SEBHA UNIVERSITY CONFERENCE PROCEEDINGS, VOL.03 NO. 2 2024
DOI: 10.51984/SUCP.V4I2.3342

وقائع مؤتمرات جامعة سبها
Sebha University Conference Proceedings

Confrence Proceeeding homepage: http://www.sebhau.edu.ly/journal/CAS

Development of Real Time System for Smoke and Fire Detection in Wide Areas Using Yolov8

Karima Ahmed & Emsaieb Geepalla

Wadi Al-Shati University, Faculty of Engineering, Brak Al-Shati, Libya

Keywords:
Artificial Intelligence
Detection
Fires
Smoke
wide areas
Real-time
YOLOv8

ABSTRACT

In an era marked by significant advancements in Artificial Intelligence (AI) and its diverse applications across various fields, including machine learning and computer vision, the domain of surveillance systems and safety measures has undergone a profound transformation. Amidst numerous natural disasters, fires emerge as one of the most devastating calamities, necessitating the utilization of AI capabilities to develop intelligent monitoring systems that bolster our defensive efforts against this disaster. This involves the early detection and notification of relevant authorities to prevent irreparable damages. Traditional fire detection devices, while yielding satisfactory results, exhibit diminished effectiveness in open or large areas and lack real-time detection capabilities. In response to these challenges, this study aims to develop an advanced real-time smoke and fire detection system specifically designed for wide deployment. Leveraging the capabilities of the YOLOv8 deep learning model, the study trained the most suitable versions of the proposed model (YOLOv8l, YOLOv8m) with varying hyperparameters on a dataset comprising 9756 images of various smoke and fire scenarios. The results demonstrate the models' capability to accurately detect fires and smoke, achieving commendable average precision rates while maintaining a delicate balance between precision and recall. Specifically, the YOLOv8l model achieved a mean average precision (mAP50) of 85.1% and an F1 score of 80%, while the YOLOv8m model achieved a mAP50 of 86% and an F1 score of 82%. These models exhibit promising results in real-time fire and smoke detection systems, indicating a new era of proactive measures for fire detection and prevention, deployable on unspecified specification surveillance cameras.

تطوير نظام الوقت الحقيقي للكشف عن الدخان والحرائق في المناطق الواسعة باستخدامYOLOv8

كريمة أحمد و امسيب جيب الله

كلية الهندسة، جامعة وادي الشاطئ ، براك الشاطئ ، ليبيا

الكلمات المفتاحية:
الذكاء الاصطناعي
الكشف عن الحرائق والدخان
مناطق واسعة
الوقت الفعلي

الملخص

في عصر تميز بالتقدم الكبير في الذكاء الاصطناعي (AI) وتطبيقاته المتنوعة عبر مختلف المجالات، بما في ذلك التعلم الآلي ورؤية الكمبيوتر، شهد مجال أنظمة المراقبة وتدابير السلامة تحولًا عميقًا. وسط العديد من الكوارث الطبيعية، تظهر الحرائق كواحدة من أكثر الكوارث تدميراً، مما يستلزم استخدام قدرات الذكاء الاصطناعي لتطوير أنظمة مراقبة ذكية تعزز جهودنا الدفاعية ضد هذه الكارثة. وينطوي ذلك على الكشف المبكر وإخطار السلطات المختصة لمنع حدوث أضرار لا يمكن إصلاحها. ومع أن الأجهزة التقليدية للكشف عن الحرائق تسفر عن نتائج مرضية، فإنها تتضاءل فعاليتها في المناطق المفتوحة أو الكبيرة وتفتقر إلى قدرات الكشف في الوقت الحقيقي. استجابة لهذه التحديات، تهدف هذه الدراسة إلى تطوير نظام متقدم في الوقت الفعلي للكشف عن الدخان والحرائق مصمم خصيصًا للنشر على نطاق واسع. من خلال الاستفادة من قدرات نموذج التعلم العميق YOLOv8، دربت الدراسة الإصدارات الأكثر ملاءمة من النموذج المقترح(YOLOv8l ، (YOLOv8m بمقاييس فرط بارامامتر مختلفة على مجموعة بيانات تضم 9756 صورة لسيناريوهات مختلفة للدخان والحريق. توضح النتائج قدرة النماذج على الكشف الدقيق عن الحرائق والدخان، وتحقيق متوسط معدلات دقة جديرة بالثناء مع الحفاظ على توازن دقيق بين الدقة والاستدعاء. على وجه التحديد، حقق النموذج YOLOv8l متوسط دقة (mAp50)بنسبة 85.1٪ ودرجة F1 بنسبة 80٪، بينما حقق النموذج YOLOv8m Map50 بنسبة 86٪ ودرجة

*Corresponding author:
E-mail addresses: ems.geepalla@wau.edu.ly

Article History : Received 15 June 2024 - Received in revised form 21 September 2024 - Accepted 06 October 2024

F1بنسبة 82٪. تُظهر هذه النماذج نتائج واعدة في أنظمة الكشف عن الحرائق والدخان في الوقت الفعلي، مما يشير إلى حقبة جديدة من التدابير الاستباقية للكشف عن الحرائق والوقاية منها، والتي يمكن نشرها على كاميرات مراقبة غير محددة.

1. Introduction

In a world fraught with disasters, fires emerge as one of the most destructive natural calamities, posing serious threats to both lives and properties, spanning from sprawling forests to bustling industrial complexes [1]. Fires, which occur almost daily on roadsides, streets, public establishments, and private properties, exhibit boundless destructive potential [2].

According to the National Fire Protection Association (NFPA), in 2018 alone, NFPA reported 1,318,500 fires, resulting in 3,655 civilian deaths, 15,200 injuries, and property losses totaling $25.6 billion [3]. In 2022, local fire departments responded to 1.5 million fires, resulting in 3,790 civilian deaths, 13,250 injuries, and property damage valued at $18 billion in the United States [4].

Given the grim statistics on fires and firefighting efforts worldwide, it is evident that controlling these outbreaks, often stemming from natural factors like rising temperatures, poses a formidable or even insurmountable challenge in preventing their occurrence. However, proactive measures can significantly mitigate the extensive losses incurred, primarily by drastically reducing response times upon fire detection. This proactive approach proves invaluable in averting the inevitable catastrophic losses, particularly concerning fires in vast areas [5-7].

