Real-Time Inspection of Fire Safety Equipment using Computer Vision and Deep Learning

Asmaa Alayed
College of Computing, Umm Al-Qura University, Saudi Arabia
asayed@uqu.edu.sa (corresponding author)

Rehab Alidrisi
College of Computing, Umm Al-Qura University, Saudi Arabia
s439016154@st.uqu.edu.sa

Ekram Feras
College of Computing, Umm Al-Qura University, Saudi Arabia
s441015984@st.uqu.edu.sa

Shahad Aboukozzana
College of Computing, Umm Al-Qura University, Saudi Arabia
s441003855@st.uqu.edu.sa

Alaa Alomayri
College of Computing, Umm Al-Qura University, Saudi Arabia
s441008331@st.uqu.edu.sa

Received: 15 December 2023 | Revised: 6 January 2024 and 27 January 2024 | Accepted: 5 February 2024

ABSTRACT

The number of accidental fires in buildings has been significantly increased in recent years in Saudi Arabia. Fire Safety Equipment (FSE) plays a crucial role in reducing fire risks. However, this equipment is prone to defects and requires periodic checks and maintenance. Fire safety inspectors are responsible for visual inspection of safety equipment and reporting defects. As the traditional approach of manually checking each piece of equipment can be time-consuming and inaccurate, this study aims to improve the inspection processes of safety equipment. Using computer vision and deep learning techniques, a detection model was trained to visually inspect fire extinguishers and identify defects. Fire extinguisher images were collected, annotated, and augmented to create a dataset of 7,633 images with 16,092 labeled instances. Then, experiments were carried out using YOLOv5, YOLOv7, YOLOv8, and RT-DETR. Pre-trained models were used for transfer learning. A comparative analysis was performed to evaluate these models in terms of accuracy, speed, and model size. The results of YOLOv5n, YOLOv7, YOLOv8n, YOLOv8m, and RT-DETR indicated satisfactory accuracy, ranging between 83.1% and 87.2%. YOLOv8n was chosen as the most suitable due to its fastest inference time of 2.7 ms, its highest mAP0.5 of 87.2%, and its compact model size, making it ideal for real-time mobile applications.

Keywords-Fire Safety Equipment (FSE); fire safety inspection; visual inspection; deep learning; computer vision; object detection; You-Only-Look-Once (YOLO)

I. INTRODUCTION

The risk of accidental fire in buildings is one of the most dangerous ones due to the potentially of incurreing huge amounts of damage and costs. According to the Saudi Civil Defense statistics, economic losses have recently increased by 61% [1]. In Saudi Arabia, fires are common, with an average of 42 fires per day [1]. The most dangerous fires are those that occur in crowded areas, as they can cause a high number of injuries and deaths, such as the Mena accident in 1997 during the Hajj season in Makkah, Saudi Arabia, which caused more than 340 deaths and 1,500 casualties. Thus, not only is the development and implementation of fire safety protocols in buildings required by the civil defense laws and regulations, but it is also crucial for the safety of all those individuals being in a building during a fire emergency [2].

In Saudi Arabia, most current fire protection measures are prescriptive and based on the requirements of the Saudi Building Code for Fire Protection [3]. Buildings are equipped with fire safety devices and procedures to minimize the risk of fire and its impact [4]. However, the mere existence of this equipment does not necessarily mean that the latter will be effective, as it is prone to failures and impairments for many reasons, such as human errors, mechanical failure, neglect, and the environment [5]. It is estimated that a third of Fire Safety Equipment (FSE) would not work properly in an emergency [6]. The Saudi government has made significant efforts to reduce the risk of fire, especially at events and during religious seasons, to protect the lives of people and minimize damage to buildings. One of these efforts is the implementation of fire safety inspections conducted by safety and preventive supervision teams in the Civil Defense. The inspection and maintenance of the FSE are designed to ensure that it is always in excellent operating condition [7]. FSE inspection is divided into several modes. Visual inspection is part of the FSE inspection process and aims to evaluate its condition (damaged or not) as well as the suitability of its configuration [7]. It is now mandatory to regularly inspect and document the presence, location, and working order of all FSEs over their lifespan according to regulations and modern safety standards [5]. The regulations related to the visual inspection of fire extinguishers are as follows:

- The fire extinguisher should be visible and free of obstructions.
- The pressure gauge should be pointed to the green zone.
- There should be no rust or damage to the extinguisher.
- Check the condition of the hose connected to the fire extinguisher.
- Ensure the presence of the safety pin.
- Ensure the existence of labels that show the expiration date of the extinguisher.

Currently, the visual approach to detect defects in safety equipment is the standard method to evaluate its quality [8]. However, due to inaccuracies in identifying defects or damage from photographic evidence, visual inspection solutions are labor-intensive, time-consuming, and error-prone [9]. Thus, conventional human visual inspection is a difficult-to-measure process with variable and subjective results [8]. Several Saudi government applications offer solutions for the inspection process, such as Madani [10], and Salamti [11], which allow the public to file safety violation reports so that preventive supervision teams can take action to avert the occurrence of accidents. However, there are limitations to these applications, including the possibility of reporting incorrect or inaccurate violations due to the public's inexperience in the inspection process. Furthermore, it is time-consuming for preventive supervision teams to personally validate each violation.

Deep learning is a popular technology used in different domains, including text processing, medical diagnosis, weather forecasting, and climate change analysis. With deep learning, computer vision has produced encouraging results, like in weed detection [12], agriculture, satellite image analysis [13], and fire detection [14]. YOLO (You Only Look Once) is a deep learning algorithm that shows high speed and accuracy. Therefore, YOLO is widely used in UAV image detection [15-16] and real-time applications [17]. Several studies have explored the use of deep learning for FSE inspection. In [18], a custom Convolutional Neural Network (CNN) approach was proposed for object recognition in fire safety systems. In [19], the Single Shot Detector (SSD) was proposed to locate fire equipment in 3D building layouts, achieving 76% accuracy for extinguisher detection. In [20-21], YOLOv5 and YOLOv7 were used for FSE object detection. In [22], an advanced solution was proposed that could do more than detect the presence of FSE, such as fire extinguishers, by identifying its condition as defective or non-defective. In this study, MobileNet V2 SSDLite, FPN Resnet-50 SSD, and Inception Resnet v2 Faster R-CNN achieved an accuracy rate of 86%, but some inspection criteria, like gauge checking, were missing. Additionally, the lack of a dataset can lead to lower accuracy in fire extinguisher detection.

