Enhancing the Quality of Ambulance Crew Work by detecting Ambulance Equipment using Computer Vision and Deep Learning

Jonab Hussain
College of Computing, Umm Al-Qura University, Saudi Arabia
s442017247@st.uqu.edu.sa

Nada Al-Masoody
College of Computing, Umm Al-Qura University, Saudi Arabia
s442004247@st.uqu.edu.sa

Asmaa Alsuraihi
College of Computing, Umm Al-Qura University, Saudi Arabia
s442000788@st.uqu.edu.sa

Fay Almogbel
College of Computing, Umm Al-Qura University, Saudi Arabia
s4420111499@st.uqu.edu.sa

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

Received: 7 May 2024 | Revised: 27 May 2024 | Accepted: 29 May 2024

ABSTRACT

Ambulance crews play an important role in responding quickly to emergencies and rescuing patients by providing appropriate treatment. Typically, fully equipped emergency vehicles are used to transport ambulance personnel to emergency locations. The ambulance crew cleans, sterilizes, and prepares equipment after each patient transfer with great care. Additionally, they check more than 70 pieces of equipment twice a day using a checklist, which is a tedious, time-consuming, and error-prone task. This study uses computer vision and deep learning techniques to replace the manual checklist process for medical equipment to assist the crew and make the equipment availability check faster and easier. To accomplish this, a dataset containing 2099 images of medical equipment in ambulances was collected and annotated with 3000 labeled instances. An experimental study compared the performance of YOLOv9-c, YOLOv8n, and YOLOv7-tiny. YOLOv8n demonstrated the best performance with a mAP50 of 99.2% and a speed of 3.3 ms total time per image. Therefore, YOLOv8 was selected for the proposed system due to its high accuracy and detection speed, which make it suitable for mobile applications. The presence of an application integrated with computer vision and deep learning technologies in paramedic devices can assist in reviewing the equipment checklist, reducing human errors, speeding up the review process, and alleviating the burden on paramedics in their work.

Keywords-Medical Equipment Detection, Ambulance Equipment Detection, Computer Vision, Deep Learning, You-Only-Look-Once (YOLO)

I. INTRODUCTION

An ambulance is a means of transporting patients in critical condition to the hospital and is essential in any Emergency Medical System (EMS). Ambulance crews provide care and transportation to patients in need, often during the most crucial moments of their lives. Ambulances are equipped with all the necessary equipment to treat patients. Therefore, paramedics and prioritized personnel must ensure the readiness and maintenance of this equipment to confirm its availability and use when needed, as it plays a significant role in preserving human life. After each patient transfer, the equipment should be cleaned, sterilized, and prepared for the next usage. In Saudi Arabia, the Red Crescent Authority made an effort to automate the equipment checking process through the Electronic Medic application, where the ambulance crew logs into the system and navigates to the reference list displaying a comprehensive equipment inventory divided into four sections, comprising over 100 types of equipment. The equipment in each section is checked for its availability and quantity to be determined. In an interview conducted on December 7, 2023, the Saudi Red Crescent Society stated that the equipment check process usually takes between 20 and 25 minutes for experienced personnel, but it may require more time for beginners. This study deploys computer vision and deep learning techniques to assist and facilitate the equipment check process.

Object detection and recognition employ computer vision, image processing, and deep learning [1]. In [2], a low-cost computer vision approach was presented to track surgical tools. Deep learning algorithms show great promise in extracting features and patterns from complex medical data, such as detecting cancer and eye diseases [3-5]. In [6], a system was presented to accurately identify surgical tools, namely curved and straight forceps, with an accuracy exceeding 99%. In [7], the YOLOv7x algorithm was applied to detect surgical tools. This study introduced the Recurrent Field-of-View Kernel (RepLK) module and the Object Detection Convolution (ODConv) structure to enhance feature extraction and recognition accuracy, addressing the challenges of identifying similar-looking, occluded, or densely arranged tools. This algorithm achieved precise surgical tool detection, with experimental results demonstrating higher accuracy, with F1, AP, AP50, and AP75 reaching 94.7%, 91.5%, 99.1%, and 98.2%, respectively.