Throughout generations, indoor environments typically utilize smoke and flame detection devices as part of fire alarm systems. However, these traditional physical sensors have limitations and may not be suitable for large spaces or outdoor locations, often resulting in frequent false alarms. This inefficiency hampers resource allocation and response management [8-10]. On the other hand, some current technologies, such as infrared remote sensing and satellite monitoring, entail infrastructure complexities and high maintenance costs, rendering them less accessible for widespread deployment (non-scalable). Additionally, they have limited effectiveness, especially in detecting small or incipient fires due to inadequate spatial accuracy and susceptibility to weather conditions and cloud cover, thereby limiting their ability to detect fires in their early stages [11, 15].

In recent years, technological advancements, particularly in the fields of artificial intelligence (AI) and deep learning, specifically computer vision, have shifted towards more advanced methods of object detection. These methods have garnered significant interest and produced robust results in various applications, including autonomous driving, object detection and recognition, medical diagnosis, and visual search [8, 11]. Moreover, numerous recent studies have demonstrated the effectiveness of computer vision and deep learning-based methods in fire and smoke detection [16], particularly models from the YOLO series.

For instance, in a study conducted by Wang et al., smoke detection based on an improved version of YOLOv5 with a processed training dataset consisting of 20,000 images showed promising results, achieving a detection performance higher by 4.4% of mAP for the base model, i.e., 87.4%, compared to traditional deep learning algorithms [17]. Another study by BEKKARI & KADI on forest fire detection using YOLOv5 trained the algorithm on a dataset consisting of 1,500 images, demonstrating its superiority over other algorithms with an accuracy of up to 83.51% [18]. Similarly, Avazov et al. emphasized the importance of fire detection and alerting in ship areas for maritime safety, achieving an impressive accuracy rate of 81% using YOLOv7 trained on ship fire images [19]. For the same purpose, ZIYANG et al. proposed a lightweight algorithm for ship fire detection based on YOLOv8n, achieving high accuracy of up to 90% [20].

However, most existing fire detection methods focus solely on fires and often overlook smoke as an early warning indicator of fire. Unlike Wang et al., who used YOLOv5 for smoke detection only, Saponara et al. and BEKKARI & KADI used YOLOv5, respectively, to detect both smoke and fires together, albeit with limitations in the datasets used.

Therefore, this study aims to create a robust and efficient system for real-time detection of smoke and fire using automated surveillance cameras. The system aims to address the challenges of mitigating disasters and reducing extensive losses, as identified in previous studies. To achieve this, the research proposes to develop a precise, cost-effective, highly reliable, and rapidly responsive system that can provide visual information to relevant authorities in real-time. This system is designed to reduce false alarm rates and enhance resource allocation efficiency, making it suitable for various scenarios, particularly wide and open spaces. The research leverages the capabilities of the latest artificial intelligence technology, the YOLOv8 model, to achieve these goals and deliver a comprehensive solution for smoke and fire detection.

The remaining sections of the paper are organized as follows: **Section 2** provides an explanation of the proposed approach, starting from the dataset and culminating in the development of a real-time smoke and fire detection system. A comprehensive explanation of the experimental results is presented in **Section 3**, while **Section 4** discusses the findings. Finally, **Section 5** presents the conclusions and outlines plans for future study.

2. Methodology

2.1 DataSet.

The utilized dataset comprises images and videos depicting fire and smoke occurrences across diverse scenarios (forests, cars, buildings, etc.) and environments (indoor or outdoor), featuring varying lighting conditions (day or night). These data were gathered from sources offering such content, such as the Kaggle platform, the Pixel website, as well as public outlets like social media and news websites.

2.1.1 Data preprocessing.

Data preprocessing is a critical technique used to transform raw data into a standardized and clean dataset before using it to train models to ensure their accuracy and effectiveness. The number of images in the dataset after preprocessing was approximately 3560 images, divided according to Table 1. Figure 1 illustrates a sample of this data.

Table 1: DataSet.

Type	Number of picture
Fire	1000
Smoke	1000
Fire & Smoke	1560

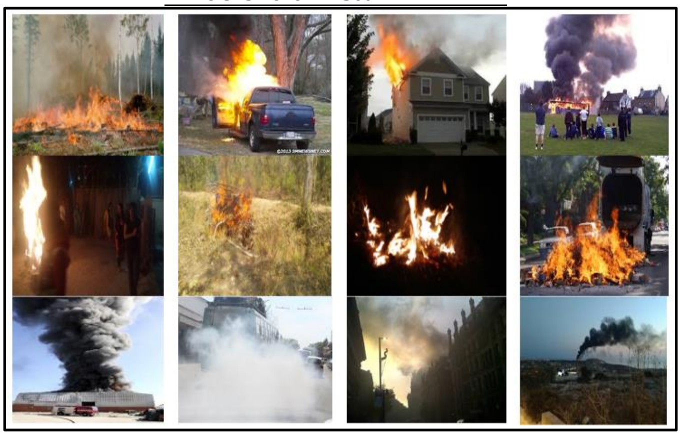

Fig. 1: Sample of a data set.

Preprocessing involves several important and necessary steps after cleaning the data and before starting the training process. These steps include:

a- Image Annotation: It involves manually placing a mark or distinguishing feature on the object (fire and smoke) in the image and assigning the corresponding label to that object.

b- Data Augmentation: It is a process applied to create new data artificially from existing training data. For the dataset in this study, two properties (Flip, 90 Rotate) were applied, resulting in approximately 9756 images in the dataset after augmentation.

c- Resizing: Resizing images is a fundamental step in preprocessing. Uniforl, or medium-sized images facilitate training processes faster and more stably, ensuring smooth image processing and efficient use of computational resources. In this paper, a size of 640x640 was used to train the proposed model.

d- Data Splitting: This step involves splitting the dataset into three subgroups representing training, validation, and testing sets with proportions of 80%, 10%, and 10%, respectively.

These preprocessing steps were smoothly executed using ROBOFLOW.