This study aims to solve the pressing problems of FSE inspection and defect detection automation and intelligence levels using deep-learning computer vision. This study trained defect inspection models to inspect various specific fire extinguisher conditions according to [3]. As shown in Figure 1, a mobile application will use the model to identify defective equipment, document it, and send it to the relevant authority to quickly determine the appropriate action procedure. This study intended to:

- Create and annotate a comprehensive data set for both defective and non-defective fire extinguishers.
- Evaluate the effectiveness of various versions of the YOLO algorithm alongside transformers, and compare precision, speed, and model size.

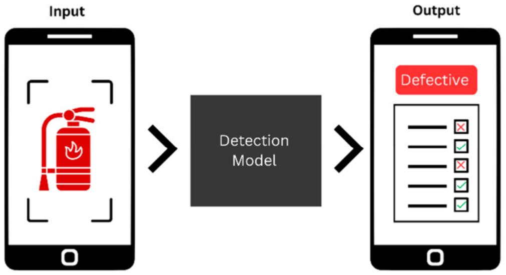


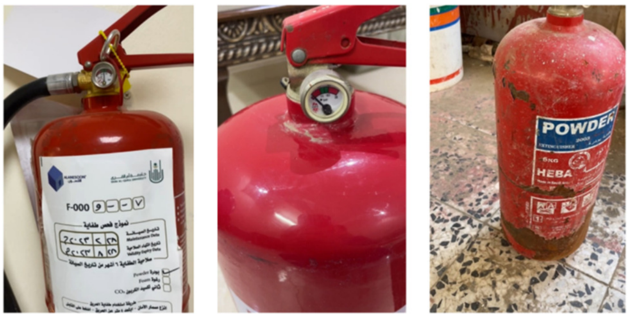
Fig. 1. Overview of a mobile application that uses the proposed model to inspect an extinguisher and run through the checklist automatically.

II. DATA COLLECTION AND RESEARCH METHOD

A. Dataset Collection

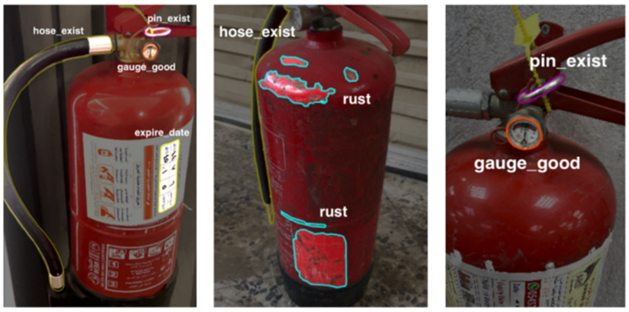
High-quality image datasets are essential for training deep learning models [23]. In the context of the fire extinguisher inspection, the dataset must encompass both non-defective and defective fire extinguishers, covering all potential reasons for defects. After conducting a thorough search, no existing dataset was found to meet these specific requirements. In addition, the available extinguisher datasets were relatively small and included extinguisher models that are not commonly used in Saudi Arabia. As a result, a dataset of fire extinguishers was built, including different types of defective and non-defective fire extinguishers. This study focused on specific types of extinguishers: powder, water, and foam. These types are similar in shape and size and are most commonly used in buildings. Images were gathered from various facilities, including the university, schools, shops, restaurants, residential buildings, and hospitals. To collect these data, a technique of capturing videos of fire extinguishers and then extracting images from these videos was adopted. This approach allowed us to generate a high volume of data while varying the angles and distances. The video speed was slow enough to avoid blurry images, but fast enough to prevent image duplication. This adjustment significantly affected the quality of the dataset. The best way to capture videos was reached after several rounds of attempts. On average, the collected video time was 6.35 seconds and the images were extracted at 5-15 fps. Finding an adequate number of fire extinguishers was not difficult since they are available in different facilities and public places. However, it was a challenge to locate fire extinguishers that were truly rusted or had defective gauges. This was due to the rigorous safety laws and regulations in Saudi Arabia that require the replacement of defective fire extinguishers with fully functional ones. Therefore, several fire safety centers were contacted to help in finding a wide variety of defective extinguishers to add such images to the dataset. As a result, 3,592 images were collected.

The next step was the annotation process that involved manual labeling of the classes and localization of the objects in the collected images. The class labels were selected based on five key criteria informed by the regulations that govern the inspection and appearance of fire extinguishers. First, the presence of a hose was only noted when the hose existed. Non-existent hoses cannot be detected by the model. The same is true for fire safety pins and expiration date labels. For the defective body, only the rusty parts were annotated, allowing the model to recognize the rust characteristics instead of attempting to identify the overall appearance of the rusted extinguishers. Additionally, the gauge was annotated as either good or bad based on its indicator position. If the indicator fell within the green area, the gauge was labeled as good, whereas a position in the red area indicated a bad gauge. Finally, the dataset consisted of six classes: "hose_exist", "pin_exist", "rust", "gauge_good", "gauge_bad", and "expire_date". Figure 2 displays a sample of the collected images. During the annotation process, it was found that individual images often contained multiple objects of interest. An instance segmentation approach was used to accurately delineate each separate object by surrounding them with polygons [24]. Roboflow [25], was employed to segment objects. Figure 3 presents the outcome of this segmentation process. The latter proved invaluable for object detection tasks and also expanded the utility of the dataset beyond mere detection. The images and videos collected were taken on various mobile devices, resulting in nonuniform sizes. Standardizing the input size and maintaining consistent resolution were crucial for the dataset. To achieve this, during the preprocessing step, all images were resized to a uniform resolution of 640×640 pixels. It is important to note that this size was chosen carefully to strike a balance: neither too small to accommodate small objects, nor too large to slow down the training process. The rectangular images were filled with black padding, ensuring that the extinguishers were not distorted or sheared.