Adopting computer vision and deep learning techniques to detect ambulance equipment could enhance the quality of work of ambulance crews. Although computer vision and deep learning have revolutionized various fields, their application within ambulance equipment detection remains largely unexplored. This study aims to fill this gap and unleash the potential of these technologies to enhance ambulance crew performance and patient care, making a unique contribution to the EMS field by exploring the potential of computer vision and deep learning. The staff simply scans the equipment using the camera on their phones or tablets, which then evaluates its availability and quantity and alerts them if any equipment is not detected. Additionally, the staff can manually check the equipment. The proposed solution can reduce the likelihood of human error, facilitate equipment management and tracking, and save time and effort for the ambulance crew, allowing them to focus on providing immediate medical care to patients. This study aimed to:

- Create, collect, and annotate a dataset for emergency medical equipment.
- Conduct a comparative analysis to evaluate the performance of three object detection models, namely YOLOv9-c, YOLOv8n, and YOLOv7-tiny.

II. DATA COLLECTION AND METHOD

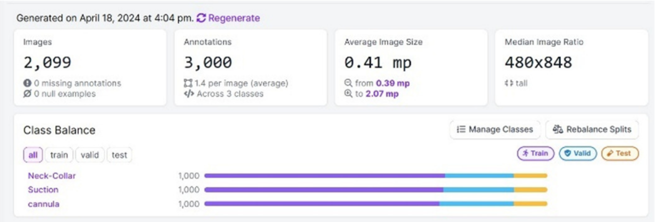
A. Dataset Collection

A dataset was collected in collaboration with the ambulance department at King Abdullah Medical City in Mecca, Saudi Arabia. Using mobile phones, 15-20 minutes of video footage was recorded on all ambulance equipment from different angles and lighting conditions, and then every second was converted to one or two images using Roboflow [8]. Due to time and resource constraints, three pieces of equipment were chosen to train the models. Sufficient data were available from all angles and under all lighting conditions to ensure that the model was robust to real-world variations. The selection of ambulance equipment was based on the following criteria:

- The importance of equipment in saving lives:
    - The neck collar used to support and stabilize the neck in cases of injury
    - The suction used to remove fluid from the airways in resuscitation cases
    - The cannula used to deliver fluids or collect samples in emergencies.
- Possibility of losing equipment outside the ambulance:
    - The suction device and cannula are easily transportable and easy to lose outside the vehicle.
- Diversity of equipment sizes:
    - The cannula is small in size
    - The neck collar and suction device are larger.
- The necessity of having a certain number of pieces of equipment:
    - The cannula and collar are basic pieces of equipment that must be available in certain numbers in the ambulance.

B. Dataset Annotation

The annotation process involved analyzing video clips recorded inside the King Abdullah Medical City ambulance in Mecca using Roboflow [8]. The clips were converted to image frames to identify instances falling into three categories: Cannula, Neck-collar, and Suction. Figure 1 depicts the dataset comprising 2099 images with 3000 instances of the specified ambulance equipment. Then, 1000 instances were annotated for each of the three categories, totaling 3000 instances. Figure 2 portrays a sample of the annotated images.



Generated on April 18, 2024 at 4:04 pm.

Images
2,099
0 missing annotations
0 null examples

Annotations
3,000
1.4 per image (average)
Across 3 classes

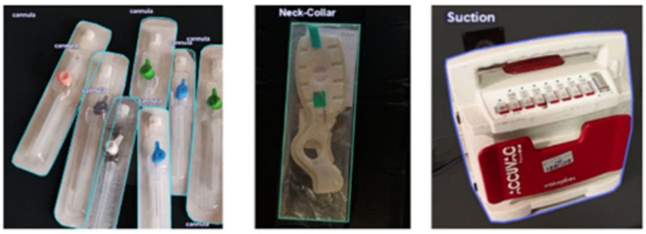
Average Image Size
0.41 mp
from 0.39 mp
to 2.07 mp

Median Image Ratio
480x848
tall

Class Balance

Neck-Collar	1,000
Suction	1,000
cannula	1,000

Fig. 1. Dataset health check.



