Fire Detection and Segmentation using YOLOv5 and U-NET

Wided Souidene Mseddi [1,2]
[1] *L2TI, Institut Galilée*
Université Sorbonne Paris Nord
Villetaneuse, France
[2] *SERCOM Laboratory*
University of Carthage
Tunis, Tunisia.
wided.mseddi@univ-paris13.fr

Rafik Ghali
SERCOM Laboratory,
Ecole Polytechnique de Tunisie
University of Carthage
Tunis, Tunisia
rafik.ghali@ept.rnu.tn

Marwa Jmal
Telnet Innovation Labs
Telnet Holding
Ariana, Tunisia
marwa.jmal@groupe-telnet.net

Rabah Attia
SERCOM Laboratory,
Ecole Polytechnique de Tunisie
University of Carthage
Tunis, Tunisia
rabah.attia@ept.u-carthage.tn

***Abstract*—The environmental crisis the world faces nowadays is a real challenge to Human Beings. One notable hazard for humans and nature is the increasing number of forest fires. Thanks to the fast development of sensors and technologies as well as computer vision algorithms, new approaches for fire detection are proposed. However, these approaches face several limitations that need to be resolved, precisely, the presence of fire-like objects, high false alarm rate, detection of small size fire objects, and high inference time. An important step for vision-based fire analysis is the segmentation of fire pixels. Hence, we propose, in this paper, a novel architecture, combining YOLOv5 and U-net architectures, for fire detection and segmentation. Using a dataset of wildland fires mixed with fire-like object images, the experimental results proved that the novel architecture is reliable for forest fire detection without false alarms.**

Keywords— Forest Fires, Fire detection, Fire segmentation, Deep learning, YOLOv5, U-Net

I. Introduction

Forest Fires (FF) are one of the most dangerous and challenging natural disasters today that can threaten humanity and the environment. FF that are not controlled, can make huge damage with disastrous effects to human properties and areas of vegetation. Fires affect more than 350 million hectares every year worldwide [21].

To avoid this dangerous disaster, systems for detecting and monitoring Forest Fire at the early stages are very crucial.

The earliest fire detection systems used to detect fire using numerous sensors such as gas detectors, smoke detectors, flame detectors and temperature detectors, but these techniques are not efficient in the case of forest or wildland Fire detection. Indeed, they have smaller coverage areas, and they do not respond in real-time. To overcome these limitations, vision sensors (embedded or fixed) are the most useful to detect fires with high accuracy, high coverage area, and less error.

Through the years, researchers have proposed many techniques that allowed them to detect and segment fires with high accuracy using image processing and computer vision methods. First, fire color features have been widely used to distinguish fire. These techniques transform the image into another color space, such as YCbCr [1] or YUV [2], and then classify its pixels into fire or non-fire through comparing pixel values to some thresholds. Nonetheless, these methods are limited by the complexity to identify fire characteristics in the image.

Machine-learning techniques also are employed to increase the reliability of fire detection systems. Numerous models are used such as SVM [8] and neural networks [9]. Recently, thanks to their great performance in detecting and identifying objects, Deep Learning (DL) approaches have been investigated to detect and localize forest fire.

Deep Learning techniques helped researchers a lot to extract relevant features that best represent the fire to be described. Indeed, these models have been successfully used in several fields such as image classification, self-driving cars, speech recognition, pedestrian detection, face recognition, cancer detection, etc. [10-12]. For all these applications, DL proved its efficiency in detecting and segmenting different classes of objects [10-12].

For the task of detection, the newly introduced algorithm named YOLOv5 [13] has proved an excellent tradeoff between accuracy and inference time. For the task of segmentation, U-Net [14] has given excellent results and performance on segmenting medical images.

In this paper, we present a novel fire detection system based on DL using pre-trained YOLOv5 and U-Net models concatenated sequentially. For this purpose, we first feed the original images of fires with their annotations to the YOLOv5 model, then, we crop the fire class using the bounding boxes obtained via the detection. Finally, we feed these cropped images to a trained U-Net model using original images with their annotations and we obtain our segmented images with their bounding lines in the same image.

