Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 1, 2024, pp. 105-122
DOI: https://doi.org/10.25195/ijci.v50i1.476
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Research Article

INTELLIGENT SURVEILLANCE SYSTEM FOR FIRE DETECTION USING YOLOV8

Muthanna S. Mohammed[1*]
dept. Computer Science
Mustansiriyah University
Baghdad, Iraq
muthana.salih@uomustansiriyah.edu.iq

Amel H. Abbas[2]
dept. Computer Science
Mustansiriyah University
Baghdad, Iraq
dr.amelhussein2017@uomustansiriyah.edu.iq

Nada A.Z. Abdullah[3]
dept. Computer Science
University of Baghdad,
Baghdad, Iraq
nada.abdullah@sc.uobaghdad.edu.iq

ARTICLEINFO
Article History
Received: 19/02/2024
Accepted : 21/03/2024
Published: 08/06/2024

ABSTRACT

This study describes a lightweight deep learning model trained on a self-made image dataset taken inside farms and open areas of the Holy Shrine of Al-Hussainiya in the City of Karbala, Iraq. This dataset includes fire and smoke images taken using a Samsung A52S camera in different weather conditions. The overall goal is to create a fire detection system model that can successfully replace the existing physical sensor-based fire detectors and lessen the issues that come with such fire detectors, including false and delayed triggering. Another goal is to control fires on farms or open areas and prevent crop damage as much as possible. Previous studies were reviewed. Moreover, the architecture of the You Only Look Once version 8 (YOLOv8) model was briefly explained, and the results it achieved were compared with those achieved by previous versions. Then, the proposed system was trained and evaluated with the YOLOv8 large model. Results showed that the proposed system outperformed the rest of the current systems in mAP, which reached 98.5%.

Keywords:* *Fire Detection System; CNNs; Computer Vision; and Deep Learning.

1. INTRODUCTION

As the frequency of unintentional fires in open places continues to increase alarmingly, fire detection systems have become a vital component of the necessary accessories for structures and open areas. In today's world, an accidental fire is the most common disaster. Human mistakes, climate change, natural disasters, and system failure are the main reasons for fires, which endanger people and their possessions. According to the figures, the wildfire disaster was responsible for the overall damages totaling USD 3.1 billion in 2015. In addition, fires and other natural disasters decimate 10,000 km2 of vegetation annually in Europe [1].

Several solutions for detecting flames in their early stages and limiting the amount of property destruction they cause have been proposed to reduce the occurrence of accidental fires. In addition to the challenge of early fire detection, the current fire alarm systems are useless regarding false activations. A critical component of intelligent surveillance

**Corresponding author. Email: muthana.salih@uomustansiriyah.edu.iq*

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

systems is the fire detection system. As part of an early warning system, the fire detection system monitors diverse settings, such as open spaces, including farms and public parks, to report, ideally, the start of a fire. Most fire detection systems currently employ point sensors for smoke analysis, temperature sampling, and particle sampling; they are not useful for outdoor surveillance and can identify fire probability slowly, generally within minutes [2].

Smoke and heat detectors turn on when enough smoke particles reach the device and the temperature increases substantially to avoid false alarms. Furthermore, timing is an important factor in minimizing fire damage. Detecting systems' response time should be decreased to increase the possibility of putting out fires and reduce financial and humanistic losses [3].

The research problem presents several questions, including how a self-made dataset (SDS) is prepared to enable the proposed system to be trained and obtain the best real-time results for fire detection and how an intelligent system that monitors open areas and detects fires and smoke with high precision is built. The intelligent fire detection surveillance system is used to prevent or limit the damage resulting from the occurrence of fire or smoke in open areas. The present work aims to apply the system according to the datasets trained with the proposed system to monitor and detect fires and smoke in real-time with high accuracy when responding to the moment of fire.

The following sections make up the rest of this paper: Section II addresses the literature review. Section III outlines the theoretical part. Section IV introduces the proposed system. Section V presents the implementation and results. Section VI concludes this paper with suggestions for further research.

2. RELATED WORK

Fire detection systems may be designed for various environments. Given the distinct demands of each location, several fire detection techniques may be utilized in different situations. Certain devices can detect fires in outdoor settings, such as open regions and woods, whereas others are designed for inside environments, such as buildings. This section elucidates the sophisticated fire and smoke detection systems employed in forests and several other environments, as extensively examined in multiple studies. For example, Tao et al. [4] proposed the use of convolutional neural networks (CNNs), which can be trained to process raw pixel values and generate classifier outputs. CNNs can automatically extract features from pictures. Research findings demonstrate that a detection rate mean accuracy of 99.4% and a false alarm rate of 0.44% can be achieved by employing an extensive dataset including four meticulously gathered sets of smoke and non-smoke data from cameras or the Internet.

Jivitesh Sharma et al. [5] developed a fire detection system by employing two deep CNNs, VGG16, and ResNet50. The enhanced VGG16 and ResNet50 models included fully linked layers and were tested on an imbalanced dataset. The results show that adding completely connected layers improves the detector models' accuracy, exceeding 90%; however, this approach significantly increases training time.

Khan Muhammad et al. [6] used a model that can assess 17 frames per second, which is sufficient to identify fire at an early stage. They also used cameras working at 25–30 frames per second. They initially used AlexNet as a basic architecture and then changed it based on the circumstances, with the consideration of accuracy and complexity. They improved fire detection accuracy while reducing false alarms in various indoor and outdoor environments.

Dongqing Shen et al. [7] proposed the most effective method for flame detection. It is in charge of the object that falls inside the cell. This method was determined by comparing the You Only Look Once (YOLO) version 1 (YOLOv1) model with shallow learning techniques. Without a completely connected layer, each cell forecasts the bounding box and the confidence score of this box using nine convolutional layers and an extra layer for pretraining. Then, the network for determining the four parameters is added. They created 1720 photographs of fire flames for the training dataset, treating each of the 172 images as a single unit. Then, 20 and 40 epochs were trained for the pretraining phase and the main training phase, respectively. The suggested flame recognition accuracy increased to 76% after Google's TensorFlow framework was adopted.

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Arpit Jadon et al. [2] developed FireNet, which had to be built from scratch. The results demonstrated that FireNet can successfully offer a real-time fire detection function at up to 24 frames per second. This rate is nearly as quick as human visual cognition. FireNet consists of three convolutional layers and four dense layers. It works well for real-time fire detection applications.

William Thomson et al. [8] investigated several CNN architectures and modifications for detecting fire pixel boundaries in real-time, nontemporal videos or still pictures. Two reduced-complexity CNN designs, namely, NasNet-A-OnFire and ShuffleNetV2-OnFire, were utilized to improve computational performance. Their superpixel localization accuracy is 97%, whereas their full-frame binary classification accuracy is 95%.