(a) (b) (c)

Fig. 2. Dataset examples of defective and non-defective fire extinguishers: (a) Non-defective extinguisher meeting all conditions; (b) Defective extinguisher lacking both safety pin and hose, and the gauge indicator being in the red area; (c) Defective extinguisher showing rust and absence of the hose.



(a) (b) (c)

Fig. 3. Examples of annotated images: (a) All parts of the extinguisher; (b) Rust detection; (c) Closed image to small objects.

Data augmentation techniques were applied to diversify, generalize, and increase the dataset. These techniques included rotations of up to 15°. The rotation aimed to expose the model to objects from various angles. Other techniques were brightness adjustments of 25% and saturation modifications of 25%. These can simulate different lighting conditions, thereby enhancing the model's ability to classify objects accurately. Eventually, the dataset consisted of 7,663 fire extinguisher images with 16,092 annotations, with an average of 2.1 objects per image. Figure 4 illustrates the number of instances per class. The dataset was sorted and randomly split into training sets, validation sets, and test sets, according to an 8:1:1 proportion. Empirical analysis has proven that 80% of training yields the highest results [26]. Inspecting an extinguisher using a single photo can be challenging since certain parts may not always be fully visible, such as when the safety pin is positioned at the back or when there is a tag covering the gauge, as shown in Figure 5. Employing a real-time solution can be advantageous as it allows users to receive instant feedback when the details are not clearly visible, enhancing the effectiveness of the inspection process.

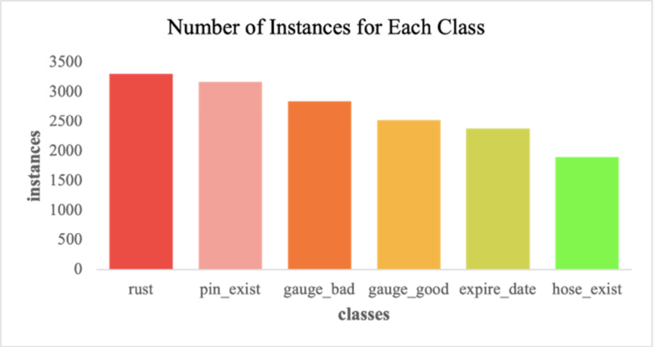


Fig. 4. Number of instances for each class.

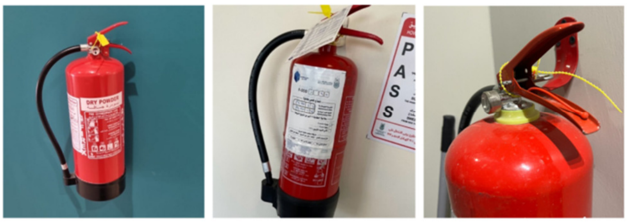


Fig. 5. Some parts of extinguishers can be hidden: (a) The gauge is not clear because it is too far away; (b) The gauge is covered by the tag; (c) The angle of the shot does not clearly show the safety pin.

B. YOLO Algorithms

Computer vision projects using deep learning approaches have produced remarkable results [27]. One such powerful example is the YOLO algorithm [28-29], which has become a cutting-edge algorithm in object detection. YOLO works by layering a grid onto an image and passing it through a neural network once only. This makes YOLO the fastest available object detection algorithm and an excellent choice for real-time applications [30]. The YOLO algorithm has been continuously improved and modified over the years, resulting in several versions. This study selected the most recent ones, which are YOLOv5, YOLOv7, and YOLOv8. These versions provide lightweight models, offering higher accuracy and faster processing times [31].

1) YOLOv5

YOLOv5n is the lightest, smallest, and fastest model in terms of detection speed [31-32]. This study considered model efficiency, accuracy, and size. Based on the YOLOv5n architecture, an improved design was developed to detect defective fire extinguishers. The YOLOv5n architecture includes three components: the backbone network, the neck network, and the detection network. In the backbone, features are extracted from input images, which are essential for object detection, while in the neck, three different scales of feature maps are created, which are used by the detection head [33].

2) YOLOv7

YOLOv7 is a real-time object detector currently driving a profound transformation in computer vision. YOLOv7 proposed the Efficient Layer Aggregation Networks (ELAN) architecture, which can improve the self-learning capacity of the network without destroying the original gradient path [34].

3) YOLOv8

One of the most significant changes in YOLOv8 is the use of free anchor boxes. Anchor boxes are typically employed in object detection models to help them predict the location and size of objects in an image. This feature reduces detection time by speeding up Non-Maximum Suppression (NMS), a post-processing step that sifts through candidate detections after inference [35]. Another key improvement in YOLOv8 is that it utilizes new loss functions for bounding box loss and classification loss. These functions improve its performance, particularly when dealing with smaller objects [32].

C. RT-DETR Model

Existing real-time detectors typically use CNN-based architectures to achieve a reasonable trade-off between detection speed and accuracy. Transformer-based detectors (DE-TRs) have recently demonstrated exceptional performance. However, their high computational cost results in slow real-time object detection. Real-Time Detection Transformer (RT-DETR) was developed [36], which is an end-to-end object detector that offers real-time performance and high accuracy. It engages a hybrid encoder that processes multiscale features by decoupling intra-scale interaction and cross-scale fusion. This one-of-a-kind vision transformer-based design reduces computational cost. RT-DETR is highly adaptable, allowing for flexible inference speed adjustments without retraining. This model outperforms current real-time detectors in accuracy and speed, requires no postprocessing, ensures stable inference speed, and fully utilizes an end-to-end detection pipeline advantage [36].

