International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.14, No.1 (Oct-23)

http://dx.doi.org/10.12785/ijcds/140188

Wildfire Detection System using YOLOv5 Deep Learning Model

Ahmed Saleem Mahdi[1] and Sawsen Abdulhadi Mahmood[1]

[1]*Computer Science Department, College of education, Mustansiriyah University, Baghdad- Iraq*

Received 23 Aug. 2022, Revised 30 Jul. 2023, Accepted 29 Aug. 2023, Published 1 Oct. 2023

Abstract: Wildfires pose a significant threat to the natural environment as well as public safety.Forest fire detection is critical for effective fighting, as once a wildfire has grown to a certain size, it is difficult to control. Recently, there has been a growing demand for forest areas to install a rapid response system to allow for prompt and timely action in the event of forest fires expanding across large areas. In this paper, a proposed framework for wildfire detection in a video sequence using the YOLOv5 deep learning model is presented and implemented. The interested regions represented by (fire object) in the video sequence are extracted using a new auto-annotation scheme to determine the ROI (Region of Interest) based on the edge detection process. Since the public wildfire datasets are yet confined, therefore we have constructed a new wildfire dataset named WILDFIRE-I dataset composed of variant fire images to conduct the performance evaluation of the proposed system. A comparison study with state of art research was performed in our experiments to demonstrate the efficiency of the proposed system based on common performance evaluation metrics. The experimental results exhibited detection accuracy of fire events close to 98 %, with a manual annotation process, while the proposed annotation process has achieved an accuracy of 96 %, with minimum time processing required for dataset image labelling.

Keywords: Wildfire Detection; YOLOv5; Fire Annotation, WILDFIRE-I dataset

1. Introduction

Preventing fire disasters and protecting lives and property requires early detection of wildfires before they may spread out of control. Fires in the wild can happen to anyone, at any age, in any environment. Wildfires and forest fires have always altered the terrain [1]. Most fires have small beginnings and grow rapidly, making them very challenging to contain. Detection of wildfires is essential in the fight against their large scale. Recently, there has been a lot of interest in developing real-time algorithms for detecting wildfires using typical video-based surveillance systems [2]. The loss of forest cover and the biodiversity it previously supported can be mitigated with the use of a forest surveillance program designed to detect and identify wildfires. Reducing the time it takes to report a fire to authorities is a major aspect that could keep wildfires under control [3]. Therefore, a deep learning approach was required to recognize wildfires, and a massive dataset was accumulated to precisely predict the risk of wildfires and aid in their prevention[4]. In this paper, a wildfire detection system is presented and investigated using YOLOv5 deep learning model to extract and detect fire object in the acquired images /frames. In the training mode, we have gathered set of wildfire images to construct a WILDFIREI-dataset based on our efforts. In this context, two main strategies were performed to accomplish the labelling process of the trained images. the first strategy based on manual labelling workflow [5], while the second one was proposed to present a new automated approach to label and annotate the training wildfire images by employing edge detection process.

2. Related Works

Wildfire detection is an active area of research, with many published papers and presentations, those who have utilized CNN, others employed R-CNN to build the Wildfire detection system, in addition to the studies that used YOLO (You Only Look Once), in the following section, the previous and related work will be illustrated:

Yuanbin W. et al. presented a Convolutional Neural Network (CNN) based wildfire recognition technique. Immediate use of the raw CNN, the network's learnt characteristics may not be precise enough, which may influence the recognition rate. To address these problems, an adaptive pooling strategy is combined with traditional image processing methods and CNNs. The flame region can be preemptively split and its features identified using this method. Both the inaccurate features being learned by the CNN and the blindness that occurs during the standard feature extraction method are avoided simultaneously. Adaptive pooling, a variant of the CNN, has been shown to get better results

E-mail address: asm_msc@uomustansiriyah.edu.iq, sawsenhadi@uomustansiriyah.edu.iq

and a higher recognition rate in experiments [6]

Hong et al. proposed a CNN-based system for real-time fire detection (FireCNN). Effectively extracting the right features of fire spots, FireCNN blends multi-scale convolution with residual acceptance architecture. The proposed method was tested on a dataset consisting of 1,823 fire locations and 3,646 non-fire locations. From the results of the tests, it is clear that the FireCNN is capable of identifying wildfires with an accuracy that is 35.2% higher than the traditional threshold method. In addition, they investigated how different architectures affected the efficacy of neural network models [7].

Seydi S. et al. offer Fire-Net, a deep learning network trained on Landsat-8 data to recognize active fires and material in a burn, for use in CNN. For a clearer picture, they fuse the optical (RGB) and thermal (heat) modalities of the images. The network also makes use of residual convolution and separable convolution blocks, which allows for the extraction of deeper features from coarse datasets. All in all, the experimental findings show an accuracy of 97.35%, with strong identification of even the smallest of active flames. Forests in Australia, the United States, Canada, the Amazon, Central Africa, and Chornobyl (Ukraine) were used because they have extensive records of wildfires that were used to compile this dataset [8].