Hongwen Du et al. [9] demonstrated an improved high-speed flame detection system using YOLO version 7 (YOLOv7). This flame detection method is lighter and more effective at extracting flame characteristics than the YOLOv7 method. A total of 2059 open-flame datasets labeled with single-flame categories were used to minimize the impact of high-quality datasets. The image input size was 640 × 640. Thus, the comparative experimental effect was entirely dependent on the performance of the flame detection method. The findings show that compared with the YOLOv7 method, the technique increases accuracy by 5% and the mean average precision (mAP) by 2.1%. F. M. Talaat and H. ZainEldin [10] presented the smart fire detection system (SFDS) as an enhanced fire detection solution for smart cities based on the YOLO version 8 (YOLOv8) model. SFDS uses deep learning (DL) to detect fire-specific characteristics in real time. Compared with traditional fire detection methods, the SFDS technique can enhance detection accuracy, minimize false alarms, and improve cost-effectiveness.

This technique may be expanded to detect additional smart city issues of interest, such as gas leaks or floods. The dataset collected from websites includes 26,520 images with flames, smoke, fire, and smoke, as well as regular scenes without fire or smoke. The dataset is partitioned into 21,216 segments for training and 5304 segments for testing. They trained the model for 300 epochs with a batch size of 16 and an initial learning rate of 0.01. Finally, SFDS exhibits cutting-edge performance in terms of precision and recall, with a precision rate of 97.1% across all classes and an overall mAP of 97.5% for the two classes. The related study by F. M. Talaat and H. ZainEldin is the closest work to the proposed model.

3. DEEP LEARNING FOR FIRE DETECTION

CNN is a DL technique that considers the visual processes of living things. It is extensively used because studies show that it can extract precise feature maps from raw images; thus, it produces positive results, such as motion detection, image classification, and object recognition on pictures [11],[12].

In general, any model in DL, such as the MobileNet v2 model, consists of three layers, namely, convolution, pooling, and fully linked convolutional layers [13]. The convolution layer extracts the feature, the pooling layer reduces the size of the extracted feature maps, and the flattened layer flattens image properties that have undergone many stages of convolution and pooling before being delivered to the fully connected layer. This layer is constructed in a manner similar to the construction of a neural network and is used for classification [11].

The convolutional layer is the core component of CNNs. Unlike other neural networks, which utilize weighted summation and connection weights, the convolutional layer uses image transform filters known as convolution kernels to build feature maps from raw pictures. The convolutional layer has several convolution kernels. The feature map is created by adding the weighted sum of the pixels that the convolution kernel floats over to compute a new pixel as it goes across images. The input, hidden, and output layers of CNN are illustrated in Fig. 1 [14].

CNNs are highly recommended because they automatically learn the unique features of fires. However, each CNN model differs depending on the method of implementation. As a result, the performance varies. These models include Faster R-CNN [15], EfficientDet [16], Single Shot MultiBox Detector [17], RetinaNet [18], MobileNetV2 [19], and all versions of YOLO from YOLOv1 to the last version of the YOLOv8 model.

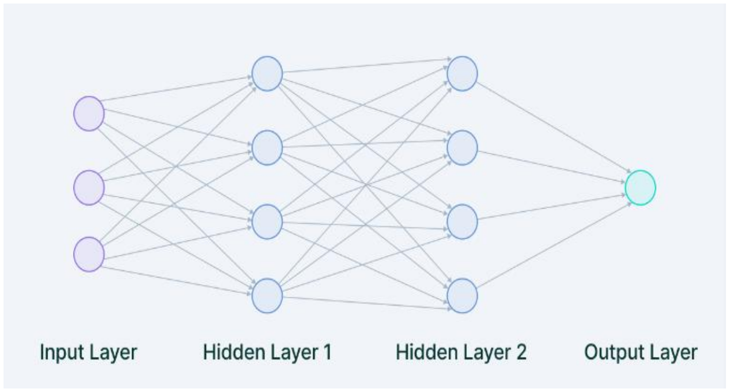


Fig. 1. General layers of CNNs [14]

3.1 YOLOv8 Architecture

Ultralytics, the firm behind YOLOv5, published YOLOv8 [20] in January 2023. YOLOv8 offers five scaled versions: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large). YOLOv8 is a cutting-edge, state-of-the-art model that builds on the success of earlier YOLO versions and offers new features and improvements to boost performance and versatility further. This adaptability enables researchers to exploit YOLOv8's capabilities across a wide range of applications and domains. Fig. 2 depicts YOLOv8's intricate architecture.

YOLOv8 has a similar backbone to YOLOv5, with minor changes to the CSPLayer, which is now known as the C2f module. The C2f module (cross-stage partial bottleneck with two convolutions) increases detection accuracy by integrating high-level characteristics with contextual information. YOLOv8 processes objectness, classification, and regression tasks independently using an anchor-free model with a decoupled head [21]. This architecture allows each branch to concentrate on its job while improving the model's overall accuracy.

The sigmoid function is utilized as the activation function for the objectness score in the YOLOv8 output layer, expressing the chance that the bounding box includes an object. It uses the softmax function to express the probability of items belonging to each conceivable class. For bounding box loss, YOLOv8 employs the complete IoU and distribution focal loss (DFL) [22] functions and binary cross-entropy for classification loss. These losses enhance object identification performance, particularly in tiny objects. In general, YOLOv8 architecture is divided into two parts: the first is the backbone, and the second is the head.

The backbone is a series of convolutional layers that extract relevant features from the input image. It consists of five conv layers, four modules of C2f, and one module of SPPF. Moreover, the three heads in YOLOv8 correspond to the detect module's cv2, cv3, and DFL layers.

3.2 Dataset Used with YOLOv8

Several prominent detection datasets have been made available over the last decade. These datasets include the Microsoft Common Objects in Context (MS COCO) detection [23], PASCAL Visual Object Classes (VOC) (such as VOC2007 and VOC2012) [24], Scene Understanding [25], and ImageNet Large Scale Visual Recognition Challenge (such as ILSVRC2014) datasets [26].

The MS COCO dataset is commonly used for training and evaluating DL models for object detection and the last releases of YOLO, particularly the YOLOv8 model.

Finally, YOLOv8 is evaluated on the MS COCO dataset test-dev 2017. YOLOv8x obtains an AP of 53.9%, with a 640-pixel picture size (compared with 50.7% for YOLOv5 on the same input size) at a speed of 280 FPS on NVIDIA A100 and TensorRT [20], as shown in Table I [27].

