Journal of Physical Chemistry and Functional Materials

Home Page of Journal: https://dergipark.org.tr/jphcfum

Image fire detection module for automatic fire extinguishing system with unmanned ground vehicles

Umut Güzel[a] Gülşah Karaduman* [a]
[a]*Firat University, Computer Engineering Department*
** Corresponding author: E-mail: gkaraduman@firat.edu.tr*

ABSTRACT

Especially in responding to large fires, the use of unmanned vehicles can reduce the risk of people getting hurt or encountering situations where they can get hurt. At the same time, the use of unmanned vehicles can increase the efficiency of the intervention. In this direction, one of the most important modules for the unmanned ground vehicles to be used to achieve the desired results is the fire detection module, which will detect the fire and report it to the necessary systems for intervention. In this study, certain deep learning networks were examined for fire detection. These networks are Faster-RCNN, Mask-RCNN, SSD and YOLO. After these networks were trained with the same data sets, they were compared with FPS and mAP data. Faster RCNN, YOLO and SSD methods were used in the study. The mAP values obtained from these methods are as follows: 0.253, 0.45472, 22.15, respectively. As a result, it was seen that the YOLO algorithm gave a more positive result than other deep learning networks in terms of both detection and output speed. As a result, YOLO was selected and used as the deep learning network to be used for fire detection.

ARTICLE INFO

***Keywords*:**

YOLO
Image Processing
Fire Detection

Received: 2024-06-15
Accepted: 2024-07-20
ISSN: 2651-3080
DOI: 10.54565/jphcfum.1501853

1. Introduction

Fires are already a significant problem for humanity, but this problem will become more significant with the effect of global warming [1]. The effects occurring in the northern forests can be given as an example of these problems. The fires that occur affect the soil, vegetation and the air in the region. As a result, the climate is affected [2]. This change can cause a more challenging and uncontrolled world for humanity. In order to minimize negative situations, it is important to intervene in fires effectively. It is an inevitable necessity to use technology for effective intervention. In its simplest form, water must be delivered to areas where there is no water to fight fires, and this is possible with technology. The development of technology at every step will increase the effectiveness of the intervention. One of these steps is unmanned ground vehicles. Unmanned ground vehicles mean the replacement of invaluable vehicles that we can always replace with a new one. In addition, they offer the opportunity to intervene in more challenging conditions. Increasing the automation of these vehicles, which will increase the efficiency of the intervention, will increase efficiency. One of the important modules for this is the fire detection module.

Pincott et al. [8] suggested using computer vision-based strategies for indoor fire detection. In their work, they considered existing models based on Faster R–CNN Inception V2 and SSD MobileNet V2 models. They used small training and testing datasets consisting of images with varying pixel density.

De Vanencio et al. [10] proposed a CNN-based fire detector system suitable for low-power, resource-constrained devices. Their proposed approach consists of training a deep detection network and then removing its less important convolutional filters to reduce the computational cost while trying to preserve the original performance. The results obtained by examining different pruning techniques show that we can reduce the computational cost up to 83.60% and the memory consumption up to 83.86% without degrading the performance of the system.

Umar et al. [11] presented a comprehensive review of the state-of-the-art smoke and fire detection techniques using image processing. In their proposed work, they first compared smoke detection methods and different types of approaches for smoke classification. Maoult et al. [12] presented a new system based on a low-cost CCD camera to detect fire in the near-infrared spectral band. Diaconu [13] presents the state of the art in the field of fire detection, prevention and spread modeling with machine learning algorithms. In order to understand how an AI application has penetrated the fire detection field, a quantitative scientific analysis was first performed.

Sierra et al. [14] provide a comprehensive review of the developments in machine learning-driven fire detection techniques. They discuss their benefits and challenges and outline potential future directions for research and development. Ghali et al. [15] review previous work on deep learning-based wildfire classification, detection and segmentation, including computer vision converters. Then, they describe the most popular and publicly available datasets used for these tasks. They show that deep learning approaches outperform traditional machine learning methods and can significantly improve the performance in detecting, segmenting and classifying wildfires. Chitram et al. [16] provide a comprehensive review of the current research in this field. In their analysis, they highlight that image-based detection can be more efficient than currently used sensors.