Xu et al. present an innovative ensemble learning system for detecting wildfires in various settings. First, two individual learners, Yolov5 and EfficientDet, are merged to complete the process of fire detection. Second, another EfficientNet learner is accountable for acquiring global knowledge to avoid false positives. In conclusion, detection results are determined based on the judgments of three learners. Experiments conducted on our dataset demonstrate that the proposed strategy increases detection performance by 2.5% to 10.9% and reduces false positives by 51.3% without introducing any additional latency [9]. To make the model more resilient to multiple forest fire scenarios, two powerful object detectors (Yolov5 and EfficientDet) with different expertise are merged. Next, EfficientNet guides the detection process to decrease false positives. Experimental results reveal that our model provides a better trade-off between average precision, average recall, false positive rate, frame accuracy, and latency than other popular object detectors. Significant improvements allow the model to perform well in real-world forestry applications [10].

In their work, Hoor et al. used UAV drones to spot potential wildfire hotspots. They show a deep learning model for fire detection based on YOLOv5. By analyzing a video pixel by frame, the suggested system may reliably spot fires in real-time and issue timely alerts to the proper authorities. Their method provides better detection performance than current fire detection systems. Our evaluation of the proposed approach found an F1-score of 94.44% on the FireNet and FLAME aerial image datasets[11]. This study presents a deep learning model for real-time fire detection that is both effective and accurate. Our proposed approach to fire detection in forests and bushlands is based on the YOLO algorithm. The suggested method outperforms the current state-of-the-art deep learning-based methods. With this method, we get an F1-score of 94.44%. Technology like this will help authorities constantly monitor fire-prone forests all around the world to catch blazes in their early stages, and it will aid forest departments in spotting flames in widely dispersed forests.

Zhang et al. shows a brand-new model of an algorithm called Swin-YOLOv5. The Swin transformation mechanism was added to the YOLOv5 network to improve the model's field of view and ability to extract features without changing the model's depth. The feature splicing method of the network's three output heads was changed to improve the feature map splicing method of weighted Concat and the ability of model pairings to combine features [12]. More changes were made to the feature fusion module, and the weighted feature splicing technique was developed, all to boost the network's feature fusion performance. According to experiments, this strategy improves the map (average rage accuracy) more quickly than the gold standard algorithm. This approach improves high-precision target recognition speed by 1.8 frames per second while increasing map accuracy by 0.7% on the same experimental dataset (fast packet switch). The improved algorithm, when tested on the same experimental dataset, was better able to detect targets that were either missed by the original algorithm or incorrectly detected by it. This demonstrated the algorithm's flexibility in detecting real-world scenes and had important practical implications. This work not only produced a practical concept for feature extraction and fusion of YOLOv5, but it also opened the door for the use of fire-smoke detection in forest and indoor scenarios [13].

Zhenyang et al. demonstrate how YOLOv5 can be used to make a better algorithm for finding small targets in wildfires. First, they improved the layer of YOLOv5 Backbone and changed the SPPF (Spatial Pyramid Pooling-Fast) module of YOLOv5 to the SPPFP (Spatial Pyramid Pooling-Fast-Plus) module to focus more on the global information of small forest fire targets. Then, they added the (CBAM) Convolutional Block Attention Module CBAM to make it easier to find targets in small forest fires. Second, a very small-target detection layer was added to the Neck layer of YOLOv5, and the PANet (Path Aggregation Network) was changed to the BiFPN (Bidirectional Feature Pyramid Network) . Because the first small-target forest fire dataset is a small sample dataset, training was done with a migration learning method [14]. The experimental results suggest that the model performs better than YOLOv5s, which bodes well for its potential use in small-target forest fire detection.

Mahdi et al. employes a YOLOv5 deep learning model is to train a model that can distinguish between fire and

TABLE I. Related works summary

Study	Method	Dataset	Accuracy
[6]	CNN	-	93.75%
[7]	FireCNN	Himawari-8	99.90%
[8]	FireNet	Coarse	97.35%
[9]	YOLOv5 and EfficientDet	Their Created dataset	99.70%
[11]	YOLOv5	FireNet	93.00%
[12]	Swin-YOLOv5	Fire-smoke dataset	60%
[14]	YOLOv5	Forest fire & Small-target forest fire dataset	-
[15]	YOLOv5	WILDFIRE-I	98.00%

non-fire events in binary classification. The proposed system architecture consists of IoT entities equipped with camera sensor capabilities and an NVIDIA Jetson Nano Developer kit as the edge computing environment. At the first level, a video camera is used to compile environment information received by the microcontroller at the middle level in order to manage and detect a potential fire event in the region of interest[15]. The experimental results demonstrated a detection accuracy of 98% for fire events within a video series.