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

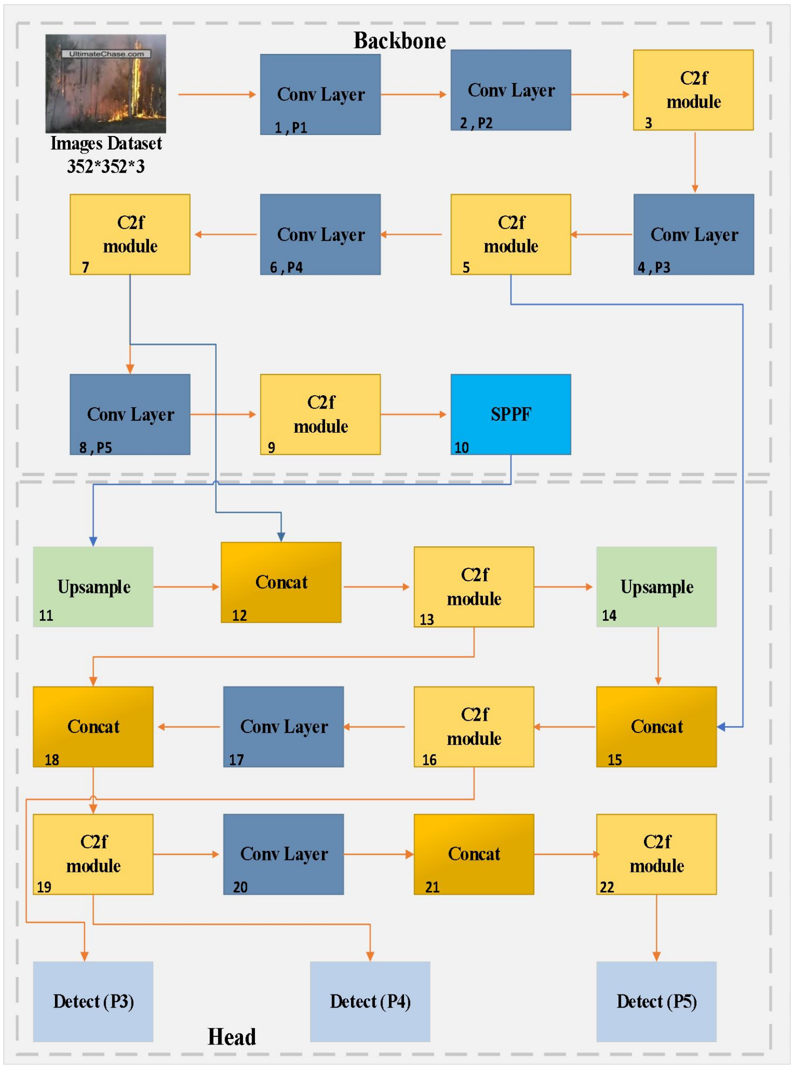


Fig. 2. YOLOv8's architecture [28]

The mAP values are for the single model and single scale on the MS COCO val2017 dataset, and speed is averaged over MS COCO Val images using an Amazon EC2 P4d instance. YOLOv8 can be fine-tuned using customized datasets to improve its accuracy for certain object detection tasks. Fig. 3 shows the efficiency of YOLOv8 [29], in object detection, particularly in real-time. YOLOv8 is faster and smaller than other versions. Its efficiency is the real reason why the author uses the YOLOv8 model.

TABLE I. YOLOV8'S FIVE VERSIONS TRAINED ON THE MS COCO DATASET [27]

Model	Size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (MS)	Speed A100 TensorRT (MS)	Parameters (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

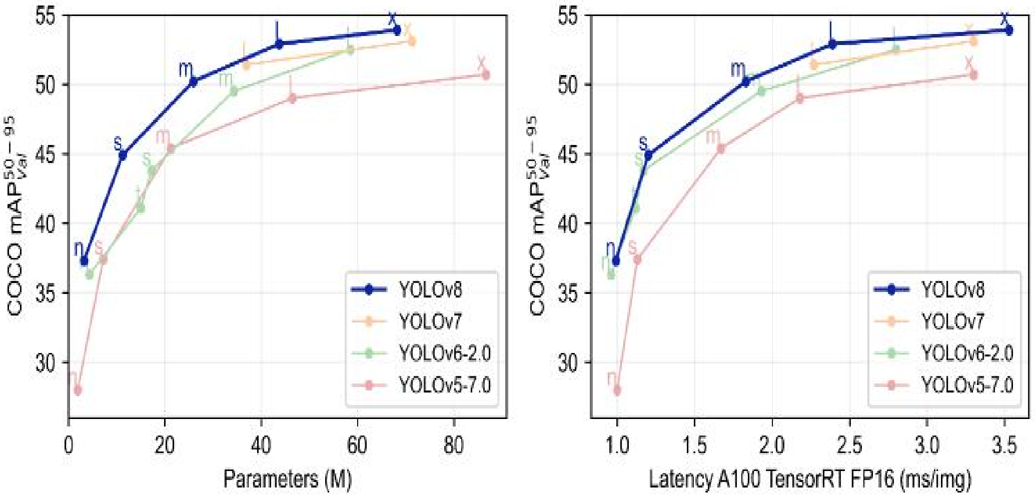


Fig. 3. Comparison of YOLOv8 and previous versions of YOLO in terms of efficiency [29]

4. THE PROPOSED SYSTEM

The general approach layout of the proposed system is shown in Fig. 4. After the data collection and preparation from the video of the SDS, the proposed system consists of five phases: image preprocessing, model training, model evaluation, model deployment, and performance metrics.

4.1 Image Dataset

The author’s SDS was created from scratch. Five videos were captured inside the farms and open areas of the Holy Shrine of Al-Hussainiya in the City of Karbala, Iraq. They include fire and smoke images taken using a Samsung A52S camera in different weather conditions. Some images were randomly selected from each video of the dataset, including the images taken for the proposed model. Fig. 5 exhibits the samples of the image’s dataset of SDS. After obtaining the official approval from the manager of Imam Hussainiya Shrine, the author, in cooperation with staff members who helped him, burned pieces of wood and palm fronds that had been collected in advance to obtain the smoke and flame. Then, the author captured videos of various climatic conditions during the day and at night to prepare the image dataset.

The complete and final SDS used to train and test the proposed CNN model contains 6,950 images, including fire images, smoke images, and normal images. This dataset was split into a training subset (70%), a validation subset (20%), and a test subset (10%). This distribution is considered the best based on the author’s view. In particular, the author tried to change the distribution of images in the training, verification, and testing subsets, and the results of mAP and accuracy remained unsatisfactory after the proposed model was trained. For example, the author split the dataset into three parts: 60% for training, 30% for validation, and 10% for testing. After the proposed system was trained, the mAP and accuracy decreased by approximately 5% compared with those obtained from the current percentages used and accepted. Likewise, when the percentages for the training, validation, and testing subsets were changed to 80%, 15%, and 5%, respectively, the final results for accuracy and mAP decreased by approximately 7%. Table II presents the details of descriptive images in SDS.