More specifically, this paper makes two main contributions. Firstly, we propose a novel architecture capable of detecting and segmenting fires in an operationally and time-efficient manner aiming to overcome the limitations of state-of-the art techniques. Secondly, our model has demonstrated its high performance on big and small-size fire objects and its ability to distinguish between fire and fire-like objects.

The remainder of the paper is organized as follows: section 2, briefly describes the related works of DL techniques for fire detection. Section 3 describes the proposed deep learning architectures. In section 4, the implementation and the experimental results are presented. Finally, section 5 concludes the paper.

I. Related works

Several fire detection methods have been proposed and have been presented in numerous reviews [5, 15].

In this related works section, we choose to highlight recent advances in Deep Learning techniques.

In [16], GoogleNet, VGG13 and AlexNet models are employed to detect wildfire using unmanned aerial vehicles (UAV). A comparative analysis showed that the modified VGG13 and the GoogleNet presented the better performance [16].

In [17], authors proposed a CNN model called "Fire_Net" to localize and identify fire in aerial images. Using 3561 images, a good performance is achieved. Zhang et al. [18] also used a deep CNN model for fire detection. They used a patch fire classifier to detect whether an image contains fire or not, if it does a CNN is implemented to localize fire. A great result with a detection accuracy of 90% and a false alarm rate of 2.3% is achieved using training data of 1153 images.

Recently, numerous region-based CNNs are employed to detect and localize fire in images/videos. In [19], authors used Faster R-CNN to detect and localize fire in real time. Using a dataset of 12620 images (forest fires, candle fires, gas ranges fire), good performances are obtained in terms of detection accuracy and precision but this solution is not adequate for real time applications due to its high response time.

In the same way, Shen et al. [20] exploited another region based CNN model, Yolo (You Only Look Once) to detect fire from video. Good results are obtained in terms of accuracy, precision and the response time, which is faster, more than 3 times compared, to Faster R-CNN. In [6], Yolo v3 is explored to detect fire using aerial images. Great recognition rate of 83% and speed of 3.2 frames per second were achieved, and proved the reliability of this model to detect forest fire using UAV. Authors, in [4], also employed various region-based CNN models (Yolo, SSD and Faster R-CNN) to identify forest. SSD model which employs multi-resolution feature maps to localize objects at various scales, showed its efficiency to detect forest fire in real-time in terms of speed and accuracy.

II. Proposed Architecture

The proposed fire detection architecture is based on combining two Deep Learning models in order to perform the detection and localization of wildfires. Our overall framework takes an image as input and outputs the localized fire flames. The first step of our model is YOLOv5 [13] and the second is U-Net [14]. In the proposed architecture, these two networks are integrated. First, the network is fed with RGB color images of forest fires, which are processed by YOLOv5 to get the bounding box around the fire area. Once we get these, a Crop Layer is applied to the image obtained from YOLOv5 results so that we get only the parts of the image limited by the bounding box. Then, these cropped images are fed into the U-Net to confirm the presence of flame and detect the precise location of fire. The result is a binary mask representing the fire pixels in the image. Finally, we take the bounding lines obtained and we put on the original images. It is important to mention that we trained U-Net offline and then used the trained weights to segment the cropped images. The model is presented in Fig.1. The novelty of our proposed method is that it allows to detect not only the bounding box where is situated the fire but the shape

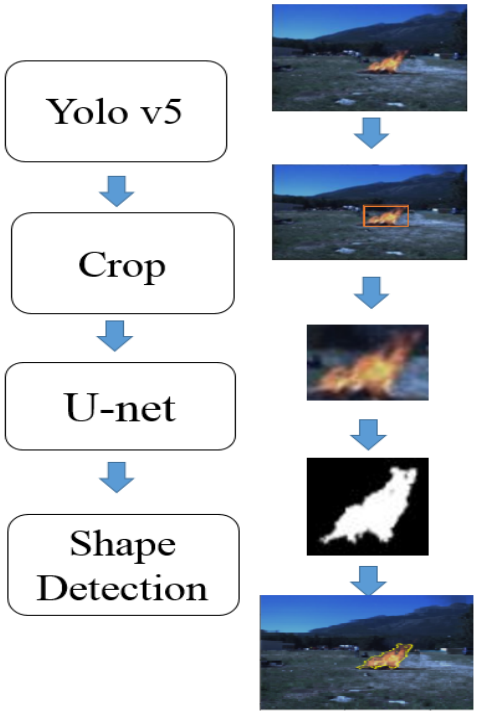

Fig.1. Proposed architecture

A. YOLOv5

YOLOv5 (You Only Look Once) is a single-stage object detector that has three important parts:

- Model Backbone: This part is used to extract important features from the given input image. The Cross Stage Partial Networks (CSPN) are used as a backbone to extract rich informative features from the input fire image
- Model Neck: It mainly consists of generating feature pyramids. Feature pyramids are used to generalize well on object scaling. It helps to detect the same object with various scales and sizes. This is useful for performing well on unseen data.
- Model Head: This is the final detection part. YOLO employs anchor boxes on features and develops the final output, namely, bounding boxes, with a class score.

B. U-NET

U-Net network is a deep convolutional network that has successfully been used in medical image segmentation. Unlike traditional DL models, which are data-hungry, U-Net can still be trained with a small amount of data.

The U-Net is a two-stage deep learning model. Its architecture includes an encoder model followed by a decoder model. It contains nine blocks, four blocks in each stage and one shared block. Each block consists of two 2D convolution layers, which use 3*3 kernel and rectified linear unit activation function, followed by 2D max-pooling layers. The number of feature channels is duplicate at each

down sampling phase. At the final, a 1*1 layer constructs a binary mask.

This model uses a set of input fire images and their corresponding binary masks.

During training and based on the binary mask as the desired output, the model learns how to classify each pixel of the images into the different object labels. For our task, we create two classes that are fire and non-fire.

III. Implementation and Results

In this section, we detail the implementation settings adopted to train and test our proposed techniques. Namely, the data preparation step: collecting dataset and performing data augmentation, the deployment of the overall architecture, the Test Time Augmentation (TTA) module and finally the results collection and the performance analysis.

A. Data Preparation

For fire detection problems, there is no benchmark dataset, which makes a comparative study between DL approaches in the field a bit critical. We create our dataset, which contains the Corsican fire dataset [3] and fire like object images. Corsican fire database is the dataset of forest fire images collected from different research teams in the world. It includes wildfire image sequences acquired in various areas, under numerous conditions like climatic conditions, burning vegetation type, distance to fire and the brightness of fire.

To diversify our dataset and improve the model capability of distinguishing between fire and fire-like objects, we added to the Corsican Fire database numerous images that include fire like colored objects, such as lights, sunrise, sunset, and firefighters clothing, in various resolutions. The newly created dataset contains about 1300 images. They include fire, non-fire, and fire-like images with different resolutions and different sizes.

Fig.2 depicts a sample of the Corsican Fire dataset and fire-like objects images containing objects having some fire characteristics like sunset, sunrise and lights.

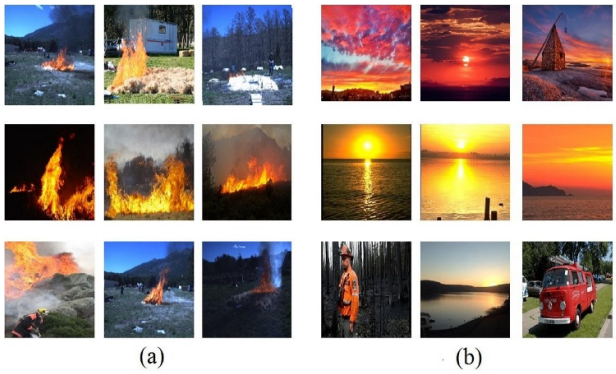

Fig. 2. Examples from the Dataset: (a) CorsicanFire images (b) Fire-like objects images

B. Data Augmentation

It is important to use data augmentation techniques to improve the performance of our model and avoid overfitting. Mainly, it consists of applying transformations on the image such as geometrical transformations (rotation, scaling, padding, cropping, image translation and flip translation), photometric transformation (brightness, contrast, and shear), image occlusion techniques (Mixup) and Mosaic data augmentation, which combine numerous transformations for a single image [7].