Table I shows a summary of the above methods, including the dataset used and how accurate each one is.

3. Materials

The purpose of this paper is to use a YOLOv5 deep learning model that was retrained on a synthetic wildfire dataset to help with the problem of detecting wildfires. Our method employs both manual annotation of WILDFIRE-I dataset samples and a novel auto-annotation scheme to annotate datasets. The proposed auto-annotation scheme employs an edge detection procedure to rapidly annotate the training samples with the presence of a fire item. Following sections explain how to configure the necessary components of the system and prepare the necessary datasets before putting the suggested system into action.

A. System Requirements

In practice, deep learning models have demanded greater specs in terms of the speed up of the processor and the amount of storage space available. Also the implementation of the proposed method necessitated the use of specialized programs, such as: (OS: Windows 10, and Programming Language: Python). In addition, the following pieces of software are needed: (PyCharm, Anaconda, PyTorch with Torchvision, Cuda, and the last version of Python libraries). Table II details the main crucial aspects of the experimental conditions that was utilized to put the suggested procedure into action.

B. Dataset Description

This paper presents an evaluation of the proposed wildfire detection system which carried out using WILDFIRE- I dataset [5], which contains 3,436 wildfire images, and 3,436 non-wildfire images. Both classes are already annotated. The experiments were implemented with the original dataset annotation, in addition of a new annotation scheme of dataset images for training and validation phases. for both techniques, the WILDFIRE-I dataset was divided into 80% training and 20% testing, as clarified in Table III.

TABLE II. Experimental conditions

Programming language	Python 3.8
Operating system	Windows 10
Laptop Type	Asus Rog
Processor	Ryzen 5
GPU	RTX 3050 4 GB
RAM	16 GB

TABLE III. WILDFIRE-I dataset splitting

Category	Train 64%	Valid 16%	Test 20%	Total
Fire	2199	550	687	3436
No Fire	2199	550	687	3436
Total	4398	1100	1374	6872

4. Method

A. Auto Annotation Scheme

In the training mode of Yolov5 network-based object detection, the annotation process must first locate the interested regions in the input image using ROI. ROI task could be done manually by drawing a box around fire zone in each training image in the dataset. To automatically extract the interesting parts of the input image, we used a modified color-based annotation strategy. Since a fire object has no typical shape, auto-detection is difficult.

The proposed auto-annotation scheme uses edge detection and algorithms to extract the fire object from photos. First, a fire region mask FRM is constructed, then canny edge detection [16] is used to identify fire object edges. This specifies the output shape's upper and lower angles. Finally, WILDFIRE-I is annotated. Figure 1 and Algorithm 1 show the auto-annotation steps.

1) Region of Interest Extraction (ROI)

In order to extract the fire object from input image, we have proposed an auto-annotation scheme involves constructing set of rules to exploits the color information of the acquired image. These rules are implemented upon the

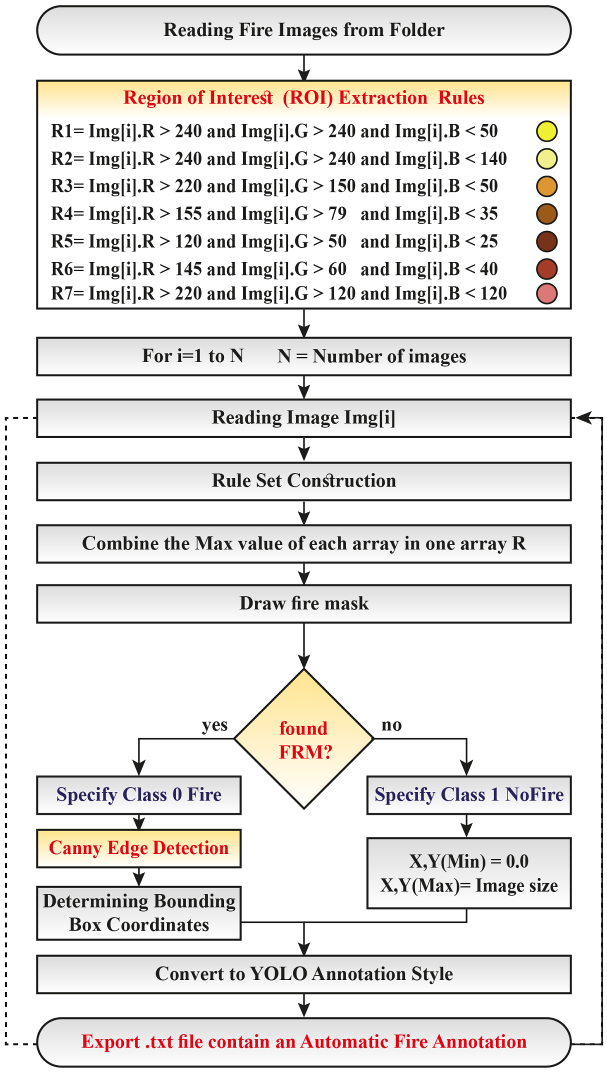


Figure 1. Auto fire feature extraction and annotation steps

three color channels (Red-Green, Blue) of input image to decide if a specific pixel is located within fire region or not. Figure 2 illustrated the RGB channels historam, displaying the amount of each color pixels values. The constructed rules R1, R2,.....R7 are illustrated in equations (1-7):