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

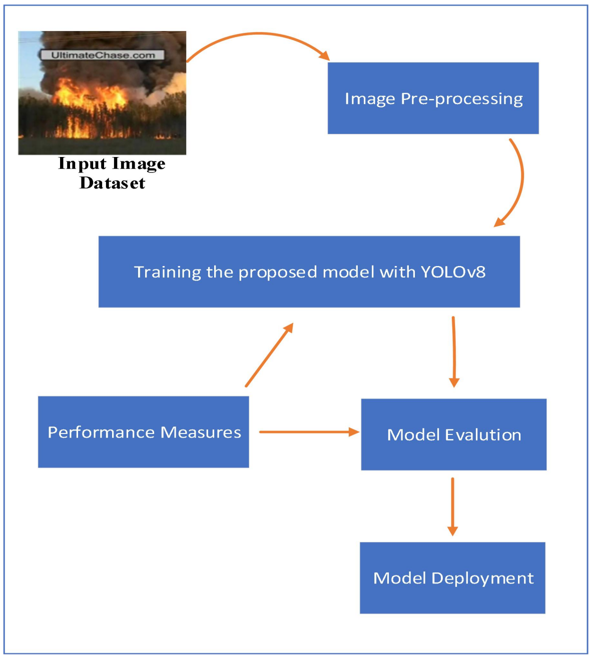


Fig. 4. General view of the proposed system

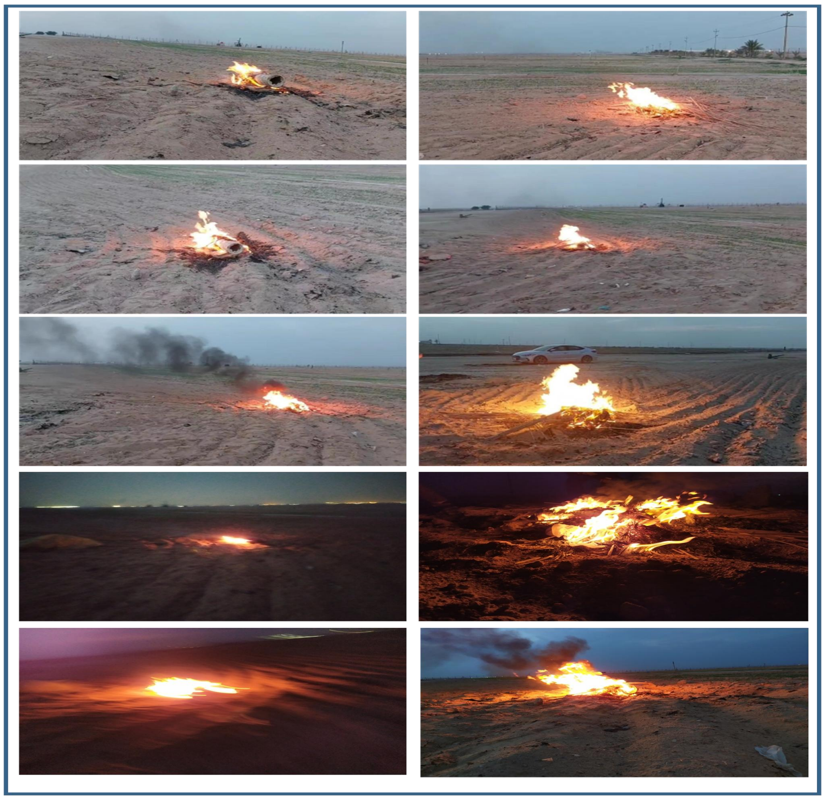


Fig. 5. Image samples captured from Al-Hussainiya Holy Shrine farm.

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

TABLE II. DESCRIPTIVE IMAGES OF SDS

Parameters	Training subset	Validation subset	Testing subset
Number of normal images	64	18	8
Number of fire images	4,794	1,359	698
Number of smoke images	705	181	124
Total	5,563	1,558	832

The annotations for each frame allow the manual use of the Roboflow platform [30] available on the website. Fig. 6 shows a general view of SDS with annotations on the Roboflow platform. The total number of annotated images for two classes of SDS is 7,863 with null examples of 90 images, as shown in Table III.

Finally, all images in the dataset were randomly shoveled and annotated manually. The images of fire and smoke were annotated as “fire” and “smoke,” respectively, by using the Roboflow platform. The samples of the annotated images from the training, validation, and testing datasets are shown in Fig. 7.

TABLE III. ANNOTATED IMAGES OF SDS.

Parameters	Images
Number of fire-annotated images	6,851
Number of smoke-annotated images	1,012
Total	7,863

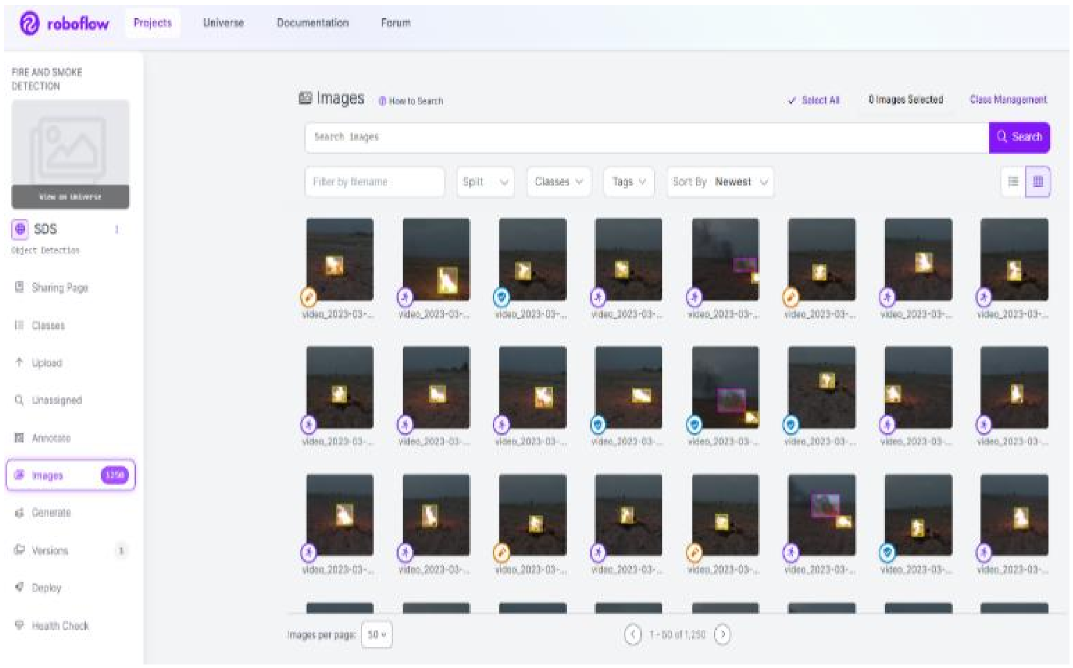


Fig. 6. SDS with annotations on the Roboflow platform.

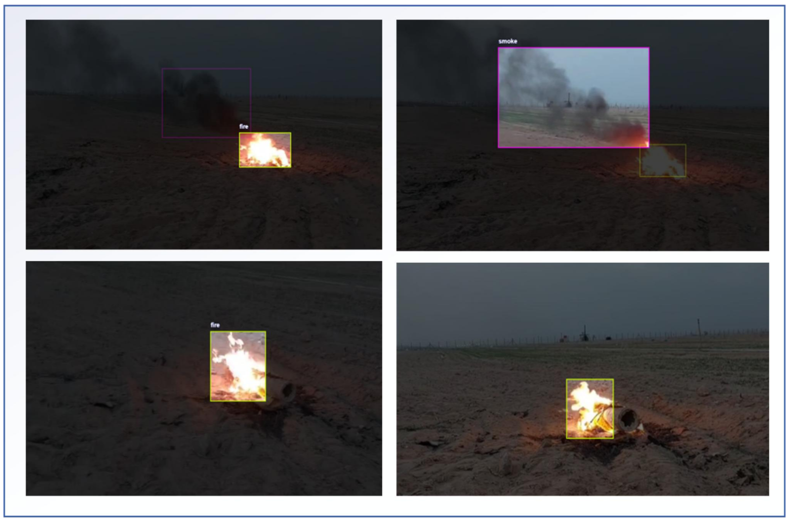


Fig. 7. Sample of annotated images in SDS.