As for our problem, we chose the data augmentation techniques based on characteristics of flame. For instance, we did not consider techniques using color space adjustments to keep the fire color information. Besides, we excluded rotation, because for example we cannot find a 90 degree rotated fire in real life. In addition, it is reasonable to flip the fire image horizontally, but it would not be reasonable to flip it vertically. As in the real world, we would not be seeing many images of fire flipped upside-down. In conclusion, we used image translation, image scale, mosaic, mix up, and horizontal flip as augmentation techniques.

C. Training

We trained the two models using the Pytorch framework on a machine with GPU NVIDIA Tesla P100 16 GB. Moreover, we divided our dataset into two subsets as presented in Table I.

TABLE I. DATASET SUBSETS

	Number of positive images	**Number of negative images**	**Number of annotated bounding boxes**
Training set	883	107	1367
Validation set	100	15	237

1. Detection Training

To train YOLOv5, the input data are PNG images and TXT files containing details of annotated objects. Our training was conducted using Binary Cross-Entropy with Logits Loss function from PyTorch for loss calculation of class probability and object score, a learning rate set to 0.01, a batch size of 8, a number of epochs set to 300, and an image size set to 416x416 or 1024*1024. Note that the training time changes depending on the models since we trained the small YOLO and the Large YOLO.

2. Segmentation Training

To train U-Net, the input data are PNG images alongside their corresponding binary masks. We used loss (LS) which is a combination for the loss function of Dice Coefficient (DC) and Binary Cross entropy (BCE) as follows:

$$LS = 1 * e^{-3} * BCE + DC \quad (1)$$

Resized images of 256x256, are used in various training epochs. We also implemented a learning rate of 10^{-4}, Adam as optimizer and a batch size of 4.

D. Test Time Augmentation

Test Time Augmentation is yet another data augmentation method. While data augmentation is done before or while the training of the model, this one is done during the inference time. It is a simple but effective way to avoid over fitting and optimize results. The idea is to show different versions of the same image to the same model, take the different outputs and extract the detected bounding boxes and then combine the results. This is a very fast way to improve the model performance (confidence of the output) without losing a precious time for data augmentation.

E. Evaluation metrics

The evaluation metrics used in this work are divided into detection metrics and segmentation metrics.

1. Fire detection metrics

- Recall is the value of the percentage of total relevant results correctly classified.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Where *TP*: True positive, *FN*: False Negative.

- MAP (Mean Average Precision) is the mean of AP. The AP value of the different classes is calculated as follows:

$$AP = \sum_{1:N} (r_m - r_{m-1}) p_m \quad (3)$$

where *r* and *p* are the recall and the precision at the mth threshold.

2. Fire segmentation metrics

- Dice Coefficient (DC): The Dice Coefficient is a statistical indicator that measures the similarity of two images (predicted and ground truth images).

$$DC = \frac{2 * TP}{2\,TP + FP + FN} \quad (4)$$

where *TP*: True positive, *FP*: False positive and *FN*: False negative.

- Accuracy: is the fraction of correct predictions over the number of total predictions achieved by the network.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (5)$$

where *TP*: True positive, *TN*: True Negative, *FP*: False positive and *FN*: False negative.

F. Experimental Results

In this section, we discuss the experimental results to demonstrate the performance of the proposed method. First, we discuss the detection results. Then we present the segmentation results. Our test set consists of 185 images containing 221 bounding boxes.

1. Detection Results

In table II, we present the performance of both YOLOv5s (Small version) and YOLOv5x (Extra-large version) with and without TTA.

TABLE II. FIRE DETECTION RESULTS

Models	TTA	Recall	MAP
YoloV5s	ON	0.842	0.732
	OFF	0.805	0.686
YoloV5x	ON	0.869	0.718
	OFF	0.769	0.654

We can see that the two versions of YOLOv5 achieved great results in terms of fire detection using TTA techniques and data augmentation techniques.