$$Img.R > 240\ AND\ Img.G > 240\ AND\ Img.B < 50 \quad (1)$$

$$Img.R > 240\ AND\ Img.G > 240\ AND\ Img.B < 140 \quad (2)$$

$$Img.R > 220\ AND\ Img.G > 150\ AND\ Img.B < 50 \quad (3)$$

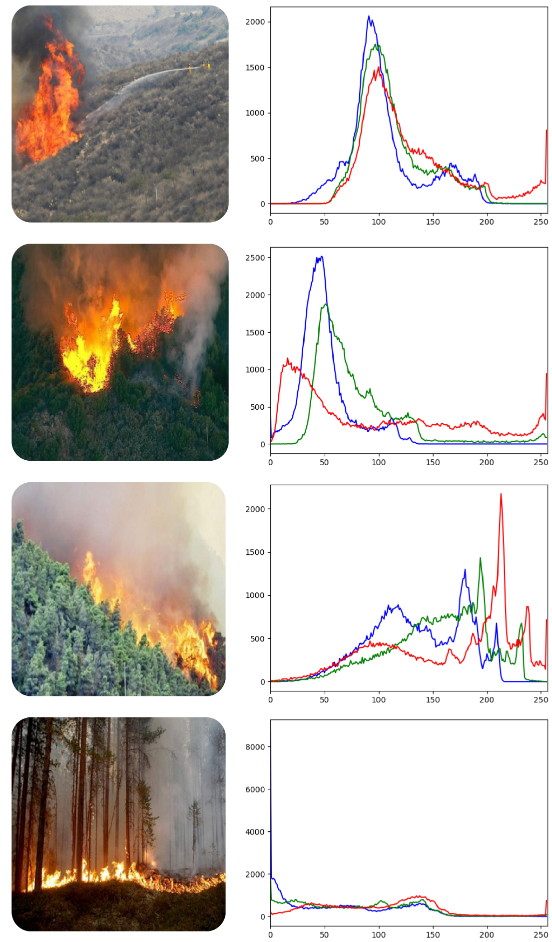


Figure 2. RGB channels histogram

$$Img.R > 155\ AND\ Img.G > 79\ AND\ Img.B < 35 \quad (4)$$

$$Img.R > 120\ AND\ Img.G > 50\ AND\ Img.B < 25 \quad (5)$$

$$Img.R > 145\ AND\ Img.G > 60\ AND\ Img.B < 40 \quad (6)$$

$$Img.R > 220\ AND\ Img.G > 120\ AND\ Img.B < 120 \quad (7)$$

The built rules are combined in one data structure array termed R by comparing each pixel's maximum value. Then, a white-and-black fire mask (FRM) is created. Then, edge detection is applied. Implementing Canny edge detection by; Noise Reduction, Finding the Intensity Gradient of the Image, and Non-maximum Suppression[16].

TABLE IV. THE REGULAR FORM OF COORDINATES

	Xmin	Xmax	Ymin	Ymax
Regular form	34	274	98	254

TABLE V. YOLO FORM OF ANNOTATION COORDINATES

	Xcenter	Ycenter	Width	Height
YOLO form	0.513	0.586	0.8	0.52

2) *Determining Bounding Box Coordinates*

The interesting region is chosen by the top and lower corner coordinates (Xmin, Ymin, Xmax, Ymax) extracted from the points created by the edge and utilized to draw the annotation's bounding box. Table IV and Figure 3 (a) shows the process of determining the coordinates of bonding box represented by maximum and minimum values of X, Y coordinates respectively.

3) *Convert to YOLO Annotation Style*

In this stage, it had to convert from the regular form to the YOLO scheme, the following table V shows the coordinates of annotation in YOLO form after converted it from regular form. Figure 3 illustrate the difference between regular and YOLO annotations forms style.

4) *Export YOLO Style File*

Finally, export txt file containing the YOLO style and carrying the same name as the image. It's mean that every image will have an annotation txt file carrying the same name of it. Table V show the contents of exported file.

B. *Learning Model – Based Wildfire Detection*

The challenge of spotting wildfires in a set of images or videos is an example of an object detection problem that has to be solved. In this paper, researchers use the YOLOv5 learning model for fire detection because it is the most efficient and fastest learning model for object recognition and classification [17]. It is possible to use CNN-based object detectors, which are typically employed in recommendation systems, to do real-time object detection with less computational overhead [18]. By starting the weights at random values between [0, 1], we may achieve precise detection while using less computational time. Figure 4 shows the main structure of the original YOLOv5.