Khan et al. [17] present a multi-attention fire network (MAFire-Net) that integrates a modified ConvNeXtTiny (ConvNeXt-T) architecture with channel attention (CA) and spatial attention (SA) modules. These attention modules are integrated after each block of the ConvNeXt-T architecture, where the CA module is responsible for capturing the dominant channels within the features, which leads to highly highlighted feature maps. The SA module enhances the spatial details, allowing the model to distinguish more accurately between fire and non-fire scenarios.

Deep learning algorithms need to be used to detect fire from images. There are many types of algorithms and approaches, and experimentation is important in choosing them. The limitation of this study is as follows: In this study, Faster-RCNN, Mask-RCNN, SSD and YOLO architectures are proposed for fire detection from images and the results of these architectures are compared.

The organization of the article is as follows: Introduction section where the subject is summarized and literature discussion is made, Material and Method section where the proposed methods are explained, Results section where the results obtained from the methods used are shared and interpreted, and finally Conclusions section where the study is summarized.

2. Material and Method

The flow diagram of the work done in this article is shown in Figure 1.

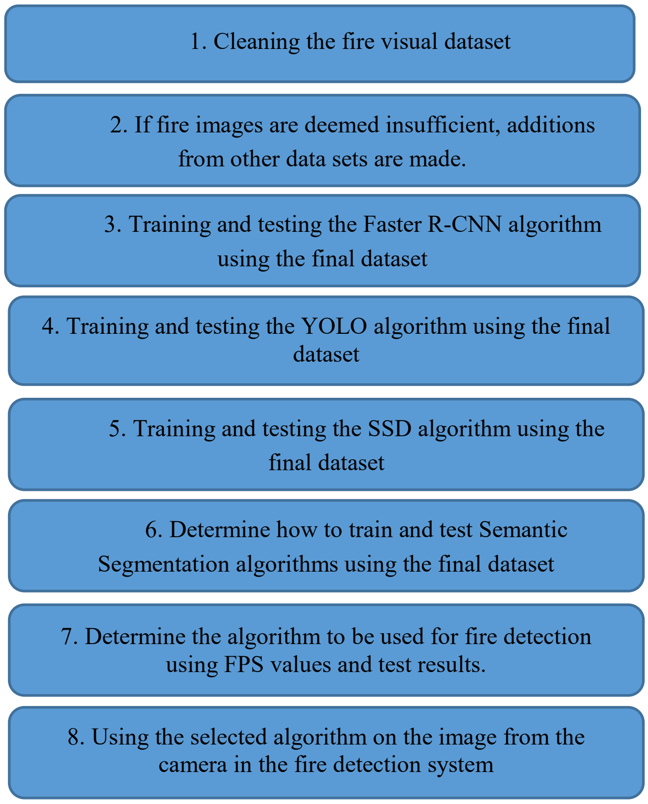

Figure 1. Flow chart of the work done

2.1. Cleaning the fire visual dataset

The necessary images were collected from various sources. Since the collected images were taken from multiple sources, it was normal to encounter duplicate images. Since the dataset consisted of thousands of images, this could not be noticed at the first stage. Duplicate images were eliminated during the labeling stage. Since the semantic segmentation and detection algorithms required different types of labels for labeling the dataset, the labeling process was carried out twice. While a quadrilateral area consisting of 4 points was used for object detection, labeling was done with a polygon structure consisting of at least 3 points for semantic segmentation. During the labeling processes, images that were thought to have insufficient visual quality and that did not contain fire were also deleted. Some of the images were damaged for an unknown reason. A small code was used to get rid of these damaged images and label files. The final labeled and cleaned dataset was 1002 images. Later, these images were increased by changing the values such as saturation, noise, exposure and mirroring. The final dataset consists of 2805 images.