Due to high resolution of input images, YOLOv5 xlarge achieved the best results compared to YOLOv5s.

In fig. 3, we can see that Yolov5 has accurately detected and localized forest fire, precisely small fire. Accordingly, the model overcomes the confusion with fire like-objects like sunrise and sunset.

Fig. 3: YOLOv5 results

2. Segmentation Results

The table III presents the best scores that we got for our model.

TABLE III. EXPERIMENTAL RESULTS OF U-NET

Model	Dice coefficient	Accuracy
U-net	92%	99.6%

We can see that the U-Net model achieved a great performance (dice coefficient of 92% and accuracy of 99.6 %) to segment forest fire. The YOLO v3 applied to our database provides an accuracy of 96.8%.

We could attest that the strength of U-Net is its ability, not only to confirm the presence of the forest fire but also to detect the precise shape of flame. In fig.4, we can see that the network accurately and precisely detects the fire and its shape. By combining the two architectures, we achieved a

robust and precise forest fire detector to solve forest fire detection and recognition problems.

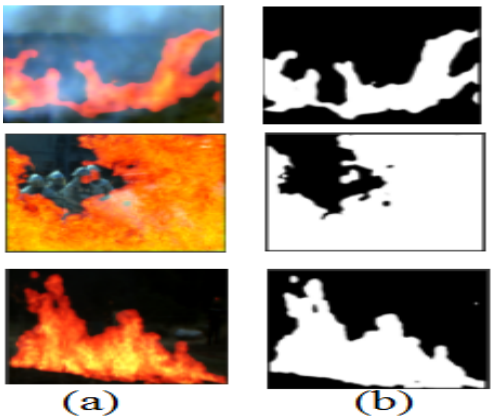

Fig. 4: U-Net results: (a) cropped images (b) predicted images

As an example, we can see in fig. 5 that our proposed model performs very well in detecting fire pixels and segmenting fire surfaces, especially small areas (figure in the middle), and it has successfully overcome the confusion with fire-like objects (figure in the right). These results outperform the state-of-the-art fire detection techniques.

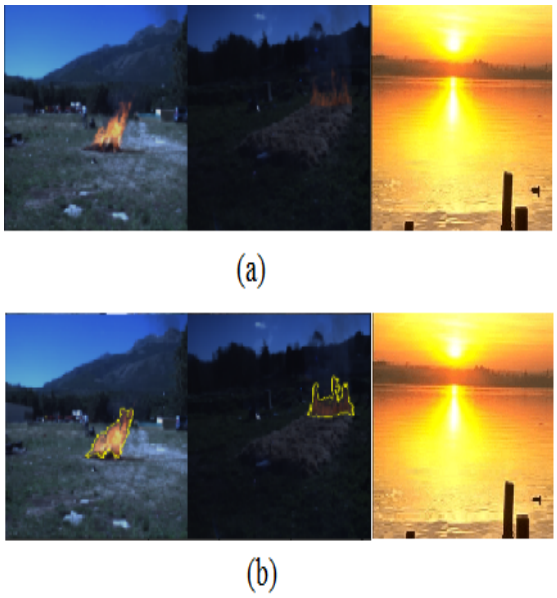

Fig. 5: Results: (a) input mages (b) output models

IV. Conclusion

In this paper, we introduced a novel method of forest fire detection and segmentation based on YOLOv5 and U-net. Using Corsican Fire dataset and various fire-like objects images, we evaluated our methods. Experimental results proved that this method is able to detect forest fires precisely small fires (with small flames) in different acquisition conditions. For future work, we aim to introduce a smoke/fire detection method based on CNN in order to identify and localize both fire and smoke without false alarms.

Acknowledgement

This project is carried out under the MOBIDOC scheme, funded by EU through the EMORI program and managed by the ANPR.

References

[1] T. Celik and H. Demirel, “Fire detection in video sequences using a generic color model,” Fire Safety Journal, vol. 44, no. 2, pp. 147–158, 2009.

[2] G. Marbach, M. Loepfe, and T. Brupbacher, “An image processing technique for fire detection in video images,” Fire Safety Journal, vol. 41, no. 4, pp. 285–289, 2006.

