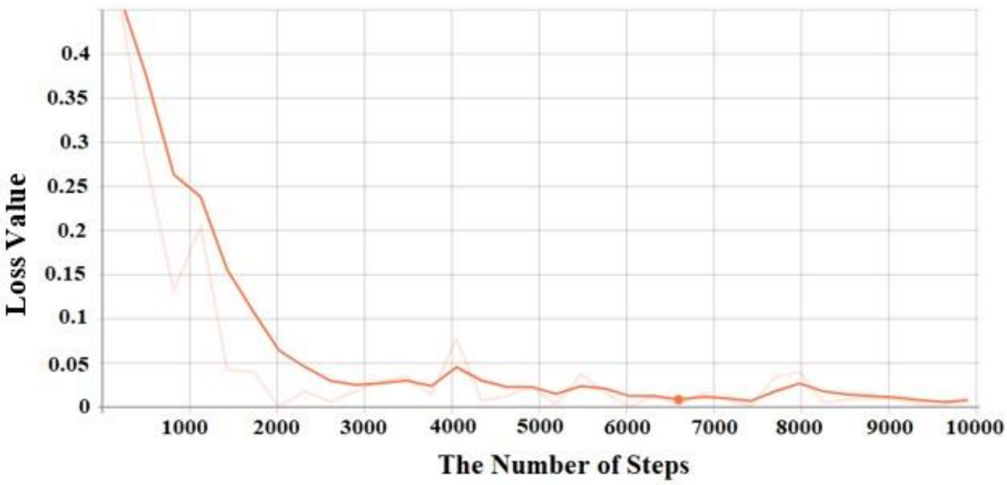


Figure 6. Loss value of training model on the faster R-CNN method

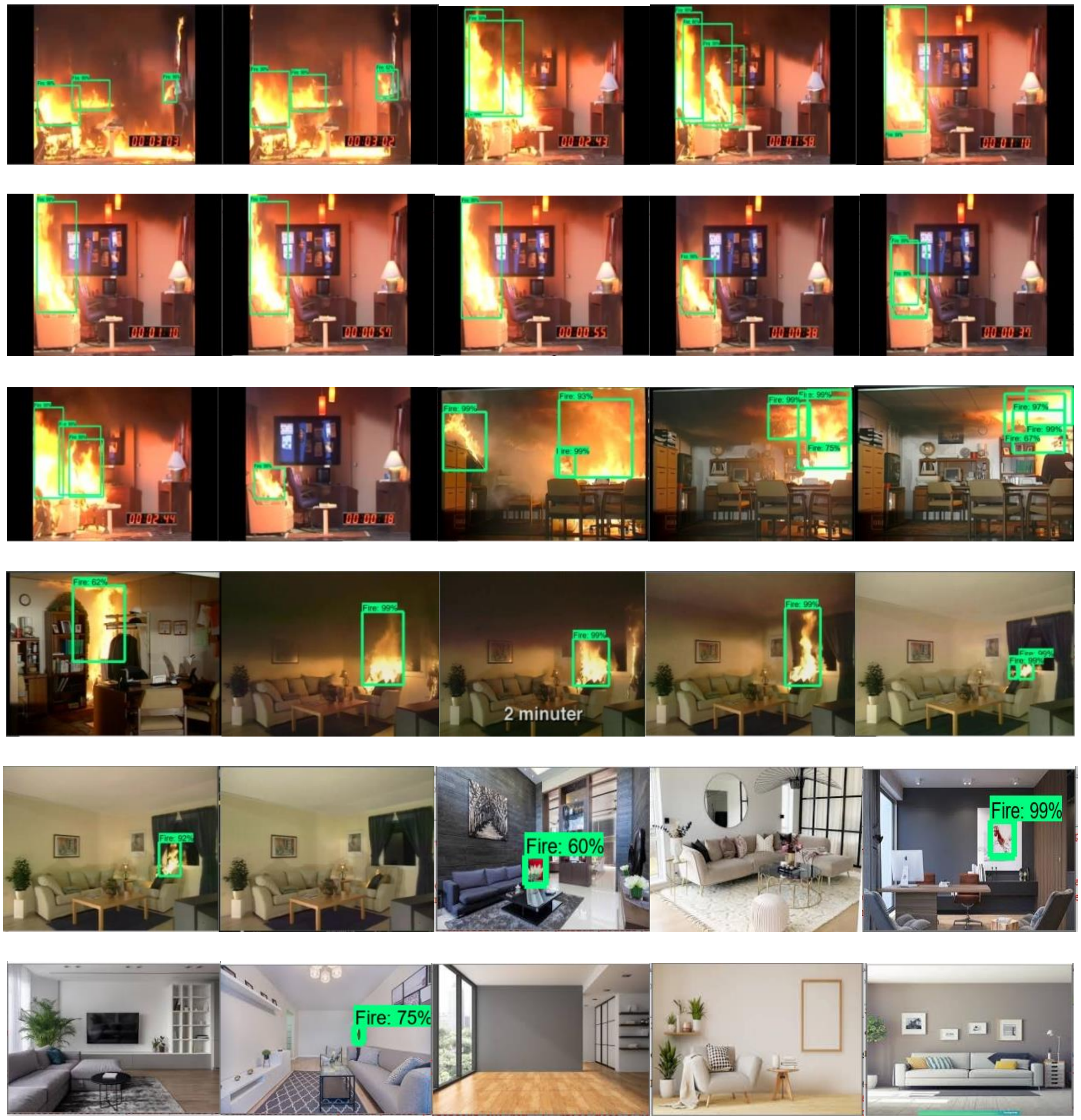


Figure 7. Prediction results of fire hotspots using the faster R-CNN method

3.4. Evaluation results of the model

In this study, we calculate precision, recall, accuracy, and F1 score to measure performance of the YOLO method and the faster R-CNN method. The formula can be seen in (5), (6), (7), and (8) [38]–[40]:

$$\text{Precision} = \frac{\text{TP}}{\text{FP+TP}} \times 100\% \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{FN+TP}} \times 100\% \tag{6}$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{FP+FN+TP+TN}} \times 100\% \tag{7}$$

$$\text{F1 score} = \frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall+ Precision}} \tag{8}$$

with FN is false negative, TN is true negative, TP is true positive, and FP is false positive. The evaluation results of the YOLO method and the faster R-CNN method is shown in Table 2. In Table 2, we can see that the YOLO method has a value of precision is 100%, recall is 54.54%, accuracy is 66.67%, and F1 score is 0.70583667. While the faster R-CNN method has a value of precision is 87.5%, recall is 95.45%, accuracy is

86.67%, and F1 score is 0.913022. We can see that precision value of YOLO better than faster R-CNN method. But recall, accuracy and F1 score of faster R-CNN better than YOLO method.

Table 2. Evaluation results of YOLO and faster R-CNN methods

Indicator	YOLO	Faster R-CNN
True Positive	12	21
False Positive	0	3
False Negative	10	1
True Negative	8	5
Precision	100%	87.5%
Recall	54.54%	95.45%
Accuracy	66.67%	86.67%

YOLO method is very good at detecting the presence of fire hotspots if the image data used is uniform (training and testing image are not much different). However, if the image data used is random (training and testing image are very different), the YOLO method is not good in detecting the presence of fire hotspots. Therefore, if the image data is random, it is suggested to use the faster R-CNN method because it is very good in detecting fire hotspots.