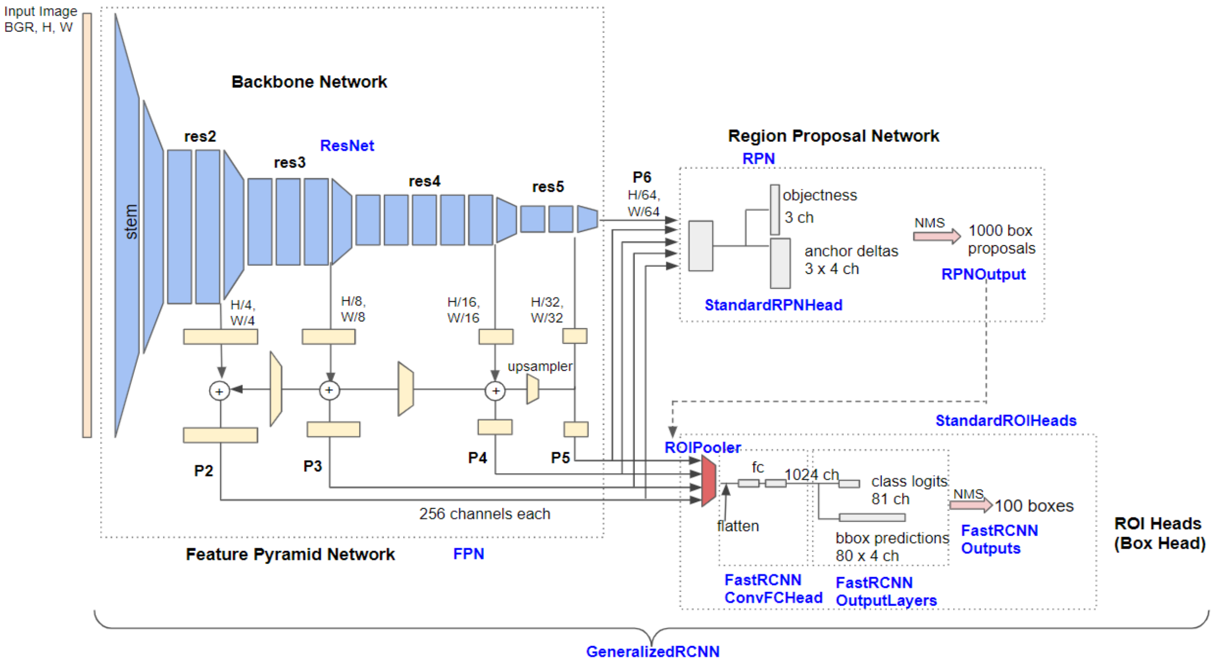

Figure 2. Faster RCNN model architecture

2.2. Training and testing the Faster R-CNN algorithm using the final dataset

Faster R-CNN is a 2-part detection algorithm. The architecture of a sample Faster RCNN model is shown in Figure 2. In the first part, it completes the feature extraction from the image and finding possible object regions. In the second part, classification and region detection are performed. The parts of the architecture in Figure 4 are explained as follows:

Backbone Network: Performs feature extraction from the image.

FPN (Feature Pyramid Network): Transmits the features coming from the Backbone in various sizes and in an enriched form.

RPN (Region Proposal Network): Determines possible objects and regions in the image.

Standard ROI Heads: Finally, it outputs regions classified in appropriate sizes in the incoming regions.

Google colab was used throughout the training. The libraries where the algorithm was located and where the training would be done were downloaded. The relevant dataset was uploaded to colab. Then, the labels of the relevant images were converted to the format requested by the library. The relevant configuration and metadata values were given. The training process was performed. Then, the model was tested and training data was taken.

ResNet was used as the backbone. It is basically an algorithm created to solve the problem of very small or very large numbers occurring during training as the network grows. It allows the creation of very deep networks.[7] Since a backbone with low depth will be positive in terms of FPS performance, ResNet-50 was chosen as the backbone.

2.3. Training and testing the YOLO algorithm using the final dataset

Figure 3 shows the architecture of an example YOLO model.