Figure 3. Regular vs YOLO annotation styles (a) Regular style (b) YOLO style

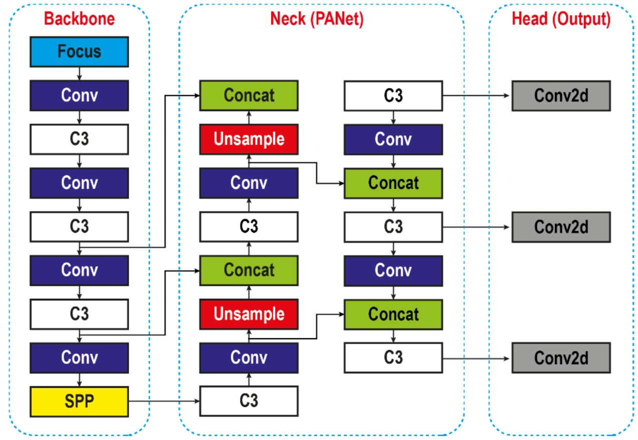


Figure 4. YOLOv5s Model Architecture [15]

TABLE VI. Description of each term of YOLOv5 structure

Conv	Typical Convolution Layer
C3	3 Convolutions Layers architecture
Concat	Joining of two separate layers.
SPPF	Layered Pyramidal Spatial Pooling
Detect	The Result of the Network

5. Evaluation

A. *Confusion Matrix*

With the use of the WILDFIRE-1 dataset, the effectiveness of the wildfire detection method provided here is assessed. Our evaluation makes use of the confusion matrix benchmark [20], which incorporates four primary metrics (TP, TN, FP, FN) to provide a transparent simulation result, as shown in Figure 5. It allows us to evaluate the efficacy of our model, pinpoint its flaws, and obtain direction for making improvements.

B. *Performance Evaluation Metrics [21]*

- Accuracy: The number of correct predictions your model made for the entire test dataset.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

- Precision: The percentage of expected successes that actually came to fruition.

$$Precision = \frac{TP}{TP + TN} \tag{9}$$

- Recall: represents the percentage of true positives that were predicted by our model.

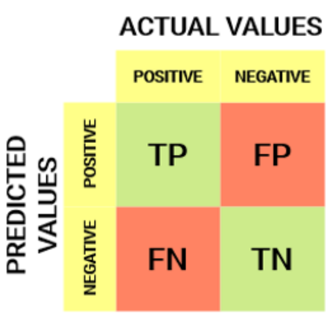
$$Recall = \frac{TP}{TP + FN} \tag{10}$$

- F1 Score: Equal to the median of both recall and accuracy.

$$F1Score = \frac{2 * (precision * recall)}{(precision + recall)} \tag{11}$$

Algorithm 1 Auto ROI Extraction and Annotation

$Input: \ WILDFIRE-I, \ Constructed \ Rules \ R$
$Output: \ Annotated \ images$
Initialization:
$FRM \leftarrow Fire \ Mask$
$N \leftarrow Image \ Count$
$Class0 \leftarrow Fire \ Class$
$i \leftarrow 0$
Begin:
while $i \neq N$ **do**
  $Read \ Image \ Img[i]$
  $Apply \ Rules-based \ Region \ of \ Interest \ (ROI)$
  $Combine \ the \ Max \ value \ of \ each \ array \ in \ one \ array \ A$
  $Draw \ FRM$
  if FRM = "Found" **then**
    $Specify \ Class \ 0$
    $Apply \ Canny \ Edge \ Detector$
    $Determine \ interested \ region \ coordinates \ (XMin, \ YMin, \ XMax, \ YMax)$
    $Convert \ to \ YOLO \ Annotation \ Style$
    $Export \ .txt \ file \ contains \ an \ Automatic \ Fire \ Annotation$
  end if
  $i = i + 1$
end while
End



True Positive (TP)
- The predicted and actual values are the same.
- The actual value was positive, as predicted by the model.

True Negative (TN)
- The predicted and actual values are the same.
- The model predicted a negative value, but the actual value was negative.

False Positive (FP) – Type 1 error
- The expected value was incorrectly predicted
- he models predicted a positive value, but the actual value was negative.
- Also known as the Type 1 error

False Negative (FN) – Type 2 error
- The expected value was incorrectly predicted
- The model predicted a negative value, but the actual value was positive.
- Also known as the Type 2 error

Figure 5. CONFUSION MATRIX WITH 2 CLASSES [19]

6. Results

In this section, all of the paper's results will be presented, the first of which pertains to auto-annotation scheme and the second to YOLOv5, which contains two tests, the first with Wildfire-I Dataset Annotation and the final with YOLOv5 and auto-annotation scheme.