2.2 The Proposed Model Structure.

In this study, the YOLOv8 model is proposed as the foundation for building a real-time fire and smoke detection system. YOLOv8 is an advanced version of the widely-used YOLO (You Only Look Once) series, released in January 2023 by Ultralytics, a company specializing in computer vision and machine learning. In terms of performance, this model has demonstrated enhanced performance in various applications, including object detection, image classification, and segmentation tasks, showing faster and more accurate results than previous YOLO versions on the standard COCO dataset. YOLOv8 incorporates new architectural improvements and updates, building upon the success of its predecessors, particularly YOLOv5, which is the first version released by Ultralytics [21, 22]. Figure 2 illustrates the basic structure of YOLOv8.

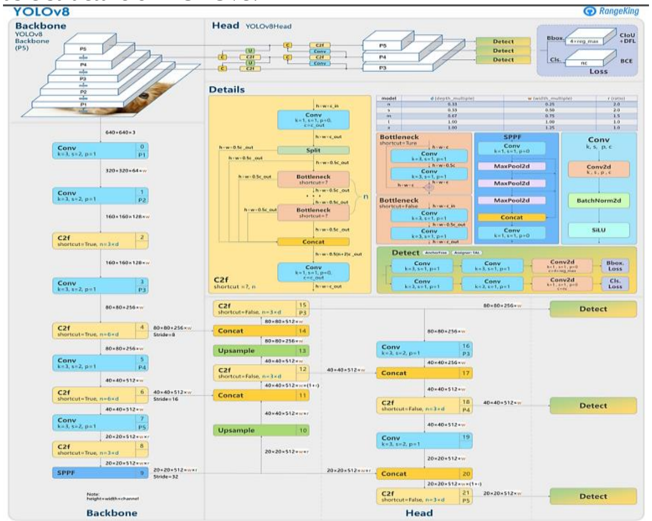

Fig. 2: YOLOv8 architecture[9].

The basic structure of YOLOv8 relies on CSP, a modified version of the Darknet53 backbone. Darknet53 consists of a series of convolutional layers and sampling blocks used to extract features from the input image. The main difference between Darknet53 and CSPDarknet53 is CSP, which utilizes partial cross-stage connections to improve information flow between layers and reduce the number of parameters in the model.

The neck architecture in YOLOv8 is designed to integrate features extracted by the backbone into a single representation for object detection. YOLOv8 utilizes a Feature Pyramid Network (FPN) as the neck architecture, consisting of a top-down pathway to create feature maps at different scales and lateral connections to merge features from the top-down pathway. This approach enables YOLOv8 to detect objects at various scales and aspect ratios [23, 24].

The detection head in YOLOv8 is responsible for predicting the location and class of objects in the input image includes advanced features such as anchor-free detection, predicting the object center directly, and then calculating the object's size by measuring the distance between the object center and its edges. This accelerates the Non-Maximum Suppression (NMS), a complex post-processing step that filters candidate detections after inference, removing low-confidence detections and selecting the highest-confidence ones [9, 23].

One of the distinguishing features of YOLOv8 is its use of a larger and deeper neural network than previous versions, allowing it to capture fine features in images and videos. This architecture has enabled YOLOv8 to achieve high efficiency, accuracy, and speed in object detection, making it the most suitable model for real-time applications such as autonomous vehicles and surveillance systems. Additionally, YOLOv8 introduced a new important feature, an easy-to-use graphical user interface implemented through a Python package [23, 24].

Ultralytics has provided five versions of the YOLOv8 models with various configurations, including mAP accuracy, estimated model parameters in millions, and computational parameters FLOPs, allowing users to choose the model that best suits their needs, ranging from high speed and low accuracy to low speed and high accuracy [9]. Table 2 illustrates the different versions of YOLOv8.

Table 2:different versions of YOLOv8[25].

Model	Size (pixels)	mAP 50-95	Params (M)	FLOPs
YOLOv8n	8.7	3.2	37.3	640
YOLOv8s	28.6	11.2	44.9	640
YOLOv8m	78.9	25.9	50.2	640
YOLOv8l	165.2	43.7	52.9	640
YOLOv8x	257.8	68.2	53.9	640

Since the specified task is considered complex and requires a precise model to identify and extract features from a total of 9755 images, it also necessitates a model with suitable speed to be effective in real-time. Therefore, versions YOLOv8m and YOLOv8l were utilized as they were deemed most suitable for the mentioned task.

2.3 Model Development.

In this stage, the models (YOLOv8m and YOLOv8l) are developed through two main phases: the training phase and the evaluation phase.

2.3.1 Model Training.

Model training is a crucial process aimed at tuning the model parameters (weights) to enable it to produce the desired output based on the available inputs. To adjust these parameters and make the model capable of detecting fires and smoke, the training process begins by training the model on preprocessed images to learn patterns and extract necessary features. This is achieved by fine-tuning its parameters and modifying them until the best values for each parameter are found, resulting in the model effectively detecting the desired object with maximum possible accuracy. The key hyperparameters that were modified in this study include:

- Learning rate (LR): Controls the extent of weight adjustment during model training.
- Batch size (BS): Determines the number of training samples processed together before updating the neural network weights.
- Number of epochs: Specifies the number of times the entire training dataset will be iterated during training.
- Optimization algorithms: Determine how the neural network weights are updated during training[26].

To adjust the hyperparameters in this paper, the following steps were followed:

- Starting with training the YOLOv8l version and modifying the hyperparameters using the Adam optimization algorithm. Initially, the LR and BS parameters were fixed (LR=0.001, BS=16), and the epoch was gradually increased starting from 20 (initial values were chosen based on experiments conducted on this model prior to adopting these values and consulting some studies that agreed on the effectiveness of these values and the available computational resources). After achieving the best model performance, the variable parameter (epoch) was fixed, and another parameter (LR) was modified until the best combination of hyperparameters was obtained, resulting in the model exhibiting the best performance.
- Next, the YOLOv8l model was trained with the SGD optimizer using the hyperparameters that showed the best model performance With the same parameters, it was trained using the natural dataset, meaning without applying the augmentation process.
- The next step involved training the YOLOv8m model with the best combination of hyperparameters from YOLOv8l, repeating the training process with an increased epoch value since this model is smaller than its counterpart.

2.3.2Model evaluation.

It is not enough for the system or model to only detect the desired objects; we must also assess the performance and accuracy of the detection system. Therefore, it must be evaluated through a series of metrics that allow us to obtain a clear analysis of whether the system is performing well or poorly. Below is a brief explanation of the series of metrics used to evaluate the model's performance.