4.2 Image Preprocessing

The image preprocessing goes through several steps before inputting the images into the proposed system. These phases are shown in Fig. 8. The colors are depicted based on three dimensions: hue, saturation, and value. Brightness is another name for the HSV or HSB color model. Afterward, the bilinear interpolation resize image method was used to retain compatibility with the pretrained proposed system.

The hue of a color is its location on the color wheel and is measured in degrees. In the proposed system, the colors of the image were adjusted between −15 and 15 degrees of hue. The saturation level used in the proposed system to adjust the color vibrancy of the images is 30%. The brightness used in the proposed system is 20%. The retrieved frames were cropped to 352 × 352 to retain compatibility with the pretrained model because the author took the median image ratio by using the bilinear interpolation resize image method in the proposed system.

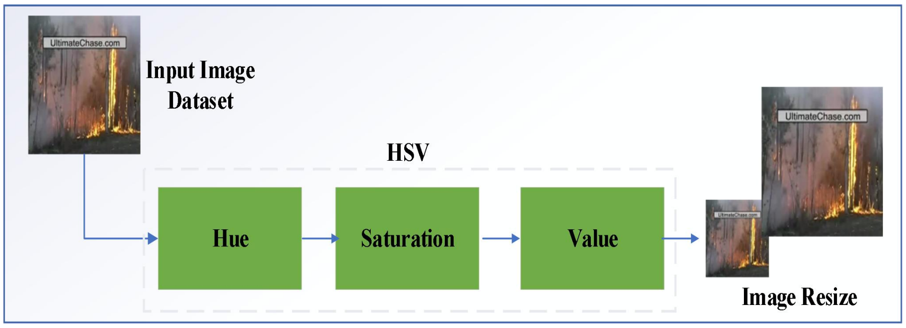


Fig. 8. Image preprocessing components.

4.3 Model Training

The training phase includes training the YOLOv8 model with SDS after the video frames were converted into images with annotation. The dataset image size of 352×352 was fed into the backbone part of the proposed system sequentially. In the backbone, conv1 layer is the first component; conv2, conv3, conv4, and conv5 layers are next, with the C2F module following each layer. The last module is SPPF, which continues to the last part of the system training, which is detection. Each component in the backbone of the proposed system is repeated several times,

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

depending on the number of parameters and arguments used, which were previously determined in each layer. The flow diagram for the training of the proposed system is shown in Fig. 9.

The proposed system was trained from scratch and fine-tuned based on the YOLOv8 architecture. The model layers used in the proposed system are presented in Table IV. Detailed information about each layer and parameter in the proposed system is also shown. All layers and parameters in the proposed system are equal to those in the YOLOv8 model.

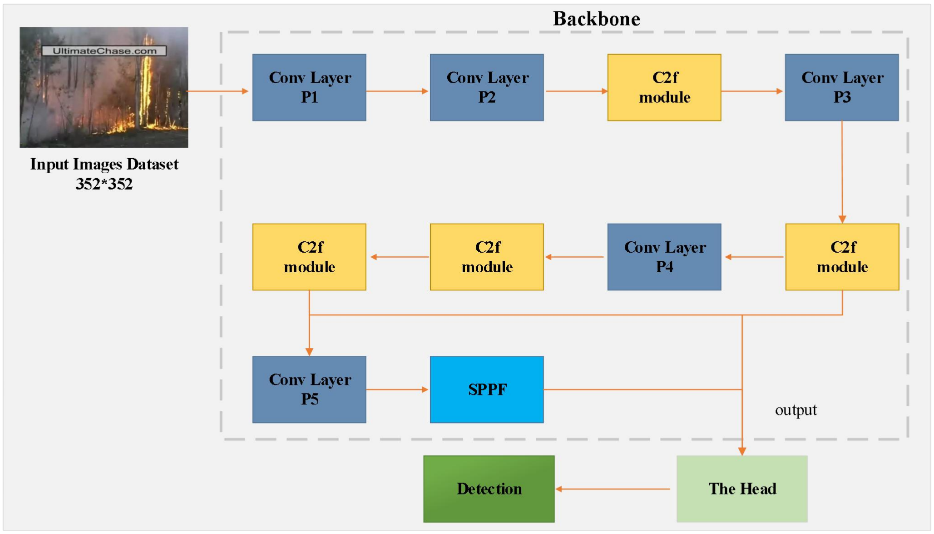


Fig. 9. Flow diagram for the training of the proposed system.

TABLE IV. THE PROPOSED SYSTEM LAYERS AND TRAINABLE PARAMETERS

Layer (Module)	Repeats	Parameters	Argument
Conv	1	1856	[3, 64, 3, 2]
Conv	1	73984	[64, 128, 3, 2]
C2f	3	279808	[128, 128, 3, True]
Conv	1	295424	[128, 256, 3, 2]
C2f	6	2101248	[256, 256, 6, True]
Conv	1	1180672	[256, 512, 3, 2]
C2f	6	8396800	[512, 512, 6, True]
Conv	1	2360320	[512, 512, 3, 2]
C2f	3	4461568	[512, 512, 3, True]
SPPF	1	656896	[512, 512, 5]
Upsample	1	0	[None, 2, 'nearest']
Concat	1	0	[1]
C2f	3	4723712	[1024, 512, 3]
Upsample	1	0	[None, 2, 'nearest']
Concat	1	0	[1]
C2f	3	1247744	[768, 256, 3]
Conv	1	590336	[256, 256, 3, 2]

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Concat	1	0	[1]
C2f	3	4592640	[768, 512, 3]
Conv	1	2360320	[512, 512, 3, 2]
Concat	1	0	[1]
C2f	3	4723712	[1024, 512, 3]
Detect	1	5584342	[2, 256, 512, 512]

After the training of the proposed model, the summary of the proposed system is 356 layers, and the total parameters are 43631382, with 43631366 gradients and 165.4 GFLOPs.

The argument values represent the output of the conv layers and the input to the next layer. Thus, the output of the next layer represents the input to the next layer until the best values with the highest matching ratio in the classification process are obtained. Table V shows that the author input the best values of the arguments used to train the proposed system and obtain the best results in mAP and accuracy after many changes in the values of arguments.

4.4 Model Evaluation and Deployment

After the training of the proposed system, the model was assessed on a validation set in this mode to determine its accuracy and generalization performance. This mode may fine-tune the model's hyperparameters to increase its performance. Some common YOLO validation configurations include batch size, the frequency with which validation is performed during training, and the metrics used to evaluate the model's performance. Carefully tuning and experimenting with these settings are crucial to ensuring that the model performs well on the validation dataset. They are also important for the detection and prevention of overfitting. Algorithm 1 describes that the phases of the proposed system were evaluated during the training of the proposed model with SDS.