4. CONCLUSION

In high-rise building, fire object detection is needed to determine whether a room has a fire or not so that it can be immediately handled by the fire department. In this study, a comparison of fire detection using 2 methods was carried out, i.e. the YOLO method and the faster R-CNN method. The data used consisted of 100 images containing fire objects. We divide data into 70 training data and 30 testing data. Later, we perform model training so that the model can learn and recognize fire objects. The next stage is to make predictions using testing dataset. From research results, we found the YOLO method has an accuracy rate is 66.67% and the faster R-CNN method has an accuracy rate is 86.67%. This indicates that the faster R-CNN method has better performance than the YOLO method. For further research, trainings with more types of backgrounds are also added.

ACKNOWLEDGEMENTS

This research was funded by Penelitian Terapan (PT) scheme of National Competitive Research Grant 2021, Directorate of Resources, Ministry of Education, Culture, Research, and Technology, Republic of Indonesia to the Gunadarma University No. 3581/LL3/KR/2021.

REFERENCES

[1] G. S. Birajdar, R. Singh, A. Gehlot, and A. K. Thakur, “Development in building fire detection and evacuation system-a comprehensive review,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 6644–6654, Dec. 2020, doi: 10.11591/ijece.v10i6.pp6644-6654.

[2] H. Alqourabah, A. Muneer, and S. M. Fati, “A smart fire detection system using iot technology with automatic water sprinkler,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, pp. 2994–3002, Aug. 2021, doi: 10.11591/ijece.v11i4.pp2994-3002.

[3] S.-J. Chen, D. C. Hovde, K. A. Peterson, and A. W. Marshall, “Fire detection using smoke and gas sensors,” *Fire Safety Journal*, vol. 42, no. 8, pp. 507–515, Nov. 2007, doi: 10.1016/j.firesaf.2007.01.006.

[4] P. Patel and A. Thakkar, “The upsurge of deep learning for computer vision applications,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 538–548, Feb. 2020, doi: 10.11591/ijece.v10i1.pp538-548.