- Precision: It measures the ratio of true positive detections among all positive predictions. Precision is given by the mathematical formula 1[26][27].

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{all\ detections} \quad (1)$$

- Recall: It measures the ratio of true positive detections among all ground truth positives. Recall is given by the mathematical formula 2[26][29].

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

- F1 Score: It is a metric that combines precision and recall into a single value, providing a balanced measure of the model's accuracy. F1 Score is given by the mathematical formula 3[26][30].

$$F1-Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (3)$$

- Mean Average Precision (mAP): It is a measure where the average precision is calculated by computing precision for each specific class at different decision thresholds (usually at 50% or 50-95%), then finding the average precision across all classes [30].
- Confusion Matrix: It is a square matrix used to evaluate the model's performance by comparing the values predicted by the system with the ground truth values. It takes the form (12.4), where if the model's prediction matches the true value, it is a true positive detection (TP), and if the prediction does not match the true value, it is a false positive detection (FP). Figure 4 illustrates the confusion matrix [27][28].

		Real Values	
Predicted Values		Positive	Negative
	Positive	TP = True Positive	FP = False Negative
	Negative	FN = False negative	TN = True Negative

Fig. 3: Confusion Matrix [27].

- Loss Functions: These provide different measures of loss, including the loss or failure of the model to predict the correct object class, the loss of predicting the object's location in the image, and the total loss of both [26].

2.4 Building a Real-Time Fire and Smoke Detection System.

In this stage, after the model has been trained, evaluated, and yielded the best results, the weights of the models (YOLOv8l, YOLOv8m) that provided the most accurate and effective results were loaded and used to build an automated system for fire and smoke detection and evaluate its performance in real-time. This was done using VSCode, a popular lightweight yet powerful free code editor developed by Microsoft, equipped with features such as syntax highlighting, code completion, and debugging capabilities.

2.4.1 System Development.

This stage involves programming a web camera, configuring it for object detection, localization, and displaying the confidence level in this detection. Additionally, it enables the camera to alert relevant authorities by sending an email notification and issuing an audible alarm upon detecting a fire or smoke.

2.4.2 System Implementation and Testing.

This stage includes system integration by integrating the trained model with the configured web camera. This stage also involves evaluating the system's performance by testing it in real-time using a web camera, evaluating it on various scenarios from the real world, ranging from a flicker to large-scale fires such as forests and buildings (this was done by running it in videos in front of the camera) at different thresholds.

3 Results

3.1 Training results.

Results of YOLOv8l release with various hyperparameters.

Table 2 illustrates the mean average precision at threshold 50 (mAP50) for the model across the three stages (training, validation, and testing) and F1 score after each training iteration with diverse hyperparameters using Adam optimizer.

Table 3: Results of YOLOv8l release with various hyperparameters.

NO	epoch	Lr	Bs	F1	mAP50 TRAIN	mAP50 validation	mAp50 TEST
1	20	0.001	16	79	84.1%	84.1%	0.487%
2	40	0.001	16	80	85.1%	85.2%	83.6%
3	50	0.001	16	81	85%	84.9%	83.4%
4	60	0.001	16	79	84.7%	85.1%	83%
5	40	0.001	18	80	84.8%	84.8%	83.3%
6	40	0.0001	16	82	86.3%	86.8%	82.8%
7	40	0.0001	18	82	84%	83.9%	82.2%

As illustrated in the table above, the best model performance was achieved with hyperparameter configuration 2. Although the model performance was slightly higher in hyperparameter configuration 6, the validation results showed an increase in loss indicators as depicted in Figure 5 compared to configuration 2, which exhibited a decreasing trend in loss indicators. Therefore, the best performance observed so far for the YOLOv8l model was with hyperparameter configuration (lr=0.001, bs=16, epoch=40), and Figures 6 to 9 detail the training results at these parameters, including overall performance curves, precision-recall curve, confusion matrix sequentially. Figure 10 displays a sample of the model predictions on test images.

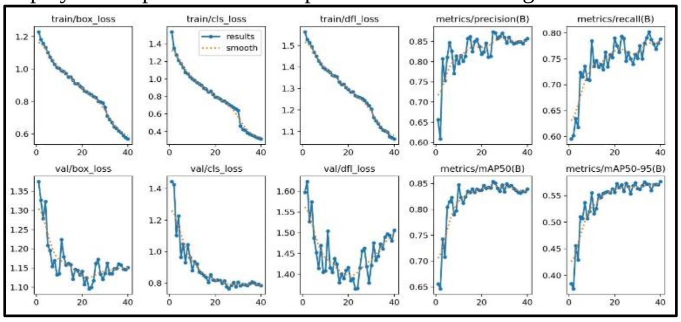

Fig. 5: Training results of YOLOv8l with parameters epoch=40, lr=0.0001, bs=16.

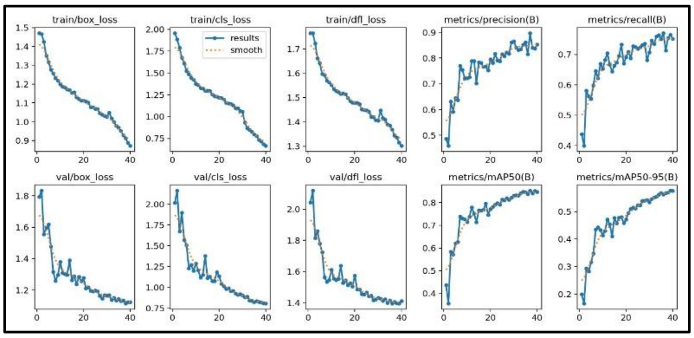

Fig. 6: Training results of YOLOv8l with parameters epoch=40, lr=0.001, bs=16.

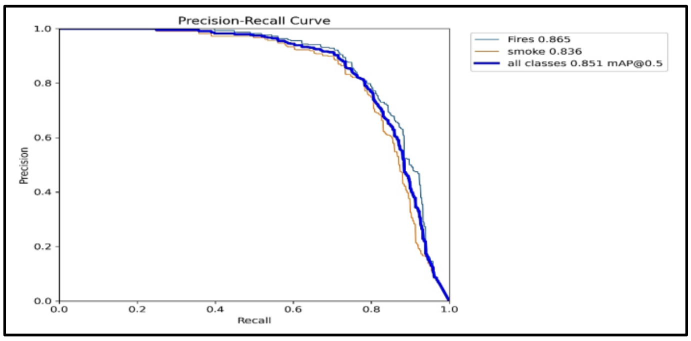

Fig. 7: The recall-precision curve.