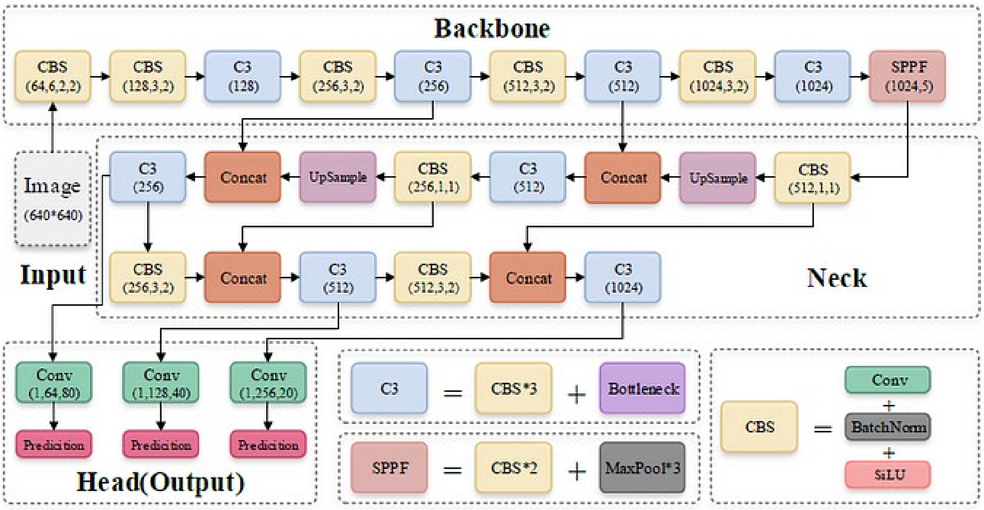

Figure 3. YOLO model architecture

In this architecture, EfficientNet Yolov5 aims to provide better detection by diversifying the features coming from the backbone from the neck section. Finally, the detection and classification process is taken as output from the head section. This structure is a single-step object detection algorithm. Since it performs the entire process in a single structure, it provides faster output compared to the corresponding 2-step models.

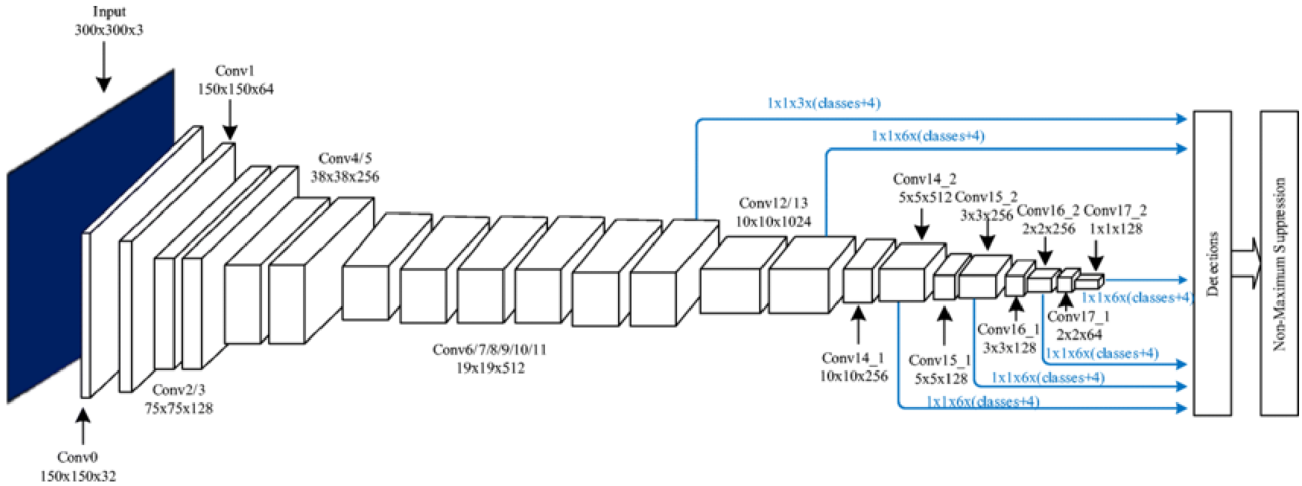

Figure 4. SSD model architecture

2.4. Training and testing the SSD algorithm using the final dataset

Figure 4 shows the architecture of an example SSD model. SSD mobilnet v2 was used as the SSD model. As can be understood from the name mobilnet used as the backbone, it tries to be more efficient in terms of computation. At the same time, it tries to preserve the existing detection success. This is suitable for our purpose. It is an algorithm that detects objects in a single step.