A. Auto-annotation scheme results

It will first reveal the results of the auto-annotation scheme procedure, the results of which are shown in the table VII, which show that the amount of images utilized in the procedure may be seen here: 2,749 images from the WILDFIRE-I Dataset's training class. The total number of correctly labeled images is 2,677, while the margin of error is 72, for a total accuracy of 97.4 %. It only took a few milliseconds for a single image to be analyzed.

TABLE VII. Auto fire features extraction and accuracy

Fire Images	True	False	Accuracy	TpF
2749	2677	72	97.4 %	0.1 sec

Figure 6 shows in detailed explanation the steps involved in auto-annotation scheme process, which contain the (Input Image, Feature Extraction Mask, Canny Edge Detection and Annotation Generation in YOLO style).

B. Exterminates Implementation

There were two studies conducted to prove the efficacy of the YOLOv5 network:

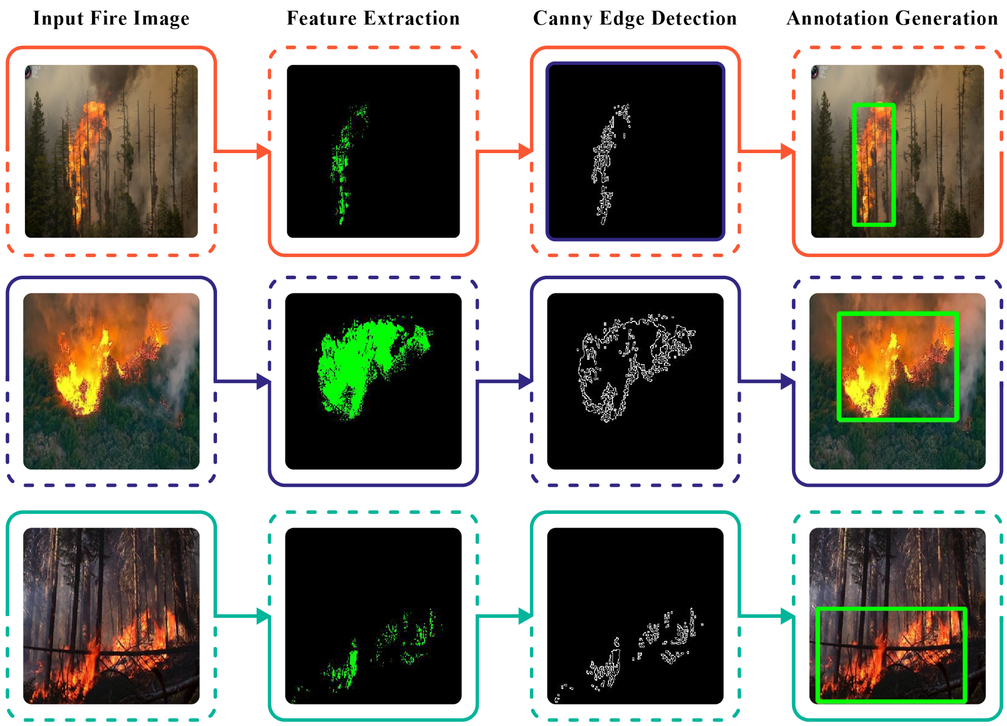


Figure 6. Auto Fire Feature Extraction and Annotation Example

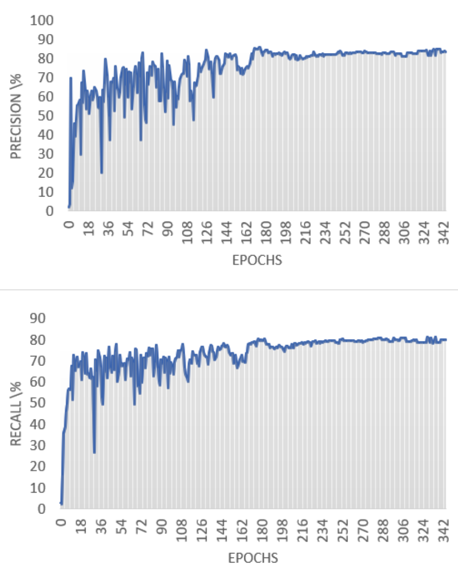


Figure 7. Results of YOLOv5 experiment training phase

TABLE VIII. Hyper-parameters of YOLOv5 experiment training phase

Train Images	2199 Images
Valid Images	550 Images
Test Images	687 Images
Image Size	224×224
Train Time	2 Hours
Batch Size	32
Filter Size	3×3

TABLE IX. Results of YOLOv5 testing phase

Pre-process testing time (ms)	0.3
Inference testing time (ms)	9.7
NMS per image testing time (ms)	1.1
Accuracy	98.76%
Recall	97.86%
Specificity	99.70%
Precision	99.71%
F1 Score	98.77%

1) YOLOv5 with WILDFIRE-I original annotation

In this experiment, YOLOv5 using WILDFIRE-I implemented with the hyper-parameters shown in Table VIII, and recorded the obtained results as described in Table IX and in Figures (7, 8).