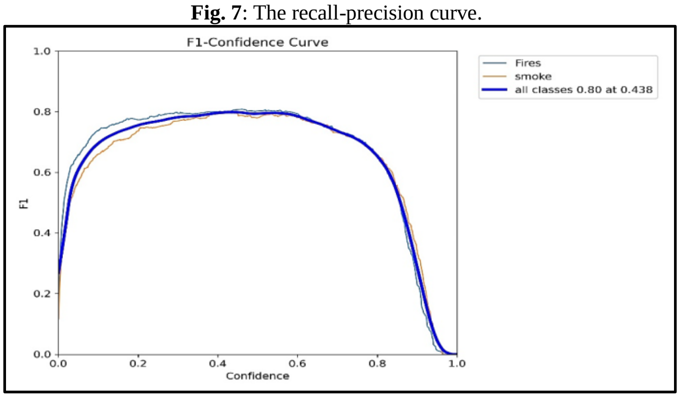

Fig. 8: the F1 curve.

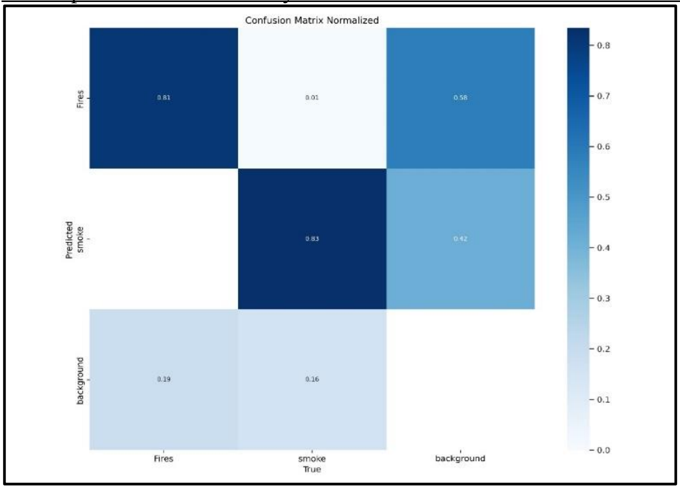

Fig. 9: Confusion Matrix.

Fig. 10: Sample of YOLOv8l model predictions on test images.

After obtaining the best parameters for the model with the Adam optimizer, the optimization algorithm was changed to SGD while maintaining the previous hyperparameters. Table 5 illustrates a comparison between the results of these algorithms.

Table. 4: Best hyperparameters for models

optimizer	F1	mAP50_TRAIN	mAP50_validation	mAP50_TEST
Adam	80	85.1%	85.2%	83.6%
SGD	81	85.3%	85.4%	83.7%

As indicated in the table above, the results of the SGD optimization algorithm showed better values in the majority of performance metrics outlined in Table 4. The reason for this is attributed to the increase in the overall loss indicator of the model, as depicted in Figure 11, compared to the results of the Adam optimizer shown in Figure 6, which indicates a decrease throughout the training in the loss indicator.

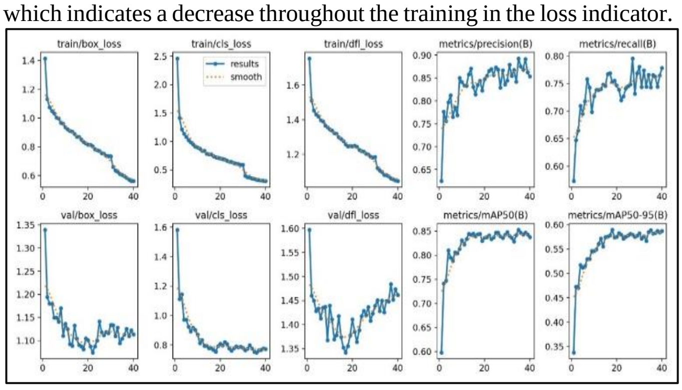

Fig. 11: Training results of the YOLOv8l model with and without data augmentation.

Table 5 illustrates the comparison results of the YOLOv8l model performance with and without data augmentation applied, using the optimal hyperparameters. It was tested using surveillance cameras, indicating another future direction.

Table 5: Comparison of the performance results of the YOLOv8l model with and without data augmentation applied.

Augmentation	F1	mAP50_TRAIN	mAP50_validation	mAP50_TEST
Yes	80%	85.1%	85.2%	83.6%
No	87%	84.4%	84.5%	80.7%

As shown in the table above, after training the model without applying data augmentation, a decrease in performance metrics was observed in all three stages, including the average precision mAP50 and F1 score, as indicated in Table 5. Additionally, the results showed an increase in loss metrics compared to training with augmentation. Moreover, the model's performance exhibited more oscillations in its curves, as depicted in Figure 12, compared to the results of the curves when data augmentation was applied, illustrated in Figure 6.

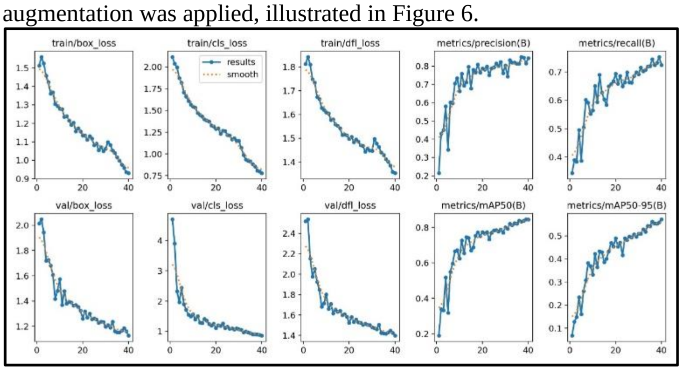

Fig. 12: Training results of the YOLOv8l model with and without data augmentation.

3.2.1 Results of YOLOv8m release with various hyperparameters.

Table 6 shows the training results of the YOLOv8m model with the optimal hyperparameters that yielded the best performance (epoch=40, lr=0.001, bs=16, optimizer=Adam), and the training results with an increase in the number of epochs to 50.