D. Training Methodology

This study aimed to perform inspection tasks with the highest performance while maintaining a minimal size suitable for use in a mobile application. YOLO offers different sizes (nano, small, medium, large, and xlarge) that provide various trade-offs between speed and accuracy. Several experiments were carried out to evaluate the previously mentioned YOLO versions for nano size, YOLOv5n, YOLOv7-tiny, and YOLOv8n. Additionally, YOLOv8m was investigated for improved accuracy. The evaluation was extended to include RT-DETR. This allows us to comprehensively assess and compare the efficacy of various models in the search for superior inspection capabilities.

As the dataset was not big enough, transfer learning was implemented to fine-tune pre-trained model weights on new data. Therefore, all the previously mentioned models were pre-trained on the Common Objects in Context (COCO) dataset, which is one of the largest datasets made by Microsoft [37]. These models were trained using a set of hyperparameters, including Adam optimizer, a learning rate of 0.001, and batch size of 64 and 50 epochs with a training dataset of 6130 images, a testing set of 766 images, and a validation set of 766 images. Throughout the training process, all models were configured consistently, except for RT-DETR's batch size, which due to its large size could only handle a batch size of 16. Table I portrays the hyperparameters.

TABLE I. THE HYPERPARAMETERS SET DURING MODEL TRAINING

Hyperparameter	Value for YOLO	Value for RT-DETR
Input image size	640	640
Epochs	50	50
Batch size	64	16
Optimizer	ADAM	ADAM
Initial learning rate	0.001	0.001
Final learning rate	0.00001	0.00001
Momentum	0.937	0.937
Weight decay	0.0005	0.0005

E. Training Environment

Training a model requires higher computational resources such as GPUs. Google Colab, which is a cloud-based platform to execute Python code, was used to train the models. The resources that Google Colab offers for free include a GPU T4 graphics card with 12GB of VRAM. The paid version was utilized for some slow experiments. Colab Pro offers more options for powerful GPUs, including GPU V100 and GPU A100, which are faster than GPU T4. Using the upgraded version considerably speeded up the training process.

F. Evaluation Metrics

Several metrics, such as precision (*P*), recall (*R*), and *mAP,* are applied to assess the performance of a model and its ability on the detection task. These metrics provide a comprehensive knowledge of how accurate and reliable the model is at detecting defective fire extinguishers. These metrics are calculated using a confusion matrix that consists of four parts:

1. True Positives (*TP*): The number of instances that belong to the positive class and are correctly classified as positive by the model.
2. False Positives (*FP*): The number of instances that belong to the negative class but are incorrectly classified as positive by the model.
3. True Negatives (*TN*): The number of instances that belong to the negative class and are correctly classified as negative by the model.
4. False Negatives (*FN*): The number of instances that belong to the positive class but are incorrectly classified as negative by the model.

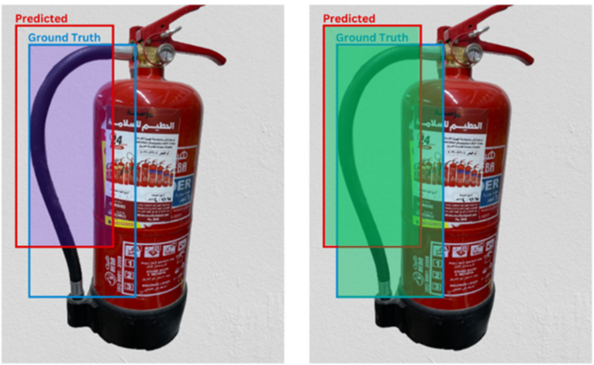
From these values:

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

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

Another important term for object detection is the Intersection over Union (IoU) [24]. This metric measures the accuracy of each bounding box by taking the ratio of overlapping areas between the actual (B_{gt}) and predicted (B_{pr}) bounding boxes to the area of their union. Equation 3 shows its formula and Figure 6 illustrates it.

$$IoU = \frac{B_{pr} \cap B_{gt}}{B_{pr} \cup B_{gt}} \quad (3)$$



(a) (b)

Fig. 6. IoU is the ratio of the intersection area over the union area: (a) Intersection area; (b) Union area.

The Mean Average Precision at 0.5 (*mAP0.5*) is a comprehensive metric that consolidates these various aspects. First, it involves the Average Precision (*AP*) which is the area under the precision-recall curve [24]. The individual-class *AP* values are then averaged to derive *mAP*. *mAP0.5* indicates that this calculation uses an *IoU* threshold of 0.5. These properties make *mAP0.5* a suitable metric for most detection applications.

III. RESULTS AND DISCUSSION

After conducting a series of experiments, a comparative analysis was performed to evaluate the models and identify theιρ optimal performance based on accuracy, speed, and model size.