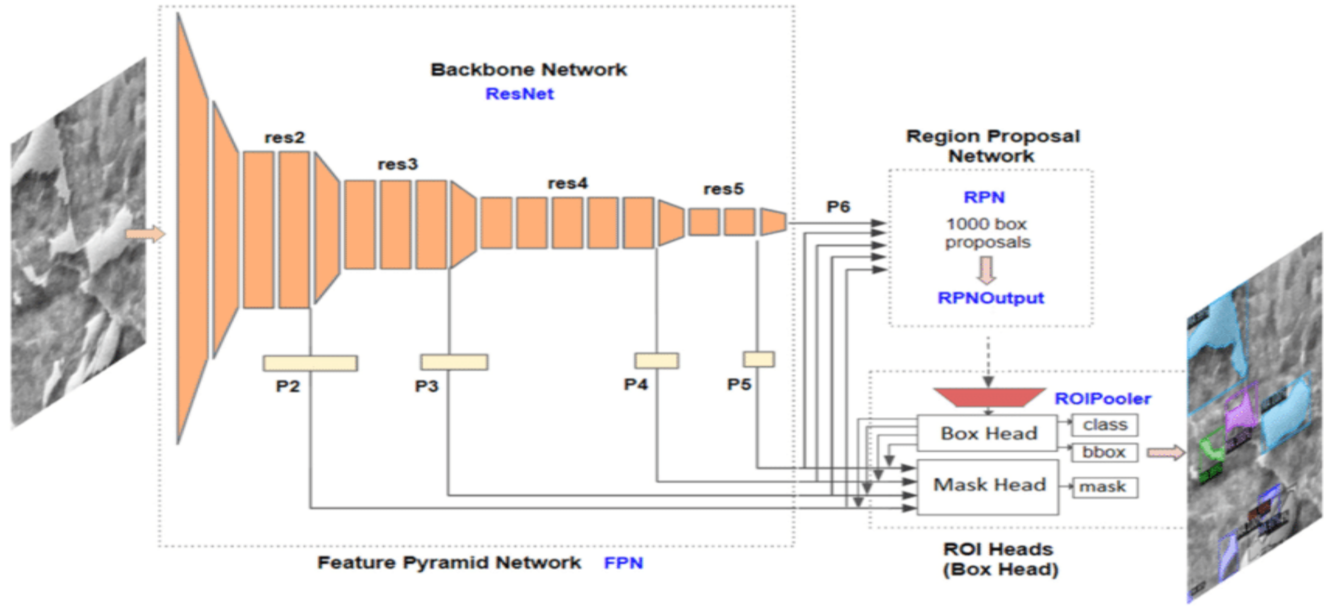

Figure 5. Mask RCNN model architecture

2.5. Training and testing Semantic Segmentation algorithms using the final dataset

The mask-rcnn algorithm was used as segmentation. Figure 5 shows the architecture of a sample Mask RCNN model. In addition to the Faster Rcnn algorithm, a separate head is added for the mask process, and this algorithm performs both object detection and segmentation. This add-on further reduces the already low Faster Rcnn FPS value.

3. Result

The Roboflow labeling method was used to label fire images. Figure 5-a shows an image labeled with the Roboflow rectangle, while Figure 5-b shows an image labeled with the Roboflow multi-point technique.

a) b)

Figure 5. a) Roboflow rectangle labeling, b) Roboflow multi-point labeling

Table 1 shows the metrics obtained as a result of the Faster RCNN algorithm.

Table 1. Metrics of Faster RCNN algorithm's result

$mAP_{0,50}$	$mAP_{0,50\text{-}0,90}$	cls_loss	FPS (GPU T4)
0,253	0.123	0,1	4,29

As can be seen in the table, we cannot say that it is very successful in determining the location of the fire. When we examine the small, medium and large values for the mAP0.50-0.90 value, one of the reasons affecting this is that small objects are almost never detected in the correct position. In order to achieve this, detection can be done with a larger input image instead of 640x640 as the input image. At the same time, detection can be done with anchor boxes of different ratios and sizes by changing the metadata values used for the anchor box. However, both of these situations will negatively affect the real-time detection performance of the model. Therefore, these changes were not applied in order not to further reduce the already low fps value. Figure 6 shows the images of the Faster RCNN test output.

Figure 6. Faster RCNN test output

When Yolov5s is compared to other Yolov5 models, it is a better choice than other options in terms of mAP and ms ratio. It does not have the lowest value in mAP, but in terms of ms value, it is approximately 2 times faster than the next model. Therefore, the Yolov5s model was preferred over other models. Table 2 shows the success metrics of the YOLO algorithm.

Table 2. Metrics of YOLO's result

$\mathbf{mAP_{0,50}}$	$\mathbf{mAP_{0,50\text{-}0,90}}$	**cls_loss**	**FPS (GPU T4)**
0.45472	0.2321	0.021823	11,77

Figure 7 shows images where the YOLO algorithm was applied and fire was detected.

Figure 7. YOLO test output

The SSD algorithm provides a result very close to the YOLO algorithm in terms of FPS. However, YOLO still provides more output. In terms of detection performance, it provides a result closer to the Faster-RCNN algorithm. Table 3 shows the success metrics of the SSD algorithm.