[5] R. Deepa, E. Tamilselvan, E. S. Abrar, and S. Sampath, “Comparison of Yolo, SSD, faster RCNN for real time tennis ball tracking for action decision networks,” in *2019 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Apr. 2019, pp. 1–4, doi: 10.1109/ICACCE46606.2019.9079965.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *roceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Jun. 2016, arxiv.org/abs/1506.02640.

[7] D. P. Lestari, R. Kosasih, T. Handhika, Murni, I. Sari, and A. Fahrurozi, “Fire hotspots detection system on CCTV videos using you only look once (YOLO) method and tiny YOLO model for high buildings evacuation,” in *2019 2nd International Conference of Computer and Informatics Engineering (IC2IE)*, Sep. 2019, pp. 87–92, doi: 10.1109/IC2IE47452.2019.8940842.

[8] S. Shinde, A. Kothari, and V. Gupta, “YOLO based human action recognition and localization,” *Procedia Computer Science*, vol. 133, pp. 831–838, 2018, doi: 10.1016/j.procs.2018.07.112.

[9] M. Lokanath, K. S. Kumar, and E. S. Keerthi, “Accurate object classification and detection by faster-RCNN,” *IOP Conference Series: Materials Science and Engineering*, vol. 263, no. 5, pp. 1–8, Nov. 2017, doi: 10.1088/1757-899X/263/5/052028.

[10] H. S. Sucuoğlu, İ. Böğrekci, and P. Demircioğlu, “Real time fire detection using faster R-Cnn model,” *International Journal of 3d Printing Technologies and Digital Industry*, vol. 3, no. 3, pp. 220–226, 2019.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[12] Y. Kawazoe *et al.*, "Faster R-CNN-based glomerular detection in multistained human whole slide omages," *Journal of Imaging*, vol. 4, no. 7, p. 91, Jul. 2018, doi: 10.3390/jimaging4070091.

[13] J. O. Pinzón Arenas, R. Jiménez, and P. C. Useche Murillo, "Faster R-CNN for object location in a virtual environment for sorting task," *International Journal of Online Engineering (iJOE)*, vol. 14, no. 07, pp. 4–14, Jul. 2018, doi: 10.3991/ijoe.v14i07.8465.

[14] G. Devnani, A. Jaiswal, R. John, R. Chaurasia, and N. Tirpude, "Performance evaluation of fine-tuned faster R-CNN on specific MS COCO objects," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2548–2555, Aug. 2019, doi: 10.11591/ijece.v9i4.pp2548-2555.

[15] G. Xu, Y. Zhang, Q. Zhang, G. Lin, and J. Wang, "Deep domain adaptation based video smoke detection using synthetic smoke images," *Fire Safety Journal*, vol. 93, pp. 53–59, Oct. 2017, doi: 10.1016/j.firesaf.2017.08.004.

[16] D. K. Appana, R. Islam, S. A. Khan, and J.-M. Kim, "A video-based smoke detection using smoke flow pattern and spatial-temporal energy analyses for alarm systems," *Information Sciences*, vol. 418–419, pp. 91–101, Dec. 2017, doi: 10.1016/j.ins.2017.08.001.

[17] M. Hendri, "The design of smoke and fire detection system using image processing," Indonesian Islamic University, 2018.

[18] Z. Mohammed, C. Hanae, and S. Larbi, "Comparative study on machine learning algorithms for early fire forest detection system using geodata.," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 5507–5513, Oct. 2020, doi: 10.11591/ijece.v10i5.pp5507-5513.

[19] E. A. Kadir, H. Irie, S. L. Rosa, and M. Othman, "Modelling of wireless sensor networks for detection land and forest fire hotspot," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 6, pp. 2772–2781, Dec. 2019, doi: 10.12928/telkomnika.v17i6.12971.

[20] P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," *Case Studies in Thermal Engineering*, vol. 19, Jun. 2020, doi: 10.1016/j.csite.2020.100625.

[21] A. Gagliardi, F. de Gioia, and S. Saponara, "A real-time video smoke detection algorithm based on Kalman filter and CNN," *Journal of Real-Time Image Processing*, vol. 18, no. 6, pp. 2085–2095, Dec. 2021, doi: 10.1007/s11554-021-01094-y.

[22] S. Saponara, A. Elhanashi, and A. Gagliardi, "Real-time video fire/smoke detection based on CNN in antifire surveillance systems," *Journal of Real-Time Image Processing*, vol. 18, no. 3, pp. 889–900, Jun. 2021, doi: 10.1007/s11554-020-01044-0.