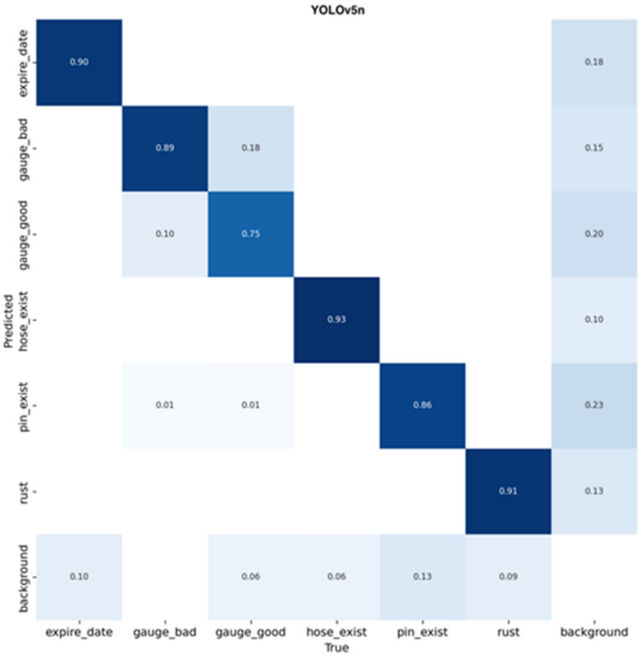
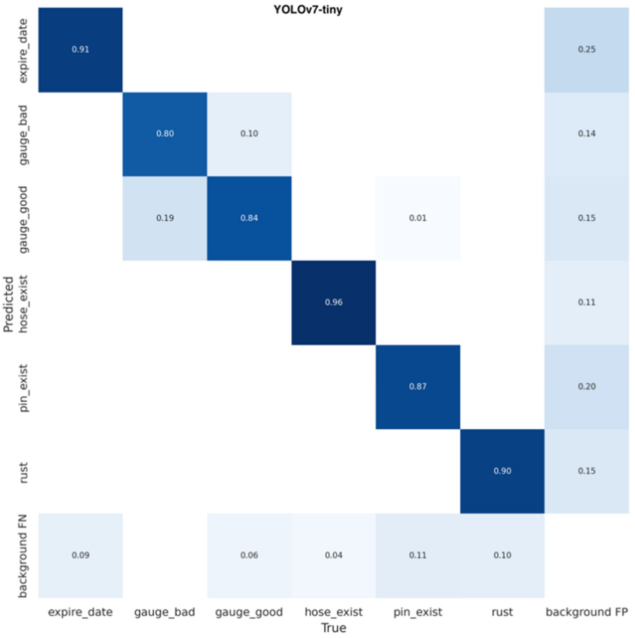


Fig. 7. Confusion matrix for YOLOv5n.

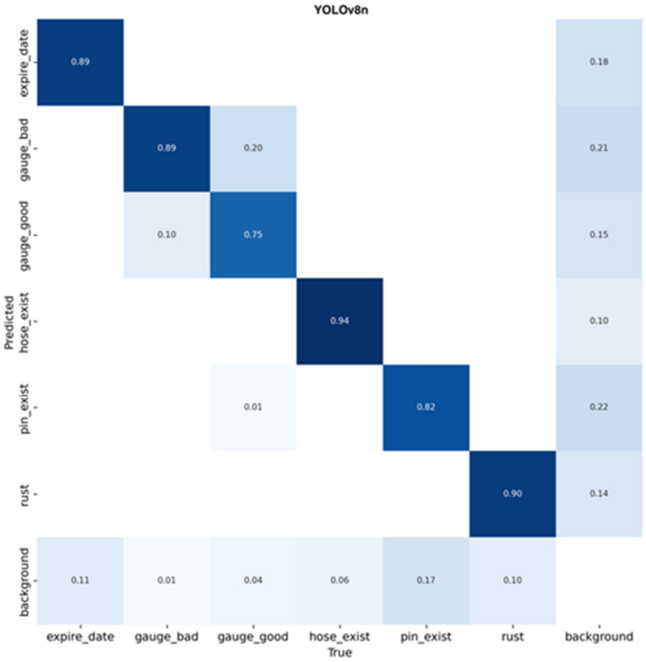
A. Confusion Matrix

The results presented in Figures 7-11 show the confusion matrices for each model at a confidence level of 0.25. In the confusion matrix, the diagonal line shows instances correctly classified by the model, offering a visual representation of its accuracy. Gauge classes are difficult to detect, leading to the lowest accuracy levels. In addition, there was some degree of misclassification between gauge good and bad.



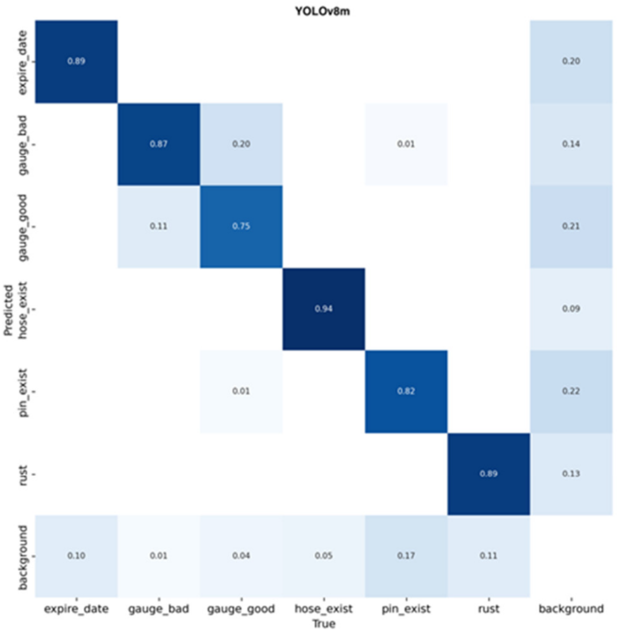
	expire_date	gauge_bad	gauge_good	hose_exist	pin_exist	rust	background FP
expire_date	0.91						0.25
gauge_bad		0.80	0.10				0.14
gauge_good		0.19	0.84		0.01		0.15
hose_exist				0.96			0.11
pin_exist					0.87		0.20
rust						0.90	0.15
background FN	0.09		0.06	0.04	0.11	0.10	

Fig. 8. Confusion matrix for YOLOv7tiny.