(a) (b) (c)

Fig. 2. Dataset sample: (a) Cannula, (b) Neck-collar, (c) Suction.

C. YOLO Algorithms

YOLO (You Only Look Once) was presented in 2016 [9], redefining object detection as a single-pass regression task, starting from image pixels and progressing to bounding box and class probabilities. This method allowed concurrent prediction of several bounding boxes and class probabilities, enhancing speed and accuracy [10]. This study selected the most recent versions of the algorithm: YOLOv7, YOLOv8, and YOLOv9. YOLOv7 introduced an Efficient Layer Aggregation Network (ELAN) architecture, which improves self-learning without affecting the original gradient [11]. YOLOv7-tiny simplifies the architecture for peripheral GPUs, sacrificing some accuracy for speed and weight advantages [12]. YOLOv8, released in 2023, is a high-performance object detection model in the YOLO series. It features improved components, such as the backbone, neck, and head, to efficiently detect and localize objects [13]. The model includes five versions, differing in network depth and width, with YOLOv8n being the fastest model for accurate target detection [14]. YOLOv9, released in 2024, has two versions: YOLOv9-e and YOLOv9-c. YOLOv9-c is lighter. It includes Programmable Gradient Information (PGI) and GELAN to improve model efficiency. PGI helps to manage gradient info propagation [15], while GELAN combines Cross Stage Partial Network (CSPNet) and ELAN to improve info integration and efficiency in model training, offering benefits such as lightweight, speed, and accuracy [16].

D. Training Methodology

This study aims to enhance the ambulance checklist review process, ensuring a compact size suitable for mobile applications. Experiments with various versions of the YOLO algorithm were carried out, including YOLOv7-tiny, YOLOv8n, and YOLOv9-c. These models were trained on a dataset of 2099 images with 3000 instances of equipment in the ambulance vehicle, each labeled with classes Neck-Collar, Suction, or Cannula. 1463 images were utilized for training, 430 images for validation, and 206 images for testing. The dataset was insufficient, so transfer learning was applied to fine-tune pre-trained model weights on new data. Thus, all previously mentioned models were pre-trained on the Common Objects in Context (COCO) dataset [17], one of Microsoft's largest datasets [17]. Table I details the hyperparameters.

TABLE I. MODEL HYPERPARAMETERS

Models	Optimizer	Learning Rate	Batch Size	Epochs
YOLOv7-tiny	Adam	0.001	16, 32,64	25, 50
YOLOv8n	Adam	0.001	16, 64	25, 30, 50
YOLOv9-c	SGD	0.01	8, 16	25, 35, 50

E. Tracking And Counting Methodology

The tracking and counting of ambulance equipment deploying the YOLO object detection models was custom-trained on the selected ambulance equipment because it gave optimal results. The ByteTrack multi-object tracking algorithm was used, as it is an efficient real-time object tracking algorithm designed for video sequences. ByteTrack is a simple, efficient, and versatile method for data association [18]. The algorithm links tracklets with high-score detection boxes, employing an one-shot detection-based approach that integrates object tracking and detection into a unified model. ByteTrack achieves high tracking speed by sharing computational resources between detection and tracking [18]. Ultralytics and supervision libraries were engaged to draw frames and label class names. A line was generated using the LineCounter and LineCounterAnnotator library to count objects that surpass this line, as evidenced in Figure 3.