TABLE V. ARGUMENTS USED TO TRAIN THE PROPOSED SYSTEM.

Argument	**Value**	**Description**
model	yolov8l.yaml	path to the model file as yaml
data	SDS.yaml	path to dataset file
epochs	200	number of epochs
batch	16	number of images per batch
imgsz	352	size of input images
device	0	device to run on CUDA
workers	8	number of worker threads
optimizer	SGD	optimizer to use
lr0	0.01	initial learning rate
lrf	0.01	final learning rate
momentum	0.937	SGD momentum
weight_decay	0.0005	optimizer weight decay
val	True	validate/test during training

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Algorithm 1: Evaluation of the proposed system

```
Input:    Trained model
Output:   An image with accuracy
Steps:
0         Load the proposed model
1         Run inference on 'image.jpg'
2         Evaluate the model's performance on the
          validation set
3         Results = model('image.jpg')
4         Show the results
5         for r in results:
              plot a BGR numpy array of predictions
              RGB PIL image
              show image
              save the image with the labeling of fire or
          smoke if found
6         END
```

4.5 Performance Metrics

Various metrics were implemented into the system. All of which are intended for use in evaluating the performance of the system. Equation (1) can be used to compute recall [31].

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (1)$$

where TP denotes true positives, and FN denotes false negatives.

Equation (2) can be used to compute precision [31].

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (2)$$

where FP denotes false positives.

In Equation (3), mAP is calculated by calculating the average precision (AP) for each class and averaging it over many classes [32].

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} APi \qquad (3)$$

where N denotes the number of classes, and APi denotes the AP of class i.

FM is determined using Equation (4), which considers precision and recall [32].

$$\text{F1} - \text{score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \qquad (4)$$

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

5. IMPLEMENTATION AND RESULTS

The proposed model is implemented using Python programming language and the Anaconda environment. Jupyter Notebook version 6.5.2 uses a system with the processor AMD Ryzen 5 3600 6-Core Processor 3.60 GHz, Windows 11 Pro 64bit OS, NVIDIA GeForce RTX 2060, 6144 MiB, and 32 GB DDR4 RAM. The author used a batch size of 16 to train the proposed model on the image dataset. This batch size is better than an 8 or 32 batch size in terms of detection accuracy. The model was trained for 200 epochs because when the number of epochs was increased to 250, the results did not change in terms of accuracy and mAP. Moreover, no overfitting was observed. An initial and final learning rate of 0.01 was used to find the best learning rate for the proposed system. When the learning rate surpassed the mentioned values, the accuracy of the system decreased significantly.

5.1 Results of the Training of the Proposed Model

The results of the training of the proposed model with SDS show that fire images can be found, with 99.9% precision and 100% recall, whereas smoke images were found with 97.8% precision and 96% recall. It has an mAP of 98.5% for both classes when the last epoch is finished in 2.624 hours. The lightweight pretrained file is 87.6 MB. Table VI shows the results after training the proposed model. Fig. 10 depicts the overall training outcomes of the proposed model, including loss, precision, recall, and other metrics. Figs. 11(a), 11(b), 11(c), and (d), depict precision, recall, precision–recall, and F1 curves, respectively. Furthermore, Figs. 12(a) and 12(b) show the results of labels and label correlograms.

TABLE VI. RESULTS AFTER TRAINING THE PROPOSED SYSTEM

Class	Precision	Recall	mAP
Fire	0.999	1	0.994
Smoke	0.978	0.96	0.976
All	0.988	0.98	0.985