Table 6: Results of YOLOv8m release with various hyperparameters.

epoch	F1	mAP50_TRAIN	mAP50_validation	mAP50_TEST
40	80%	84.3%	84.3%	84%
50	82%	86%	84.3%	84%

As shown in the table above, the model's performance with an increase in the number of epochs to 50 is higher than when trained with the same parameters at 40 epochs. Figures 13 to 16 present detailed results of the best training for the YOLOv8m model at epoch 50, depicting overall performance curves, recall-precision curve, confusion matrix sequentially, and Figure 17 displays a sample of the model's predictions on test images.

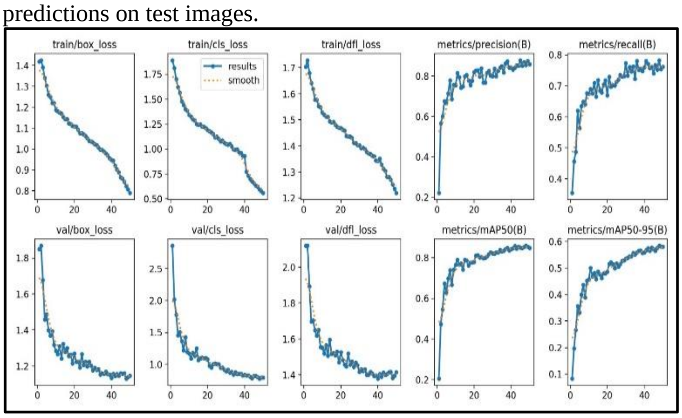

Fig. 13: Training results of YOLOv8l with parameters epoch=50, lr=0.001, bs=16.

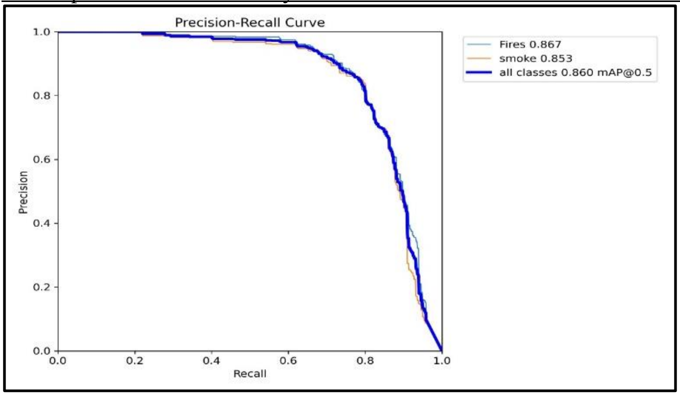

Fig. 14: The recall-precision curve.

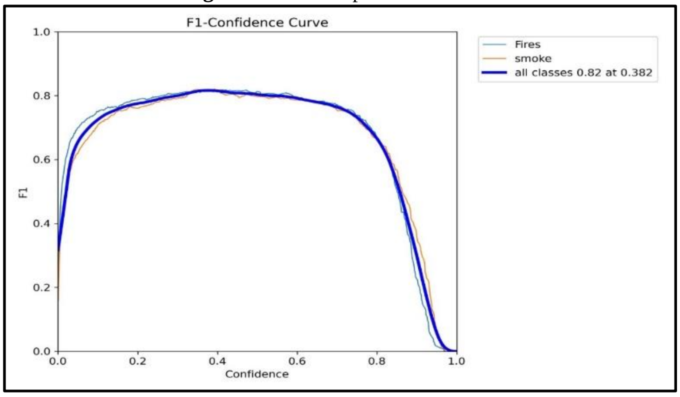

Fig. 15: the F1 curve.

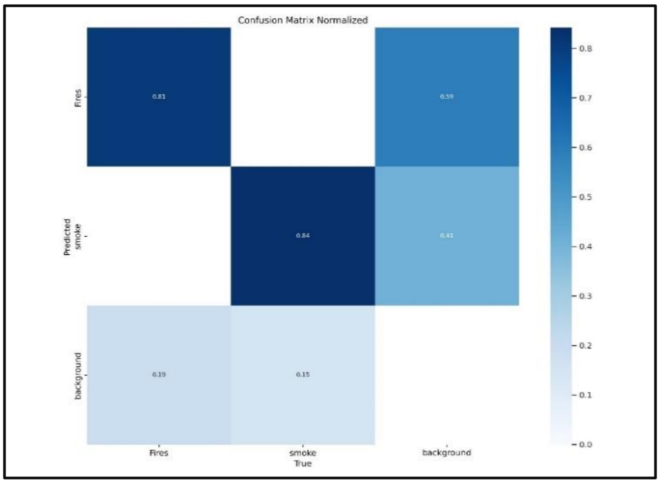

Fig. 16: Confusion Matrix.

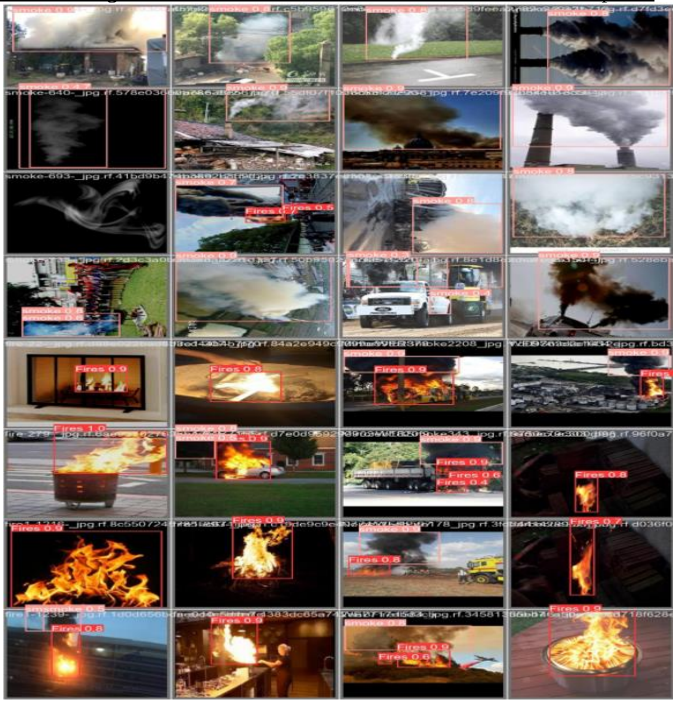

Fig. 17: Sample of YOLOv8m model predictions on test images.