[3] T. Toulouse, L. Rossi, A. Campana, T. Celik, M. Akhloufi.:” Computer vision for wildre research: An evolving image dataset for processing and analysis,” Fire Safety Journal 92, 188-194, 2017.

[4] S. Wu and L. Zhang, “Using popular object detection methods for real time forest fire detection,” in 11th International Symposium on Computational Intelligence and Design, 1, 280–284, IEEE, 2018.

[5] R. Ghali, M. Jmal, W. Souidene Mseddi, and R. Attia, “Recent advances in fire detection and monitoring systems: A review,” in the International Conference on the Sciences of Electronics, Technologies of Information and Telecommunications. Springer, 2018.

[6] Z. Jiao, Y. Zhang, J. Xin, et al., “A , 2019 based forest fire detection approach using uav and yolov3,” in 1st International Conference on Industrial Artificial Intelligence (IAI), 1–5, IEEE, 2019.

[7] C. Shorten and T. M Khoshgoftaar, “A survey on image data augmentation for deep learning,” Journal of Big Data, vol. 6, no. 1, pp. 1–48, 2019.

[8] Y. Chunyu, F. Jun, W. Jinjun, et al., “Video fire smoke detection using motion and color features,” Fire Technology, 46 (3), 651–663, 2010.

[9] M. A. I. Mahmoud and H. Ren, “Forest fire detection and identification using image processing and svm,” Journal of Information Processing Systems, 15(1), 159–168, 2019.

[10] Z.-Q. Zhao, P. Zheng, S.-t. Xu, et al., “Object detection with deep learning: A review,” IEEE transactions on neural networks and learning systems, 30(11), 3212–3232, 2019.

[11] L. Liu, W. Ouyang, X. Wang, et al., “Deep learning for generic object detection: A survey,” International Journal of Computer Vision, 128 (2), 261–318, 2020.

[12] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz & D. Terzopoulos. “Image segmentation using deep learning: A survey,” arXiv preprint arXiv:2001.05566, 2020.

[13] G. Jocher, A. Stoken, J. Borovec, N. Code, C. STAN, L. C. Laughing, T. YxNONG, A. Hogan, L. mammana, A. Wang, A. Chaurasia, L. D. Marc, W. Haoyang, D. Durgesh, F. Ingham, F. Guil-hen, A. Colmagro, H. Ye, J. Solawetz, J. Poznanski, J. Fang, J. Kim, K. Doan, and L. Yu, “ultralytics/yolov5:v4.0, PyTorch Hubintegration,” Jan. 2021

[14] O. Ronneberger, P. Fischer and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” Springer , 2015.

[15] A. Gaur, A. Singh, A. Kumar, A. Kumar,K. Kapoor.: “Video flame and smoke based re-detection algorithms: A literature review,” FIRE TECHNOLOGY, 2020.

[16] W. Lee, S. Kim, Y.-T. Lee, et al., “Deep neural networks for wildfire detection with unmanned aerial vehicles,” in IEEE International Conference on Consumer Electronics (ICCE), 252–253, IEEE ,2017.

[17] Y. Zhao, M. Jiale, L. Xiaohui and Z. Jie, “Saliency detection and deep learning-based wildfire identification in UAV imagery,” Sensors, vol. 18, no. 3, p. 712, 2018.

[18] Z. Qingjie, X. Jiaolong, X. Liang and G. Haifeng, “Deep convolutional neural networks for forest fire detection,” in International Forum on Management, Education and Information Technology Application. Atlantis Press, 2016.

[19] Q.-x. Zhang, G.-h. Lin, Y.-m. Zhang, G. Xu and J.-j. Wang, “Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images”, Procedia engineering, vol. 211, pp. 441-446, 2018.

[20] D. Shen, X. Chen, M. Nguyen and W. Q. Yan, “Flame detection using deep learning,” in 4th International Conference on Control, Automation and Robotics (ICCAR). IEEE, 2018.

[21] D. Stav. "Fighting fire with science". Nature, vol. 576, no 7786, p. 328-329, 2019