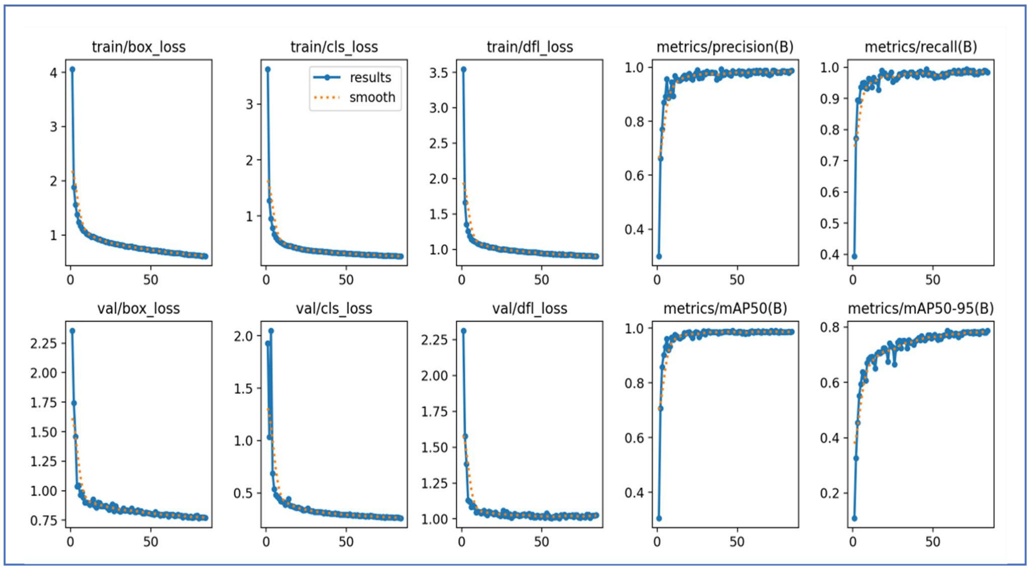


Fig. 10. Results of the training of the proposed model

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

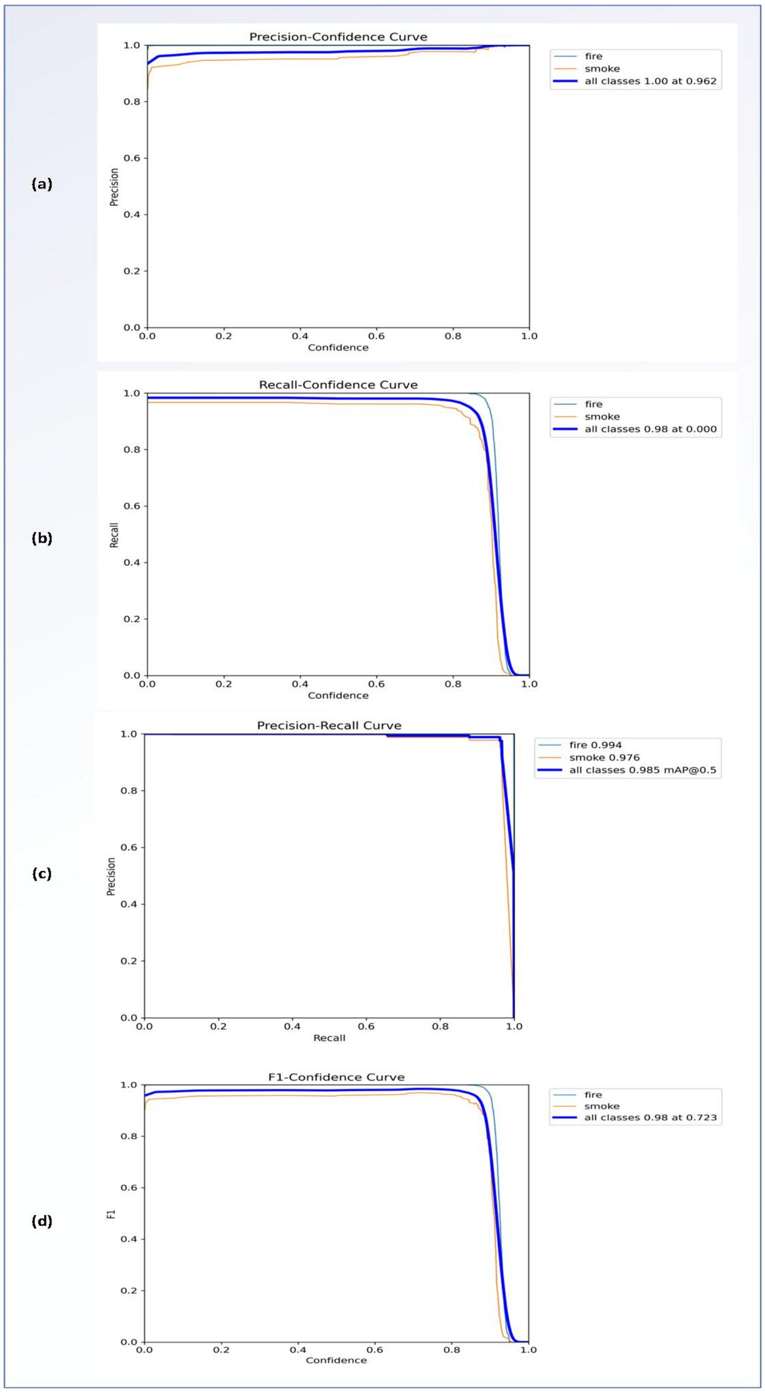


Fig. 11. Confidence curves of the proposed model with SDS: (a) precision confidence curve, (b) recall confidence curve, (c) precision–recall confidence curve, and (d) F1 confidence curve.

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

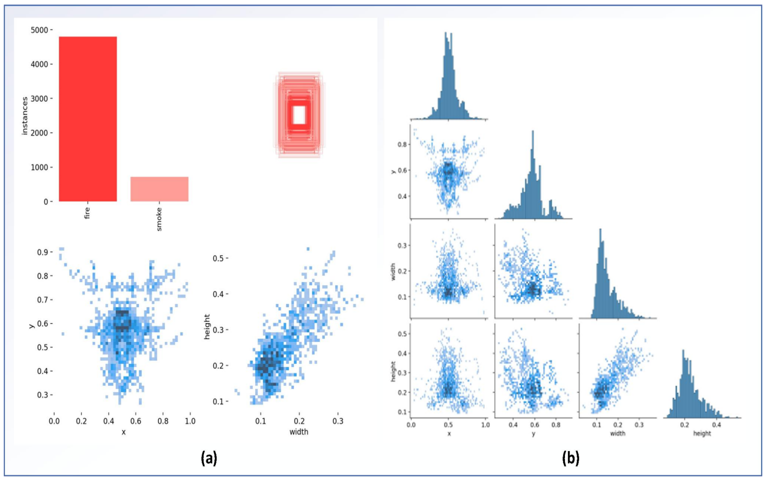
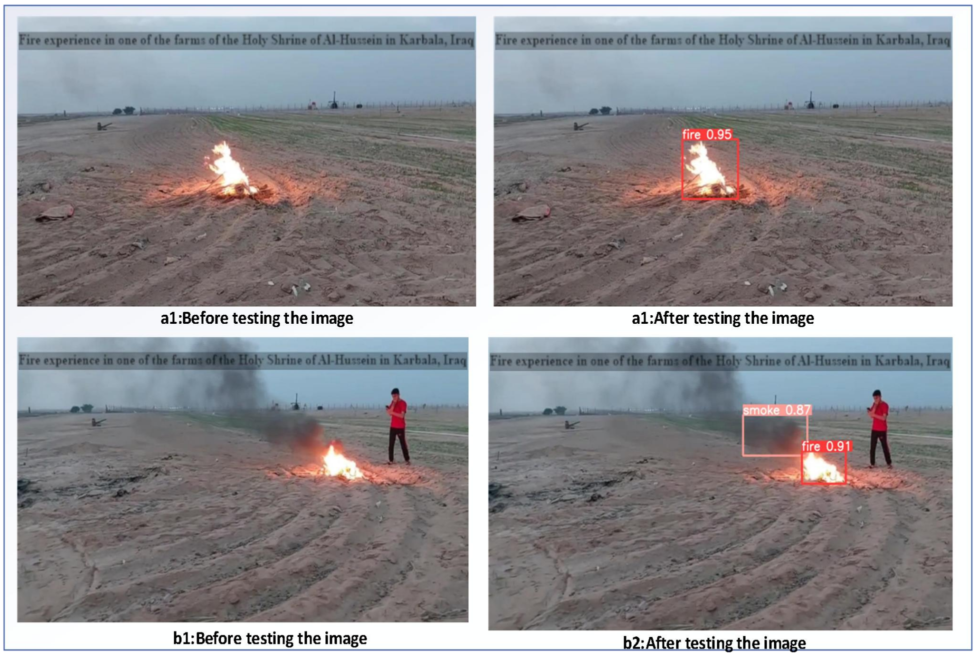


Fig. 12. Results of the (a) labels and (b) label correlograms of the proposed model with SDS.

5.2 Results of the Evaluation of the Proposed Model

After the test on a sample of images was conducted, the evaluation results were obtained, as shown in Fig. 13. The result of test image a1 is the presence of the fire class, which is identified and annotated within its frame. The results of the model's detection accuracy on the testing data in image a2 are 95% fire. Moreover, the preprocessing, inference, and postprocessing speeds are 1, 43, and 2 MS, respectively, per image at various shapes (1, 3, 224, 352). Moreover, image b2 in Fig. 13 shows that the detection accuracy of the model on the testing data is 92% and 87% in the presence of fire and smoke, respectively. It was not assigned to the color red worn by the person standing near the flames of the fire. This was considered a background or normal image. The result of the test is completed in 49.0 MS, and the preprocessing, inference, and postprocessing speeds are 1.0, 49.0, and 2.0 MS, respectively, per image at various shapes (1, 3, 224, 352).


Fire experience in one of the farms of the Holy Shrine of Al-Hussein in Karbala, Iraq
fire 0.95
smoke 0.87
fire 0.91
a1:Before testing the image
a1:After testing the image
b1:Before testing the image
b2:After testing the image
Fig. 13. Result of the proposed model for fire and smoke detection

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Table VII illustrates the performance of the proposed model compared with that of previous works. The proposed system outperforms the previous systems in precision, recall, F1 score, and mAP, indicating its ability to identify fires in real-world conditions.