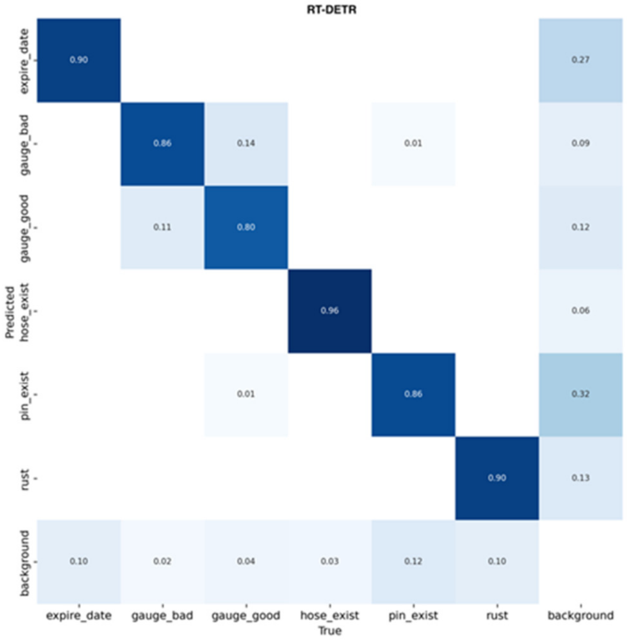
	expire_date	gauge_bad	gauge_good	hose_exist	pin_exist	rust	background
expire_date	0.89						0.18
gauge_bad		0.89	0.20				0.21
gauge_good		0.10	0.75				0.15
hose_exist				0.94			0.10
pin_exist			0.01		0.82		0.22
rust						0.90	0.14
background	0.11	0.01	0.04	0.06	0.17	0.10	

Fig. 9. Confusion matrix for YOLOv8n.



	expire_date	gauge_bad	gauge_good	hose_exist	pin_exist	rust	background
expire_date	0.89						0.20
gauge_bad		0.87	0.20		0.01		0.14
gauge_good		0.11	0.75				0.21
hose_exist				0.94			0.09
pin_exist			0.01		0.82		0.22
rust						0.89	0.13
background	0.10	0.01	0.04	0.05	0.17	0.11	

Fig. 10. Confusion matrix for YOLOv8m.



	expire_date	gauge_bad	gauge_good	hose_exist	pin_exist	rust	background
expire_date	0.90						0.27
gauge_bad		0.86	0.14		0.01		0.09
gauge_good		0.11	0.80				0.12
hose_exist				0.96			0.06
pin_exist			0.01		0.86		0.32
rust						0.90	0.13
background	0.10	0.02	0.04	0.03	0.12	0.10	

Fig. 11. Confusion matrix for RT-DETR.

B. Mean Average Precision and Model Size

Table II presents the metrics for each model, along with their respective sizes. Compared to other models, the YOLOv8n model excelled in terms of *mAP*, achieving 87.2%. On the contrary, RT-DETR achieved the lowest mAP of 83.1%. When considering model size, YOLOv5n stood out by attaining the most modest model size among other models, while RT-DETR had the largest model size, which is unsuitable for a mobile application. Figure 12 illustrates the *mAP0.5* of each model over 50 epochs.

TABLE II. MODEL EVALUATION METRICS COMPARISON.

Model	Model Size (MB)	*P* (%)	*R* (%)	*mAP0.5* (%)
YOLOv5n	**3.9**	82.1	82.9	86.1
YOLOv7- tiny	12.3	**84.5**	82.1	85.8
YOLOv8n	6.3	82.5	83.7	**87.2**
YOLOv8m	52.0	83.8	83.5	86.6
RT-DETR	66.2	82.7	**83.8**	83.1

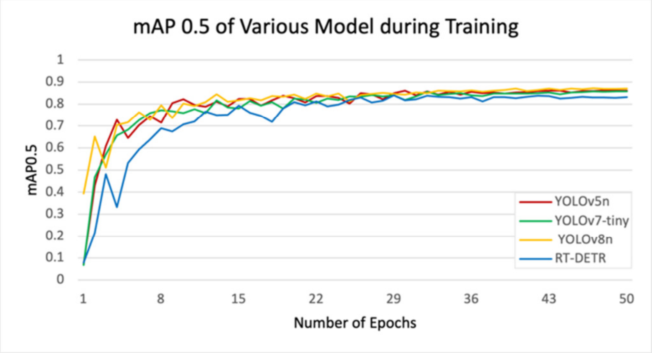


Fig. 12. *mAP0.5* of each model over 50 epochs.

C. Speed

As the proposed solution operates in real-time, temporal efficiency is of paramount importance. Table III provides a breakdown of time across two phases, inference (time taken for passing the image through the neural network), and post-processing required for the NMS algorithm. YOLOv8n exhibited the fastest overall time at 2.7ms, whereas YOLOv5n had the slowest performance. Notably, RT-DETR stands out for its commendable post-processing time, as its key feature. However, the inference time hinders the achievement of an expedited total detection time.

TABLE III. COMPARISON OF DETECTION TIME OF EACH MODEL (USING T4 GPU)

Model	Inference (ms)	Postprocessing (ms)	Total Time (ms)
YOLOv5n	43.1	1.5	44.6
YOLOv7- tiny	4.9	1.5	6.4
YOLOv8n	2	0.7	**2.7**
YOLOv8m	23.2	2	25.2
RT-DETR	16.5	0.3	16.8

D. Discussion

The results demonstrate that YOLOv8n fulfills the requirements in terms of overall performance, given the high importance of attaining real-time solutions. Although YOLOv8n did not achieve the smallest model size, it successfully met the requirement of accurately detecting defective equipment at high speed. Table IV displays the precision, recall, and mAP for each class. The "hose_exist" class reveals the highest results because it is easy to see from any angle or distance. However, "gauge_bad" exhibits lower precision due to more mistakes. This happens when the gauge is far away or in dark light, making it hard to detect if it is in the green or red area. So, the model sometimes predicts both good and bad situations, leading to less accurate results. Figure 13 depicts a sample of testing results.

TABLE IV. YOLOV8N RESULTS FOR EACH CLASS

Class	*P* (%)	*R* (%)	*mAP0.5* (%)
all	82.5	83.7	87.2
expire_date	82.6	85.0	81.4
gauge_bad	67.9	89.0	88.9
gauge_good	81.9	69.6	84.3
hose_exist	**93.6**	**93.1**	**97.5**
pin_exist	78.9	78.3	80.7
rust	89.9	86.9	90.3

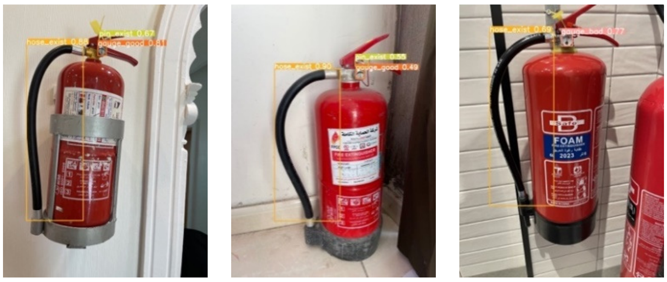


Fig. 13. Sample of testing results of YOLOv8n.