Table 3. Metrics of SSD's output

$\mathbf{mAP_{0,50}}$	$\mathbf{mAP_{0,50\text{-}0,90}}$	**cls_loss**	**FPS (GPU T4)**
22.15	9,66	0.16	9.59

Figure 8 shows images where the SSD algorithm is applied and fire detection is performed.

Figure 8. SSD test output

Similar metadata was tried to be selected in this architecture as in the Faster Rcnn architecture, for example, ResNet-50 backbone was used in this architecture. Table 4 shows the success metrics of the Mask RCNN algorithm.

Table 4. Metrics of Mask RCNN's result

Bounding Box	
$\mathbf{mAP_{0,50}}$	$\mathbf{mAP_{0,50\text{-}0,90}}$
0.415	0.247
Segmentation	
$\mathbf{mAP_{0,50}}$	$\mathbf{mAP_{0,50\text{-}0,90}}$
0.421	0.227

FPS (GPU T4)
3,77
cls_loss
0,065

Figure 9 shows images in which fire detection was performed by applying the Mask RCNN algorithm.

Figure 9. Mask RCNN test output

3.1. Determine the algorithm to be used for fire detection using FPS values and test results

According to the obtained data, using the YOLO algorithm will give better results in terms of both the number of outputs and detection accuracy. Figure 10 shows the FPS values of the methods used.

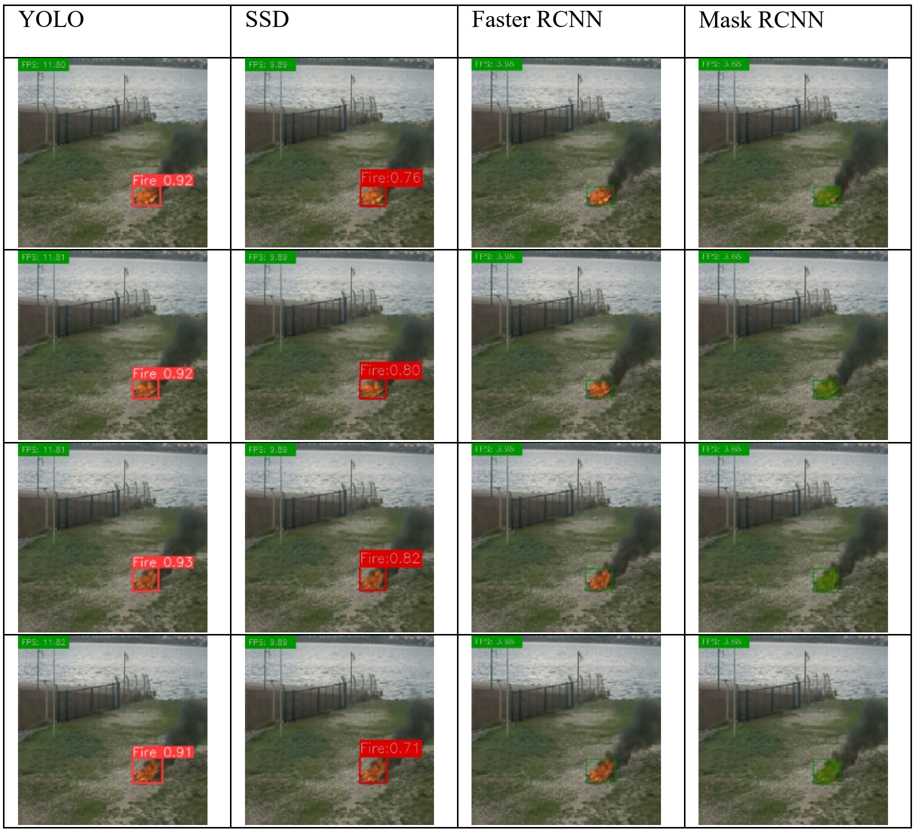

Figure 10. Outputs of the models

3.2. Using the selected algorithm on the image from the camera in the fire detection system

Some images where fire detection was performed using the final model are shown in Figure 11.