TABLE VII. COMPARATIVE PERFORMANCE OF THE PROPOSED SYSTEM WITH THE RELATED WORKS

Works	**Models**	**Precision %**	**Recall %**	**F1 Score**	**mAP %**
[33]	YOLOv5	94.99	78.28	0.858	94
[34]	YOLOv6	93.48	28.29	0.9	93.40
[9]	YOLOv7				69.7
[10]	YOLOv8	97.5	95.7	0.962	97.5
Proposed system	YOLOv8	98.8	98.3	0.99	98.6

6. CONCLUSIONS AND FUTURE WORK

This study describes a lightweight deep learning model trained on an SDS captured using a mobile camera. The overall goal is to create a fire detection system that can replace the existing sensor-based fire detectors and reduce the problems that come with erroneous and delayed operation using these fire detectors. In addition to controlling fires in open areas and not harming crops as much as possible, reducing the level of environmental pollution in the country is endeavored. The author captured various video clips within the farms of the Holy Hussein Shrine and preprocessed the images by converting the frames of the video clips into images using the Roboflow platform. Then, each image was manually labeled by specifying a frame for the fire and smoke scenes within the image, applying HSV to the images, and resizing the images to a resolution of 352 × 352. The author trained the proposed system with SDS using the YOLOv8 model and a set of parameters and arguments. The proposed system outperforms the previous systems. The proposed model obtains a precision of 98.8%, a recall of 98.3%, an F-score of 99%, and a mAP of 0.986. The proposed system was trained in two classes: fire and smoke. By contrast, most modern systems relied only on fire or smoke training dataset alone. As a potential future direction, the architecture of the YOLOv8 model will undergo changes to improve its performance at finding fires in open areas. This approach will also include adding other kinds of images in open areas with different types of fire and smoke scenes to improve the system's accuracy.

References

[1] K. Muhammad, S. Khan, M. Elhoseny, S. Hassan Ahmed, and S. Wook Baik, "Efficient Fire Detection for Uncertain Surveillance Environment," *IEEE Transactions on Industrial Informatics,* vol. 15, no. 5, pp. 3113-3122, 2019, doi: 10.1109/tii.2019.2897594.

[2] A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "FireNet A Specialized Lightweight Fire & Smoke Detection Model for Real-Time IoT Applications," *IEEE, Software Research Institute,* no. 1905.11922v2, pp. 1-6, 2019, doi: 10.48550/arXiv.1905.11922.

[3] A. Ayala, B. Fernandes, F. Cruz, D. Macêdo, A. Oliveira, and C. Zanchettin, "KutralNet A Portable Deep Learning Model for Fire Recognition," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, July 2020: IEEE, doi: 10.1109/IJCNN48605.2020.9207202.

[4] C. Tao, J. Zhang, and P. Wang, "Smoke Detection Based on Deep Convolutional Neural Networks," *International Conference on Industrial Informatics - Computing Technology, Intelligent Technology,*

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Industrial Information Integration (ICIICII), pp. 150-153, 2016.
[5] J. Sharma, O.-C. Granmo, M. Goodwin, and J. Fidje, "Deep Convolutional Neural Networks for Fire Detection in Images," *Communications in Computer and Information Science,* 2017.
[6] K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," *Neurocomputing,* vol. 288, pp. 30-42, 2018.
[7] 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).* 2018, pp. 416-420.
[8] W. Thomson, N. Bhowmik, and T. P. Breckon, "Efficient and Compact Convolutional Neural Network Architectures for Non-temporal Real-time Fire Detection," presented at the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020.
[9] H. Du, W. Zhu, K. Peng, and W. Li, "Improved High Speed Flame Detection Method Based on YOLOv7," *Open Journal of Applied Sciences,* vol. 12, no. 12, pp. 2004-2018, 2022.
[10] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Computing and Applications,* pp. 1-16, 2023.
[11] Y. S. Taspinar, M. Koklu, and M. Altin, "Fire Detection in Images Using Framework Based on Image Processing, Motion Detection and Convolutional Neural Network," *International Journal of Intelligent Systems and Applications in Engineering,* vol. 9, no. 4, pp. 171-177, 2021.
[12] Y.-D. Zhang *et al.*, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimedia Tools and Applications,* vol. 78, pp. 3613-3632, 2019.
[13] A. Q. Nguyen, H. T. Nguyen, V. C. Tran, H. X. Pham, and J. Pestana, "A Visual Real-time Fire Detection using Single Shot MultiBox Detector for UAV-based Fire Surveillance," presented at the 2020 IEEE Eighth International Conference on Communications and Electronics (ICCE), 2021.
[14] P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," in *Case Studies in Thermal Engineering* vol. 19, ed, 2020, pp. 1-11.
[15] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.
[16] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10781-10790.
[17] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 2016: Springer, pp. 21-37.
[18] Y. Li and F. Ren, "Light-weight retinanet for object detection," *arXiv preprint arXiv:1905.10011,* 2019.
[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520.
[20] G. Jocher, A. Chaurasia, and J. Qiu. "YOLO by Ultralytics." Ultralytics. https://github.com/ultralytics/ultralytics (accessed February 30, 2023).
[21] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. arXiv 2023," *arXiv preprint arXiv:2304.00501*.
[22] X. Li *et al.*, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," *Advances in Neural Information Processing Systems,* vol. 33, pp. 21002-21012, 2020.
[23] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014: Springer, pp. 740-755.
[24] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision,* vol. 111, pp. 98-136, 2015.
[25] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE computer society conference on computer vision and pattern recognition*, 2010: IEEE, pp. 3485-3492.

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 49, No. 2, 2023, pp. 105-8
DOI: https://doi.org/10.25195/ijci.v49i2.417
Print ISSN: 2313-190X, Online ISSN: 2520-4912

[26] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision,* vol. 115, pp. 211-252, 2015.
[27] ultralytics. https://github.com/ultralytics/ultralytics (accessed.
[28] https://github.com/ultralytics/ultralytics/issues/189 (accessed.
[29] A. C. G. Jocher, J. Qiu. "YOLOv8 by MMYOLO." https://github.com/open-mmlab/mmyolo/tree/main/configs/yolov8 (accessed May 13, 2023).
[30] "Roboflow platform." https://app.roboflow.com/fire-and-smoke-detection (accessed.
[31] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila, "Improved precision and recall metric for assessing generative models," *Advances in Neural Information Processing Systems,* vol. 32, 2019.
[32] R. Yacouby and D. Axman, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models," in *Proceedings of the first workshop on evaluation and comparison of NLP systems*, 2020, pp. 79-91.
[33] Z. Wang, L. Wu, T. Li, and P. Shi, "A smoke detection model based on improved YOLOv5," *Mathematics,* vol. 10, no. 7, p. 1190, 2022.
[34] S. Norkobil Saydirasulovich, A. Abdusalomov, M. K. Jamil, R. Nasimov, D. Kozhamzharova, and Y.-I. Cho, "A YOLOv6-based improved fire detection approach for smart city environments," *Sensors,* vol. 23, no. 6, p. 3161, 2023.