3.2 Experimental results.

3.2.1 Actual detection results using YOLOv8l.

Figures 18 and 19 display samples of detection results using a webcam at thresholds of 0.25 and 0.50, respectively.

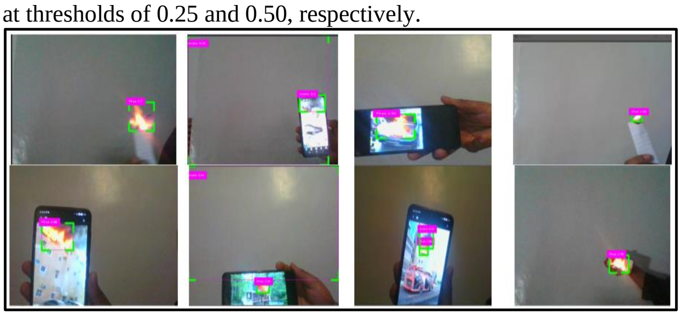

Fig. 18: Real-time detection results at threshold 0.25.

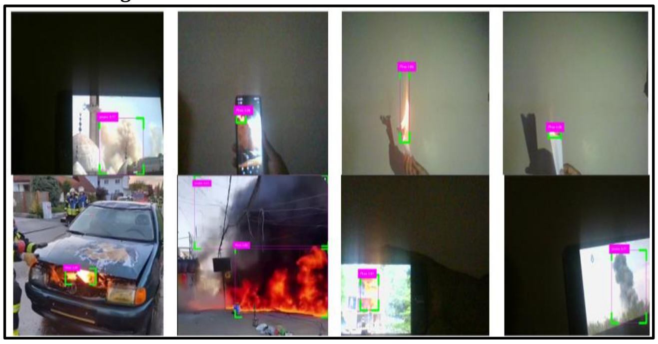

Fig. 19: Real-time detection results at threshold 0.5.

3.2.2 Actual detection results using YOLOv8m.

Figures 20 and 21 display samples of detection results using a webcam at thresholds of 0.25 and 0.50, respectively.

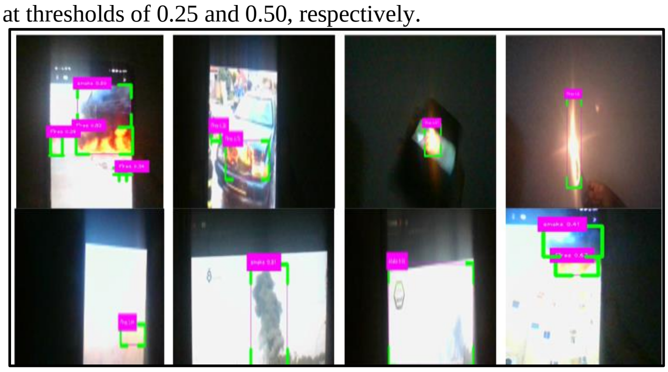

Fig. 20: Real-time detection results at threshold 0.25.

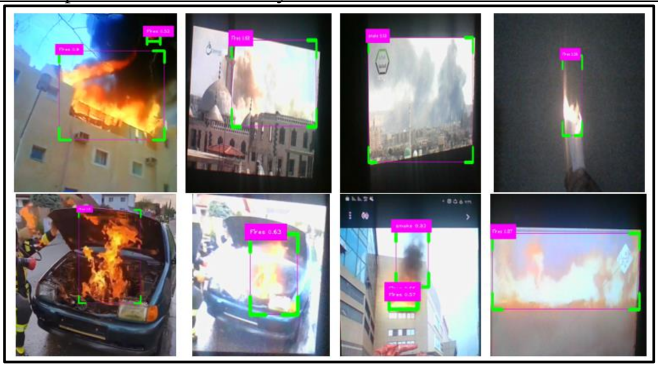

Fig. 21: Real-time detection results at threshold 0.5.

4 Discussion.

In this study, a real-time fire and smoke detection system was developed using the YOLOv8 model. The researchers experimented with two YOLOv8 models, YOLOv8l and YOLOv8m, and trained them with various hyperparameters, including learning rate, batch size, number of epochs, and optimization algorithm.

The YOLOv8l model prioritized accuracy over speed to reduce false alarms. The researchers tested different hyperparameter configurations and found that the best performance was achieved with a learning rate of 0.001, batch size of 16, and 40 epochs, using the Adam optimizer. The model showed an overall average precision of 85.1%, with higher precision for fire detection (86.5%) compared to smoke detection (83.6%).

The YOLOv8m model was trained to balance accuracy and speed. Using the same hyperparameters as the YOLOv8l model, the researchers further increased the number of epochs to 50, resulting in a more accurate performance across all indicators, with a training mAP50 of 86%, validation mAP50 of 84.8%, and testing mAP50 of 84%.

When tested in real-time, both the YOLOv8l and YOLOv8m models showed promising results under different lighting conditions and scenarios, with the performance being almost identical. At a lower threshold (0.25), the models were able to recall and detect fire and smoke effectively, although they made some errors in smoke-like cases. At a higher threshold (0.50), the results were more accurate, with fewer errors in detecting genuine cases.

Overall, the study demonstrates the effectiveness of the YOLOv8 model in developing a real-time fire and smoke detection system that can be deployed in various scenarios, with the ability to balance accuracy and speed depending on the specific requirements.

A comparison between the results of the current study and the findings from previous studies is presented in Table 7. This table provides a side-by-side comparison of the key performance metrics, highlighting the improved average precision achieved by the system developed in this research.

Table. 7: Comparison of the current study with previous studies.

Authors	Year	Number of Image	Model	Fire/Smoke	(mAp50)
Wang et al.	2022	Smoke	YOLOv5	20,000	84.5%
Bekkari & Kadi	2023	Fire/Smoke	YOLOv5	1,500	83.51%
Avazov at al.	2023	Fire	YOLOv7	4.622	81%
Zhang et al.	2014	Fire	YOLOv8	4,227	90%
Proposed method	2024	Fire/Smoke	YOLOv8	9,756	86%

5 Conclusion and Future Directions

In this study, a real-time smoke and fire detection system was developed using the YOLOv8 model. Two versions of the model, YOLOv8m and YOLOv8L, showed the ability to detect fires and smoke with an average precision (mAP50) of 85.1% and 86%, respectively, and a balance between precision and recall (F1 score) of 80% and 82%, respectively. Comparisons of the model's performance with different parameters revealed that the Adam optimizer yielded better results than SGD, and data augmentation improved the model's training efficiency. The system demonstrated real detections for all fires and smoke scenarios under diverse conditions, even with a low-quality camera.