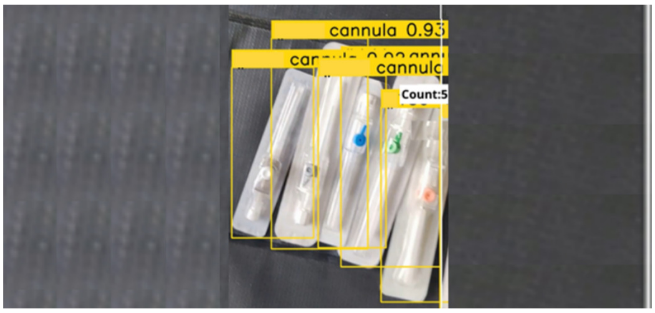


Fig. 3. Example of counting the ambulance equipment.

F. Training Environment

Model training requires high computational resources, such as GPUs. This study utilized Google Colab for model training, which is a cloud platform to run Python code. Training was carried out on resources provided by Google Colab for free, a T4 GPU with 12GB VRAM. The Colab Pro paid version was used for the slower and more complex model YOLOv9-c, offering access to more powerful GPUs, including the faster GPU L4, to accelerate the training process.

G. Evaluation Metrics

Basic metrics were put into service to evaluate the models' performance in the detection task and gauge their effectiveness in recognizing and classifying small objects. The primary metrics deployed were Average Precision (AP) and mean Average Precision (mAP). These were determined by comparing the algorithm output with the actual object labels of the image to assess the algorithm's accuracy for each feature class using (1) and (2). Precision measures the ratio of true predictions to total optimistic predictions (3), and recall measures the ratio of true positives to the number of positive ground truths (4). The F1-Score considers both precision and recall by calculating their harmonic mean (5) and helps to evaluate the balance between them [19].

$$\mathrm{AP} = \frac{1}{\mathrm{classes}} \sum_{\mathrm{R_i}} \mathrm{PR_i} \quad (1)$$

$$\mathrm{mAP} = \frac{1}{\mathrm{classes}} \sum_{\mathrm{i=1}}^{\mathrm{classes}} \mathrm{AP_i} \quad (2)$$

These metrics were computed utilizing a confusion matrix comprising four components: True Positives (TP) indicate the number of instances correctly classified as positive by the model from the positive class, False Positives (FP) denote the number of instances incorrectly classified as positive by the model from the negative class, True Negatives (TN) is the number of instances correctly classified as negative by the model from the negative class, and False Negatives (FN) is the number of instances incorrectly classified as negative by the model from the positive class. Precision, recall, and F1-Score were calculated by:

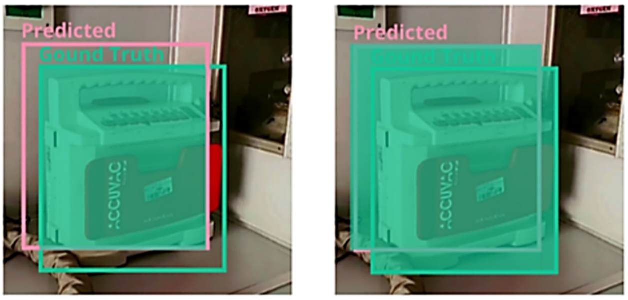
$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (3)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (4)$$

$$\text{F1} - \text{Score} = \frac{2 \times \text{P} \times \text{R}}{\text{R+P}} \quad (5)$$

Intersection over union (IoU) is another concept for object detection [20]. This metric measures the accuracy of each bounding box by taking the ratio of the overlapping areas between the actual (B_{gt}) and predicted (B_{pr}) bounding boxes to their union area. Equation (6) shows its formula and Figure 4 represents its application.

$$\text{IoU} = \frac{\text{B}_{\text{pr}} \cap \text{B}_{\text{gt}}}{\text{B}_{\text{pr}} \cup \text{B}_{\text{gt}}} \quad (6)$$



(a) (b)

Fig. 4. IoU is the ratio of the intersection area over the union area: (a) intersection area, (b) union area.

III. RESULTS AND DISCUSSION

A comparative analysis was performed after several experiments to evaluate the selected models and determine their optimal performance in various aspects.