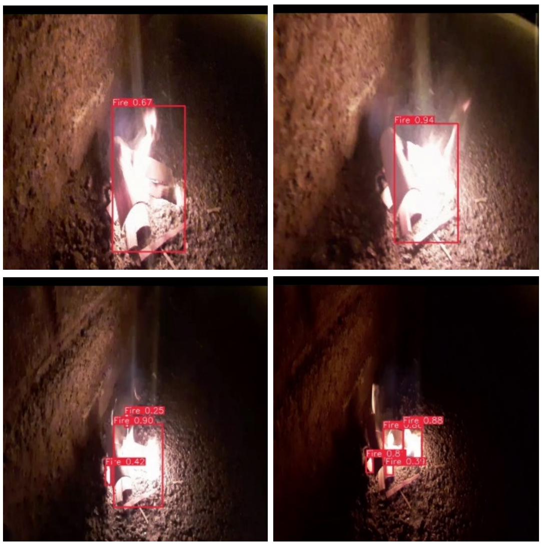

Figure 11. Outputs of the final YOLO model

Each deep learning network has its own advantages and disadvantages. In our evaluation phase, our purpose of use and location play an important role in our deep learning network selection. In terms of usage, it was observed that one-stage deep learning networks were more efficient in terms of FPS when it was desired to be used in an area where both mobile and computational constraints were present. Again, it was observed that YOLOv5s algorithm was more efficient among YOLOv5s and SSD algorithms. Therefore, it was observed that YOLOv5s algorithm would be a more suitable algorithm for fire detection. As a result of the study, we trained four different models and then compared these models to obtain the most suitable model for our purpose.

4. Conclusion

Each deep learning network has its own advantages and disadvantages. In our evaluation phase, our purpose of use and location play an important role in our deep learning network selection. In terms of usage, it was observed that one-stage deep learning networks were more efficient in terms of FPS when it was desired to be used in an area where both mobile and computational constraints were present. Again, it was observed that YOLOv5s algorithm was more efficient among YOLOv5s and SSD algorithms. Therefore, it was observed that YOLOv5s algorithm would be a more suitable algorithm for fire detection. As a result of the study, we trained four different models and then compared these models to obtain the most suitable model for our purpose.

Competing interests

The authors declare that they have no competing interests.

References

[1] Pechony, O., & Shindell, D.T. (2010). Driving forces of global wildfires over the past millennium and the forthcoming century. Proceedings of the National Academy of Sciences, 107, 19167 - 19170.

[2] Asselin, H., Ali, A.A., Finsinger, W., & Bergeron, Y. (2014). Effect of increased fire activity on global warming in the boreal forest. Environmental Reviews, 22, 206-219.

[3] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

[6] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[8] Pincott, J., Tien, P. W., Wei, S., & Calautit, J. K. (2022). Indoor fire detection utilizing computer vision-based strategies. Journal of Building Engineering, 61, 105154.

[9] Moumgiakmas, S. S., Samatas, G. G., & Papakostas, G. A. (2021). Computer vision for fire detection on UAVs—From software to hardware. Future Internet, 13(8), 200.

[10] de Venâncio, P. V. A., Lisboa, A. C., & Barbosa, A. V. (2022). An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. Neural Computing and Applications, 34(18), 15349-15368.

[11] Umar, M. M., Silva, L. C. D., Bakar, M. S. A., & Petra, M. I. (2017). State of the art of smoke and fire detection using image processing. International Journal of Signal and Imaging Systems Engineering, 10(1-2), 22-30.

[12] Le Maoult, Y., Sentenac, T., Orteu, J. J., & Arcens, J. P. (2007). Fire detection: a new approach based on a low cost CCD camera in the near infrared. Process Safety and Environmental Protection, 85(3), 193-206.

[13] Diaconu, B. M. (2023). Recent advances and emerging directions in fire detection systems based on machine learning algorithms. Fire, 6(11), 441.

[14] Sierra, D., Montanaro, W., Kuo, L., & Zohuri, B. (2023). Enhancing Fire Detection through CNN and Transfer Learning: A Comprehensive Research Study. Journal of Engineering and Applied Sciences Technology. SRC/JEAST-242, 174(5), 2-6.

[15] Ghali, R., & Akhloufi, M. A. (2023). Deep Learning Approaches for Wildland Fires Remote Sensing: Classification, Detection, and Segmentation. Remote Sensing, 15(7), 1821.

[16] Chitram, S., Kumar, S., & Thenmalar, S. (2024). Enhancing Fire and Smoke Detection Using Deep Learning Techniques. Engineering Proceedings, 62(1), 7.

[17] Khan, T., Khan, Z. A., & Choi, C. (2023). Enhancing real-time fire detection: an effective multi-attention network and a fire benchmark. Neural Computing and Applications, 1-15.