2) YOLOv5 with Auto fire feature extraction and annotation

This stage implemented with the resulted annotations from our method, Table X shows the hyper-parameters used in this experiment. In addition, an optimal design of hyperparameters for the YOLOv5 network based on auto-annotation scheme was accomplished. The outcomes of this experiment are depicted in Table IX and in Figures (9, 10).

		Actual Values	
		Positives	Negatives
Predicated Values	Positives	685	2
	Negatives	15	672

Figure 8. Results of YOLOv5 confusion matrix

TABLE X. Hyper-parameters of YOLOv5 based on auto-annotation scheme training phase

Train Images	2199 Images
Valid Images	550 Images
Test Images	687 Images
Image Size	224 × 224
Train Time	0:55 Hour
Batch Size	32
Filter Size	3 × 3

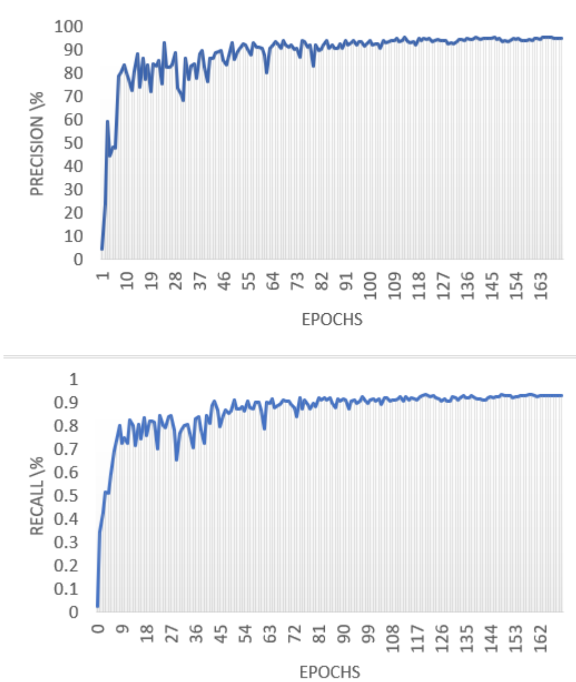


Figure 9. Results of YOLOv5 based on auto-annotation scheme training phase

		Actual Values	
		Positives	Negatives
Predicated Values	Positives	650	37
	Negatives	18	669

Figure 10. Results of YOLOv5 based on auto-annotation scheme confusion matrix

TABLE XI. Results of YOLOv5 testing phase

Pre-process testing time (ms)	0.4
Inference testing time (ms)	8.0
NMS per image testing time (ms)	1.1
Accuracy	96.00%
Recall	97.31%
Specificity	94.76%
Precision	94.61%
F1 Score	95.94%

C. Comparison Study

According to Table I mentioned in section 2, the most related works were listed along with methods and dataset employed in their frameworks, revealing a wide range of approaches and levels of accuracy. The studies that incorporated YOLO model into their methodology represented a considerable portion of the prior research literature. However, their results varied widely in terms of accuracy, the authors in [9] used YOLO network combined with EfficientDet yielding the best results. the framework of research [12], offered their thoughts on how YOLO could be improved, yielding the lowest. In contrast, [15] found that the purity of a single YOLO, without any additions, yielded an accuracy of 98

Based on knowledge, all of the datasets used in the training mode of the related works were labelled manually. It is well known that the manual labelling of the annotation process takes a lot of time and efforts. Hence, in order to reduce the consuming time of training workflow, we have introduced an auto-annotation scheme based labelling process to facilities our framework. Comparing the researches in Table I with own results stated in Table XI, we have found that the proposed method has a higher detection accuracy.

7. Discussion

Referring to table IX, which depicts YOLOv5 training with Manual Annotation, and Table 9, which represents YOLOv5 training with the auto-annotation scheme, as well as table IV, which explains the time spent in auto-annotation scheme, is concluded that, while the time spent in the auto-annotation scheme process is relatively short in comparison to manual annotation, the results obtained from training the network within the dataset resulting from auto-annotation scheme are superior to those obtained from manual annotation, and recorded a difference of 2% in accuracy. The results obtained in this section on auto-annotation scheme show that the method is effective in extracting Fire ROI with little variation in accuracy and efficient use of time.

8. Conclusions

Using the YOLOv5 deep learning model, this research presents an innovative way for identifying wildfires. The results of our tests demonstrate that the YOLOv5 method is capable of achieving a high level of detection accuracy. The batch size parameter was principally adjusted in order to get the desired result of more precise detection. Depending on the computer specifications that are used, the training operation can take a broad variety of amounts of time and provide a large variety of outcomes. The underlying technology that allows for the extraction of the auto fire feature is adaptable enough to be utilized in other scenarios.