A. *Evaluation Matrix And Model Size*

Tables II-IV illustrate the evaluation metrics Precision (P), Recall (R), and Mean Average Precision (mAP) for YOLOv7-tiny, YOLOv8n, and YOLOv9-c with various hyperparameters, including learning rate (*Lr*), batch size, and epochs. Figure 5 displays a comparison of YOLOv7-tiny, YOLOv8n, and YOLOv9-c based on mAP0.5. Hyperparameters for YOLOv8n and YOLOv7-tiny were 50 epochs, 0.001 learning rate, and 64 batch size. The hyperparameters for YOLO9-c were different, as this model was heavy for the available GPU, so epochs were set to 50, the learning rate was set to 0.01, and batch size was set to 16. YOLOv8n had the best performance, YOLOv7-tiny was close, and YOLOv9-c had the worst results.

TABLE II. EVALUATION OF YOLOV7-TINY

Size: 12.3 MB			
Hyperparameters	**P (%)**	**R (%)**	**mAP50(%)**
Lr = 0.001	96.9	96.2%	97.2
Batch size = 32			
Epochs =50			
Lr = 0.001	95.4	95.3%	96.7
Batch size = 64			
Epochs =50			
Lr = 0.001	76.8	78.6	79.2
Batch size = 16			
Epochs =25			

TABLE III. EVALUATION OF YOLOV8N

Size: 6.3 MB			
Hyperparameters	**P (%)**	**R (%)**	**mAP50(%)**
Lr = 0.001	98.7	98.7	99.2
Batch size = 64			
Epochs =50			
Lr = 0.001	98.7	97.8	98.7
Batch size = 16			
Epochs =30			
Lr = 0.001	98.5	98.3	99.1
Batch size = 16			
Epochs =25			

TABLE IV. EVALUATION OF YOLOV9-C

Size: 51.5 MB			
Hyperparameters	**P (%)**	**R (%)**	**mAP50(%)**
Lr = 0.01	88.5	47.0	59.1
Batch size = 16			
Epochs =50			
Lr = 0.01	88.5	47.0	59.1
Batch size = 16			
Epochs =35			
Lr = 0.01	83.1	42.9	59.3
Batch size = 8			
Epochs =25			

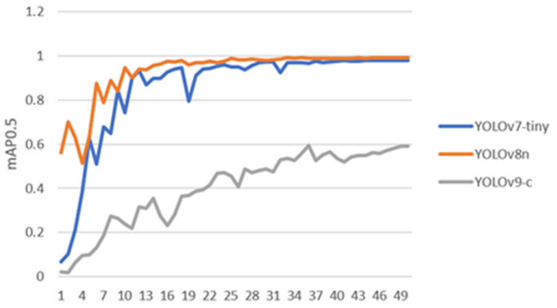


Fig. 5. mAP0.5 of the YOLO models examined under 50 epochs.

B. *Confusion Matrix*

Figures 6-8 disclose the confusion matrices of the models, evaluating their performance and summarizing how well each one performs in distinguishing different classes. Confusion matrices go beyond a simple assessment of correct or incorrect, revealing exactly how often a model makes certain types of errors [21]. Based on the confusion matrices, YOLOv8n was the best, while YOLOv7-tiny was close. However, YOLOv9-c had the lowest performance due to its heavy size, which affected its accuracy.

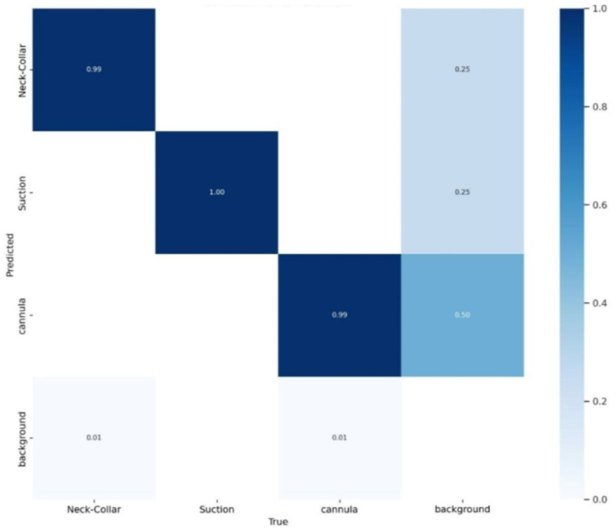


Fig. 6. Confusion matrix of YOLOv7-tiny.