In contrast, the poor efficiency and large model size of RT-DETR make it an unsuitable choice for the detection of fire equipment in complex environments. Additionally, the slower performance speeds of both YOLOv5n and RT-DETR further diminish their suitability for such applications. This study demonstrated better results than [22], which is attributed to the use of the latest version of the YOLO algorithm (YOLOv8n). Furthermore, the models in this study benefited significantly from the rich dataset, which contributed to their improved performance.

The application of YOLOv8n was put to a practical test by expert inspectors, who offered insightful comments. Their first-hand experience confirmed the efficiency, effectiveness, and usefulness of the proposed model, adding substantial value to inspections with technologies previously unavailable. The experts also highlighted the future potential of the model, emphasizing how well it may integrate with other technologies to increase effectiveness. Moreover, they pointed out its potential to improve accuracy, speed up the inspection process, and reduce errors.

Potential errors originating from the model can be addressed by providing administrators with access to fire extinguisher images to analyze them and enhance performance. This contributes to the system's reliability and competence and increases the safety levels of individuals and properties. Consequently, this would increase the trust of society in the safety inspection mechanisms used by the safety inspectors.

IV. CONCLUSION AND FUTURE WORK

This study addressed a critical issue in Saudi Arabia, the challenges faced by safety equipment inspectors. Taking advantage of computer vision and deep learning techniques, this research presented a system dedicated to inspecting fire extinguishers and identifying defects in real-time. To achieve this, a rich dataset was built, comprising 7,663 images with 16,092 instances of fire extinguisher defects, each labeled with classes like gauge bad, expiration date, gauge good, hose exist, pin exist, and rust. Subsequently, experiments were carried out, evaluating RT-DETR and different versions of the YOLO algorithm, including YOLOv5n, YOLOv7-tiny, and YOLOv8n. Among these models, the YOLOv8n algorithm emerged as the best performer, achieving a *mAP0.5* score of 87.2%. Therefore, this model was selected as the most suitable option for the proposed system due to its efficient inference time, making it ideal for real-time mobile applications. Although YOLOv5 outperformed YOLOv8n in terms of its compact model size, it is less favorable due to its long inference time. Additionally, the performance of YOLOv7-tiny did not provide a significant advantage over YOLOv8n. In contrast, RT-DETR exhibited significantly lower detection accuracy, larger model size, and longer inference times, further indicating its inability to identify fire equipment defects, especially in complex environments.

Despite its success, the limitations of the proposed system are acknowledged. The model was trained exclusively to identify specific types of extinguishers, namely, powder, foam, and water extinguishers, all of a particular size. Regarding the detection of expiration data, it is important to note that the model's current functionality is limited to identifying the presence of an expiration date on the extinguisher. However, it cannot determine whether the extinguisher has expired due to limitations arising from Python's library support for extracting expiration dates from Arabic language text. Future improvements would address this constraint. Regarding research directions, expanding training datasets to include other sizes and types of extinguishers, such as carbon dioxide gas extinguishers, water extinguishers, and others, would widen the scope of the model. The proposed approach is scalable, with the ability to evolve and adapt to detect new classes of defective safety equipment, including but not limited to electrical extensions and smoke detectors. These improvements would further enhance the model's capacity to identify safety equipment violations across a wide spectrum. Moreover, this system can be seamlessly integrated with a mobile application, thereby expanding its usability and effectiveness. Ultimately, this technological progress can make a substantial impact, propelling the improvement of safety inspection practices, not only in Saudi Arabia but also on a global scale.

REFERENCES

[1] "Annual Statistical Report," Civil Defense Directorate, Saudi Arabia, 2020.

[2] "Why is Fire Safety Important? | Alsco." https://alsco.com/resources/why-is-fire-safety-important/.

[3] "Saudi Fire Protection Code Fire Protection Requirements," Saudi Building Code National Committee, SBC 801, 2018.

[4] V. Kodur, P. Kumar, and M. M. Rafi, "Fire hazard in buildings: review, assessment and strategies for improving fire safety," *PSU Research Review*, vol. 4, no. 1, pp. 1–23, Jan. 2019, https://doi.org/10.1108/PRR-12-2018-0033.

[5] D. E. Della-Giustina, *Fire Safety Management Handbook*, 3rd ed. Boca Ration, FL, USA: CRC Press, 2014.

[6] D. Dieken, "Inspection, testing and maintenance of fire protection systems at industrial plants," *Process Safety Progress*, vol. 18, no. 3, pp. 151–155, 1999, https://doi.org/10.1002/prs.680180306.

[7] Y. J. Chen, Y. S. Lai, and Y. H. Lin, "BIM-based augmented reality inspection and maintenance of fire safety equipment," *Automation in Construction*, vol. 110, Feb. 2020, Art. no. 103041, https://doi.org/10.1016/j.autcon.2019.103041.

[8] Y. C. How, A. F. A. Nasir, K. F. Muhammad, A. P. P. A. Majeed, M. A. M. Razman, and M. A. Zakaria, "Glove Defect Detection Via YOLO V5," *MEKATRONIKA*, vol. 3, no. 2, pp. 25–30, 2021, https://doi.org/10.15282/mekatronika.v3i2.7342.

[9] S. H. Hsu, H. T. Hung, Y. Q. Lin, and C. M. Chang, "Defect inspection of indoor components in buildings using deep learning object detection and augmented reality," *Earthquake Engineering and Engineering Vibration*, vol. 22, no. 1, pp. 41–54, Jan. 2023, https://doi.org/10.1007/s11803-023-2152-5.