For future work, the researchers plan to retrain the proposed model with a larger dataset, utilize the latest YOLO models (e.g., YOLOv9 and YOLOv8.2), and deploy and test the system using surveillance cameras.

6 References.

[1]- Chaudhary, M. T., & Piracha, A. (2021). Natural Disasters—Origins, Impacts, Management. Encyclopedia, 1 (4), 1101-1131.

[2]- Barmpoutis, P., Papaioannou, P., Dimitropoulos, K., & Grammalidis, N. (2020). A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing. Sensors *(Basel, Switzerland)*, 20 (22), 6442.

[3]- Vollmayer, D., & Gutman, M. (2019, November 26). Home fires spike over the holidays. What can you do to get out alive? ABC News.

[4]- Hall, S. (2023, October 31). Fire Loss in the United States. National Fire Protection Association.

[5]- Sun, J., Qi, W., Huang, Y., Xu, C., & Yang, W. (2023). Facing the Wildfire Spread Risk Challenge: Where Are We Now and Where Are We Going?, 6(6), 228.

[6]- Robinne, F.-N., Burns, J., Kant, P., Flannigan, M. D., Kleine, M., de Groot, B., & Wotton, D. M. (Eds.). (2018). Global fire challenges in a warming world: Summary note of a global expert workshop on fire and climate change, Vienna.

[7]- Sathishkumar, V. E., Cho, J., Subramanian, M., & Naren, O. S. (2023). Forest fire and smoke detection using deep learning-based learning without forgetting. Fire ecology, 19(1), 1-17.

[8]- Avazov, K., Mukhiddinov, M., Makhmudov, F., & Cho, Y. I. (2022). Fire Detection Method in Smart City Environments Using a Deep-Learning-Based Approach. Electronics, 11(1), 73.

[9]- Talaat, F. M., & ZainEldin, H. (2023). An improved fire detection approach based on YOLO-v8 for smart cities. Journal Title (italicized), Volume(Issue), Page Range.

[10]-An, L., Chen, L., & Hao, X. (2023). Indoor Fire Detection Algorithm Based on Second-Order Exponential Smoothing and Information Fusion. Information, 14(5), 258.

[11]-Saydirasulovich, S. N., Mukhiddinov, M., Djuraev, O., Abdusalomov, A., & Cho, Y.-I. (2023). An Improved Wildfire Smoke Detection Based on YOLOv8 and UAV Images.

[12]-Li, X., Vázquez-López, A., Sánchez Del Río Sáez, J., & Wang, D. Y. (2022). Recent advances on early-stage fire-warning systems: Mechanism, performance, and perspective. Nanomicro Letters, 14 (1), 197. Erratum in Nanomicro Letters, 14 (1), 226.

[13]-Truong, C. T., Nguyen, T. H., Vu, V. Q., Do, V. H., & Nguyen, D. T. (2023). Enhancing Fire Detection Technology: A UV-Based System Utilizing Fourier Spectrum Analysis for Reliable and Accurate Fire Detection. Applied Sciences, 13 (13), 7845.

[14]-Carta, F., Zidda, C., Putzu, M., Loru, D., Anedda, M., & Giusto, D. (2023). Advancements in Forest Fire Prevention: A Comprehensive Survey. Sensors, 23(14), 6635.

[15]-Yang, H., Wang, J., & Wang, J. (2023). Efficient Detection of Forest Fire Smoke in UAV Aerial Imagery Based on an Improved Yolov5 Model and Transfer Learning. Remote Sensing, 15(23), 5527.

[16]-Avazov, K., Jamil, M. K., Muminov, B., Abdusalomov, A. B., & Cho, Y.-I. (2023). Fire Detection and Notification Method in Ship Areas Using Deep Learning and Computer Vision Approaches. Sensors, 23(16), 7078.

[17]-Wang, Z., Wu, L., Li, T., & Shi, P. (2022). A smoke detection model based on improved YOLOv5. Mathematics, 10(7), 1190.

[18]-BEKKARI, L., & KADI, N. Détection D'incendie Et De Fumée A L'aide De L'apprentissage Par Transfert (Fire and Smoke Detection Using Transfer Learning (Doctoral dissertation, UNIVERSITY OF KASDI MERBAH OUARGLA).

[19]-Avazov, K., Jamil, M. K., Muminov, B., Abdusalomov, A. B., & Cho, Y. I. (2023). Fire detection and notification method in ship areas using deep learning and computer vision approaches. Sensors, 23(16), 7078.

[20]-Zhang, Z., Tan, L., & Tiong, R. L. K. (2024). Ship-Fire net: an improved YOLOv8 algorithm for ship fire detection. Sensors, 24(3), 727.

[21]-Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines, 11*(7), 677.

[22]-Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 5(4), 1680-1716.

[23]-https://blog.roboflow.com/whats-new-in-yolov8/#what-is-yolov8.

[24]-https://learnopencv.com/ultralytics-yolov8/#YOLOv8-vs-YOLOv5.

[25]-Arregui Arias, A. B. (2023). Detección de cascos de seguridad en tiempo real mediante modelos de Deep Learning.

[26]-Efraimsson, A., & Lemón, E. (2022). Railway Fastener Fault Detection using YOLOv5.

[27]-Rozada Raneros, S. (2021). Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning.

[28]-Taco Apolo, C. P. (2022). Aplicación de técnicas de aprendizaje de máquina para la detección de aglomeraciones de personas y comportamientos anómalos mediante segmentación semántica y redes convolucionales: Modelo de Redes Convolucionales basadas en regiones para la detección de aglomeraciones en videos aplicando segmentación semántica (Bachelor's thesis, Quito: EPN, 2022.).

[29]-Sánchez Pozo, J. J. (2022). Detección y Segmentación en Placas de Rayos X de Pinturas utilizando IA.

[30]-Arregui Arias, A. B. (2023). Detección de cascos de seguridad en tiempo real mediante modelos de Deep Learning.