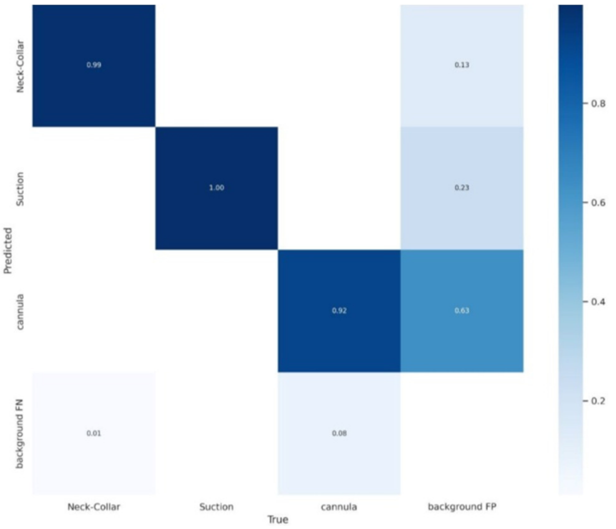


Fig. 7. Confusion matrix of YOLOv8n.

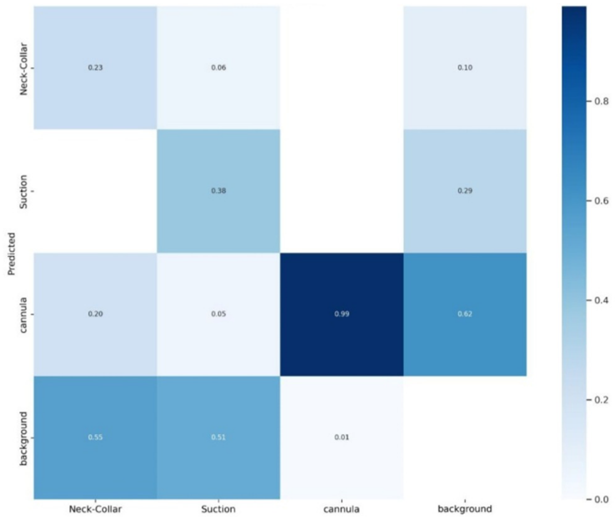


Fig. 8. Confusion matrix of YOLOv9-c.

C. F1-Confidence, Precision-Confidence, Precision-Recall Curve, and Recall Confidence.

Figure 9 exhibits several performance curves of YOLOv7-tiny, demonstrating its strong performance in various evaluation metrics. The F1 confidence curve shows an F1-Score of 0.97 at a confidence threshold of 0.474, indicating a good balance between precision and recall. The precision confidence curve displays a precision of 1.00 at 0.869 confidence, suggesting accurate positive predictions. The precision-recall curve displays an impressive mAP of 0.979 at a confidence threshold of 0.5, demonstrating the model's overall effectiveness in detecting objects. The recall confidence curve manifests a perfect recall of 0.99 at 0.000 confidence, showing that the model successfully identifies almost all positive instances.