[10] "Madani Application," *AppStore*. https://apps.apple.com/sa/app/%D9%85%D8%AF%D9%86%D9%8A/id1596908770.

[11] "Salamti Application - Saudi Civil Defense," 2017. https://my.998.gov.sa/app/salamati.

[12] M. Anul Haq, "CNN Based Automated Weed Detection System Using UAV Imagery," *Computer Systems Science and Engineering*, vol. 42, no. 2, pp. 837–849, 2022, https://doi.org/10.32604/csse.2022.023016.

[13] M. Anul Haq, "Planetscope Nanosatellites Image Classification Using Machine Learning," *Computer Systems Science and Engineering*, vol. 42, no. 3, pp. 1031–1046, 2022, https://doi.org/10.32604/csse.2022.023221.

[14] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Computing and Applications*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023, https://doi.org/10.1007/s00521-023-08809-1.

[15] A. Jawaharlalnehru *et al.*, "Target Object Detection from Unmanned Aerial Vehicle (UAV) Images Based on Improved YOLO Algorithm," *Electronics*, vol. 11, no. 15, Jan. 2022, Art. no. 2343, https://doi.org/10.3390/electronics11152343.

[16] M. A. Haq, G. Rahaman, P. Baral, and A. Ghosh, "Deep Learning Based Supervised Image Classification Using UAV Images for Forest Areas Classification," *Journal of the Indian Society of Remote Sensing*, vol. 49, no. 3, pp. 601–606, Mar. 2021, https://doi.org/10.1007/s12524-020-01231-3.

[17] B. Xiao, M. Nguyen, and W. Q. Yan, "Fruit ripeness identification using YOLOv8 model," *Multimedia Tools and Applications*, Aug. 2023, https://doi.org/10.1007/s11042-023-16570-9.

[18] A. Corneli, B. Naticchia, M. Vaccarini, F. Bosché, and A. Carbonari, "Training of YOLO Neural Network for the Detection of Fire Emergency Assets," *ISARC Proceedings*, pp. 836–843, Oct. 2020.

[19] R. Kostoeva, R. Upadhyay, Y. Sapar, and A. Zakhor, "Indoor 3D Interactive Asset Detection Using a Smartphone," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2-W13, pp. 811–817, Jun. 2019, https://doi.org/10.5194/isprs-archives-XLII-2-W13-811-2019.

[20] H. Bayer and A. Aziz, "Object Detection of Fire Safety Equipment in Images and Videos Using Yolov5 Neural Network," in *Forum Bauinformatik*, Munich, Germany, 2022, pp. 62–69.

[21] J. H. Heinbach and A. Aziz, "Visual partial inspection of fire safety equipment using machine learning," presented at the 34th Forum Bauinformatik, Bochum, Germany, Sep. 2023, https://doi.org/10.13154/294-10096.

[22] B. G. Ferreira, B. G. Lima, and T. F. Vieira, "Visual Inspection of Collective Protection Equipment Conditions with Mobile Deep Learning Models," presented at the 1st International Conference on Deep Learning Theory and Applications, Feb. 2024, pp. 76–83, https://doi.org/10.5220/0009834600760083.

[23] H. Bichri, A. Chergui, and M. Hain, "Image Classification with Transfer Learning Using a Custom Dataset: Comparative Study," *Procedia Computer Science*, vol. 220, pp. 48–54, Jan. 2023, https://doi.org/10.1016/j.procs.2023.03.009.

[24] R. Szeliski, *Computer Vision: Algorithms and Applications*. Cham, Switzerland: Springer International Publishing, 2022.

[25] "Roboflow: Give your software the power to see objects in images and video." https://roboflow.com/.

[26] A. Gholamy, V. Kreinovich, and O. Kosheleva, "Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation," The Unicersity of Texas at El Paso, UTEP-CS-18-09, Feb. 2018. [Online]. Available: https://scholarworks.utep.edu/cs_techrep/1209.

[27] L. Petricca, T. Moss, G. Figueroa, and S. Broen, "Corrosion Detection Using A.I : A Comparison of Standard Computer Vision Techniques and Deep Learning Model," in *CS & IT Conference Proceedings*, May 2016, vol. 6.

[28] R. Rajamohanan and B. C. Latha, "An Optimized YOLO v5 Model for Tomato Leaf Disease Classification with Field Dataset," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12033–12038, Dec. 2023, https://doi.org/10.48084/etasr.6377.

[29] M. Salemdeeb and S. Erturk, "Multi-national and Multi-language License Plate Detection using Convolutional Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 5979–5985, Aug. 2020, https://doi.org/10.48084/etasr.3573.

[30] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," *Journal of Physics: Conference Series*, vol. 1004, no. 1, Dec. 2018, Art. no. 012029, https://doi.org/10.1088/1742-6596/1004/1/012029.

[31] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, https://doi.org/10.3390/make5040083.

[32] T. Saidani, "Deep Learning Approach: YOLOv5-based Custom Object Detection," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12158–12163, Dec. 2023, https://doi.org/10.48084/fetasr.6397.

[33] H. Gong *et al.*, "Swin-Transformer-Enabled YOLOv5 with Attention Mechanism for Small Object Detection on Satellite Images," *Remote Sensing*, vol. 14, no. 12, Jan. 2022, Art. no. 2861, https://doi.org/10.3390/rs14122861.

[34] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7464–7475.

[35] J. Solawetz, F. JAN 11, and 2023 12 Min Read, "What is YOLOv8? The Ultimate Guide.," *Roboflow Blog*, Jan. 11, 2023. https://blog.roboflow.com/whats-new-in-yolov8/.

[36] W. Lv *et al.*, "DETRs Beat YOLOs on Real-time Object Detection." arXiv, Jul. 06, 2023, https://doi.org/10.48550/arXiv.2304.08069.

[37] T. Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, Zurich, Switzerland, 2014, pp. 740–755, https://doi.org/10.1007/978-3-319-10602-1_48.