One of the future ideas that might be developed for the study is to utilize this method to extract features for other objects that are similar to fire, such as light sources and the sun, by using the color spectrum. This is one of the future possibilities. Additionally, it is feasible to extract the Binary Mask for the shape of the fire. This will be a genuine extraction of the ROI, which is of utmost significance in this sector.

References

[1] WHO, "Wildfires," www.who.int, 2022. [Online]. Available: https://www.who.int/health-topics/wildfires

[2] H. Pan, D. Badawi, and A. E. Cetin, "Computationally efficient wildfire detection method using a deep convolutional network pruned via fourier analysis," *Sensors*, vol. 20, p. 2891, 05 2020.

[3] Z. J. Al-Araji, S. S. Syed Ahmad, and R. S. Abdullah, "Attack prediction to enhance attack path discovery using improved attack graph," *Karbala International Journal of Modern Science*, vol. 8, pp. 313–329, 08 2022.

[4] A. Apostolakis, S. Girtsou, G. Giannopoulos, N. S. Bartsotas, and C. Kontoes, "Estimating next day's forest fire risk via a complete machine learning methodology," *Remote Sensing*, vol. 14, p. 1222, 03 2022.

[5] A. S. Mahdi and S. A. Mahmood, "Wildfire-i," 2022. [Online]. Available: https://www.kaggle.com/datasets/ahmedsaleemmahdi/wildfire

[6] Y. Wang, L. Dang, and J. Ren, "Forest fire image recognition based on convolutional neural network," *Journal of Algorithms & Computational Technology*, vol. 13, p. 1748302619887689, 2019. [Online]. Available: https://doi.org/10.1177/1748302619887689

[7] Z. Hong, Z. Tang, H. Pan, Y. Zhang, Z. Zheng, R. Zhou, Z. Ma, Y. Zhang, Y. Han, J. Wang, and S. Yang, "Active fire detection using a novel convolutional neural network based on himawari-8 satellite images," *Frontiers in Environmental Science*, vol. 10, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenvs.2022.794028

[8] S. T. Seydi, V. Saeidi, B. Kalantar, N. Ueda, and A. A. Halin, "Fire-net: A deep learning framework for active forest fire detection," *hindawi*, 2022. [Online]. Available: https://doi.org/10.1155/2022/8044390

[9] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, 2021. [Online]. Available: https://www.mdpi.com/1999-4907/12/2/217

[10] Y. N. Jurn, S. A. Mahmood, and J. A. Aldhaibani, "Anti-drone system based different technologies: Architecture, threats and challenges," in *2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2021, pp. 114–119.

[11] H. Ul Ain Tahir, A. Waqar, S. Khalid, and S. M. Usman, "Wildfire detection in aerial images using deep learning," pp. 1–7, 2022.

[12] S. G. Zhang, F. Zhang, Y. Ding, and Y. Li, "Swin-yolov5: Research and application of fire and smoke detection algorithm based on yolov5," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–8, 06 2022.

[13] A. S. Mahdi and S. A. Mahmood, "Analysis of deep learning methods for early wildfire detection systems: Review," p. 271–276, 05 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9888515

[14] Z. Xue, H. Lin, and F. Wang, "A small target forest fire detection model based on yolov5 improvement," *Forests*, vol. 13, no. 8, 2022. [Online]. Available: https://www.mdpi.com/1999-4907/13/8/1332

[15] A. S. Mahdi and S. A. Mahmood, "An edge computing environment for early wildfire detection," *Annals of Emerging Technologies in Computing*, vol. 6, pp. 56–68, 7 2022. [Online]. Available: http://aetic.theiaer.org/archive/v6/v6n3/p5.html

[16] R. Reddy, C. Nagaraju, and I. R. S. Reddy, "Canny scale edge detection," *IJETT*, 1 2016.

[17] D. Chu, "Machine-learning-model-api-using-yolov5," 2021. [Online]. Available: https://betterprogramming.pub/machine-learning-model-api-using-yolov5-with-fast-api-192f1290a982

[18] S. Gutta, "Object detection algorithm-yolo v5 architecture," 2021. [Online]. Available: medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef

[19] S. Narkhede, "Understanding confusion matrix," 2018. [Online]. Available: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[20] A. Bhandari, "Everything you should know about confusion matrix for machine learning," 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

[21] M. Hossin and S. M.N, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining Knowledge Management Process*, vol. 5, pp. 01–11, 3 2015.

Ahmed Saleem Mahdi Computer Science Master Student, Ministry of Higher Education and Scientific Research, Baghdad-Iraq, Mustansiriyah University, College of Education, Computer Science Department

Sawsen Abdulhadi Mahmood Professor of Computer Science, Ministry of Higher Education and Scientific Research, Baghdad-Iraq, Mustansiriyah University, College of Education, Computer Science Department