[23] M. Park and B. C. Ko, "Two-step real-time night-time fire detection in an urban environment using static ELASTIC-YOLOv3 and temporal fire-tube," *Sensors*, vol. 20, no. 8, p. 2202, Apr. 2020, doi: 10.3390/s20082202.

[24] Z. Zhong, M. Wang, Y. Shi, and W. Gao, "A convolutional neural network-based flame detection method in video sequence," *Signal, Image and Video Processing*, vol. 12, no. 8, pp. 1619–1627, Nov. 2018, doi: 10.1007/s11760-018-1319-4.

[25] J. Lu *et al.*, "A vehicle detection method for aerial image based on YOLO," *Journal of Computer and Communications*, vol. 06, no. 11, pp. 98–107, 2018, doi: 10.4236/jcc.2018.611009.

[26] X. Zhao, H. Jia, and Y. Ni, "A novel three-dimensional object detection with the modified you only look once method," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, pp. 1–13, Mar. 2018, doi: 10.1177/1729881418765507.

[27] G. Li, Z. Song, and Q. Fu, "A new method of image detection for small datasets under the framework of YOLO network," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Oct. 2018, pp. 1031–1035, doi: 10.1109/IAEAC.2018.8577214.

[28] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: a metric and a loss for bounding box regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 658–666, doi: 10.1109/CVPR.2019.00075.

[29] H. Handoko, J. H. Pratama, and B. W. Yohanes, "Traffic sign detection optimization using color and shape segmentation as pre-processing system," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 1, pp. 173–181, Feb. 2021, doi: 10.12928/telkomnika.v19i1.16281.

[30] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers," in *2018 IEEE International Conference on Big Data (Big Data)*, Dec. 2018, pp. 2503–2510, doi: 10.1109/BigData.2018.8621865.

[31] B. Kim and J. Lee, "A video-based fire detemtion using deep learning models," *Applied Sciences*, vol. 9, no. 14, p. 2862, Jul. 2019, doi: 10.3390/app9142862.

[32] R. Jimenez-Moreno, A. R. Fonseca, and J. L. Ramirez, "Object gripping algorithm for robotic assistance by means of deep learning," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 6292–6299, Dec. 2020, doi: 10.11591/ijece.v10i6.pp6292-6299.

[33] N. Gilbert and A. Rusli, "Single object detection to support requirements modeling using faster R-CNN," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, pp. 830–838, Apr. 2020, doi: 10.12928/telkomnika.v18i2.14838.

[34] A. M. Alkababji and O. H. Mohammed, "Real time ear recognition using deep learning," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 523–530, Apr. 2021, doi: 10.12928/telkomnika.v19i2.18322.

[35] D. Alamsyah and M. Fachrurrozi, "Faster R-CNN with inception V2 for fingertip detection in homogenous background image," *Journal of Physics: Conference Series*, vol. 1196, pp. 1–6, Mar. 2019, doi: 10.1088/1742-6596/1196/1/012017.

[36] J. T. Hardinata, M. Zarlis, E. B. Nababan, D. Hartama, and R. W. Sembiring, "Modification of learning rate with lvq model improvement in learning backpropagation," *Journal of Physics: Conference Series*, vol. 930, no. 1, pp. 1–5, Dec. 2017, doi: 10.1088/1742-6596/930/1/012025.

[37] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[38] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011, doi: 10.9735/2229-3981.

[39] V. Agarwal and V. A, "Recommender system for surplus stock clearance," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 3813–3821, Oct. 2019, doi: 10.11591/ijece.v9i5.pp3813-3821.

[40] B. Adepu, J. Gyani, and G. Narsimha, "An improved differential evolution algorithm for data stream clustering," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2659–2667, Aug. 2019, doi: 10.11591/ijece.v9i4.pp2659-2667.

BIOGRAPHIES OF AUTHORS

Dewi Putrie Lestari She received the B.S. degree in Mathematics from Indonesia University, Indonesia, in 2008, the M.S. in Mathematics from Indonesia University, Indonesia, in 2012, and the Ph.D. degree from Gunadarma University, Indonesia, in 2015. She has been worked as lecturer in Departement of Informatics, Gunadarma University. Her research interests are medical image processing, fire object detection, credit scoring, and deep learning. Email: dewi_putrie@staff.gunadarma.ac.id.

Rifki Kosasih He received the B.S degree in Mathematics (Indonesia University), Indonesia, in 2009, the M.S degree in Mathematics (Indonesia University) in 2012, and the Ph.D. degree (Gunadarma University), Indonesia, in 2015. He has been worked as lecturer in Department of Informatics (Gunadarma University). His research interests are manifold learning, segmentation, image processing, object recognition, and deep learning. Email: rifki_kosasih@staff.gunadarma.ac.id.