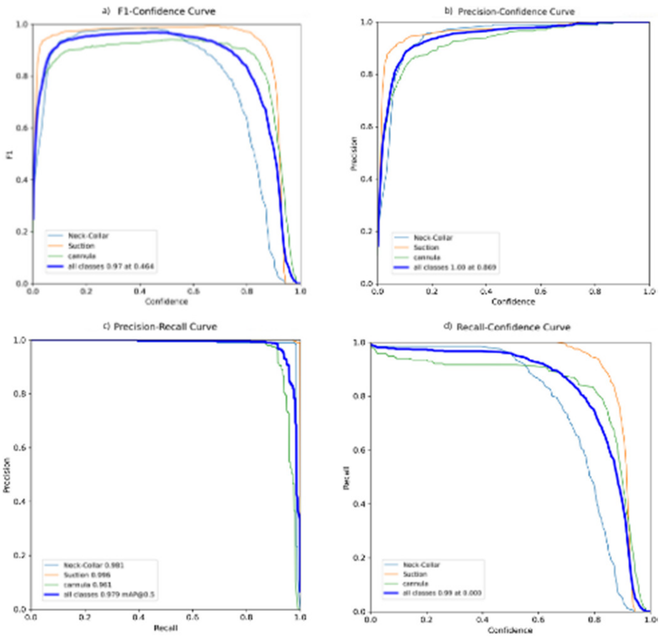


Fig. 9. YOLOv7-tiny: (a) F1-confidence, (b) Precision-confidence curve, (c) Precision-recall curve, and (d) Recall confidence.

Figure 10 portrays the performance of the YOLOv8n model, which had even better performance metrics. The F1 confidence curve shows an excellent F1-Score of 0.99 at 0.513 confidence, indicating a high balance between precision and recall. The precision confidence curve demonstrates a precision of 1.00 at 0.941 confidence, suggesting accurate positive predictions. The precision-recall curve displays an impressive mAP of 0.992 at a confidence threshold of 0.5, demonstrating the model's overall effectiveness in object detection. The recall confidence curve exhibits a perfect recall of 1.00 at 0.000 confidence, indicating that the model did not miss any positive instances.

Figure 11 depicts the performance of the YOLOv9-c model. This model had lower performance in terms of accuracy and overall effectiveness. The F1 confidence curve manifests an F1-Score of 0.73 at 0.591 confidence, indicating a relatively lower accuracy in balancing precision and recall compared to the other models. However, the precision confidence curve exhibits a precision of 1.00 at 0.981 confidence, indicating accurate positive predictions. The precision-recall curve discloses an mAP of 0.771 at a confidence threshold of 0.5, suggesting a lower overall effectiveness. The recall confidence curve presents a recall of 0.96 at 0.000 confidence, suggesting a high sensitivity but with missed positive instances.

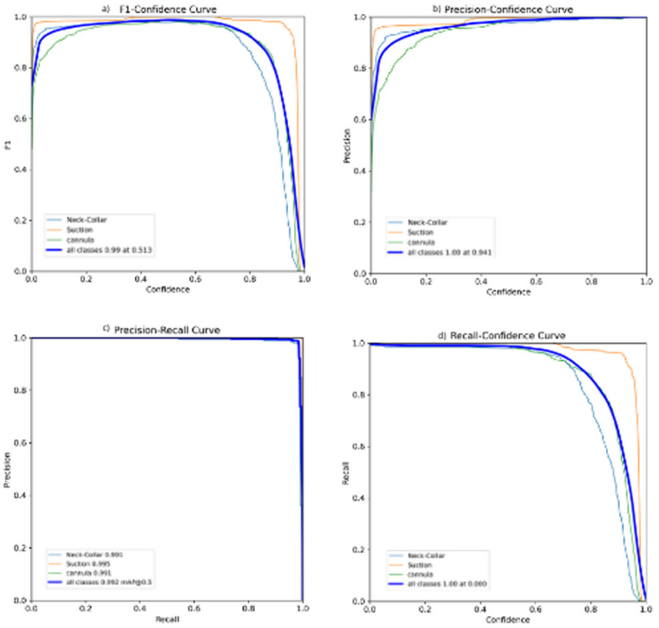


Fig. 10. YOLOv8-n: (a) F1-confidence, (b) Precision confidence curve, (c) Precision-Recall curve, and (d) Recall confidence curve.

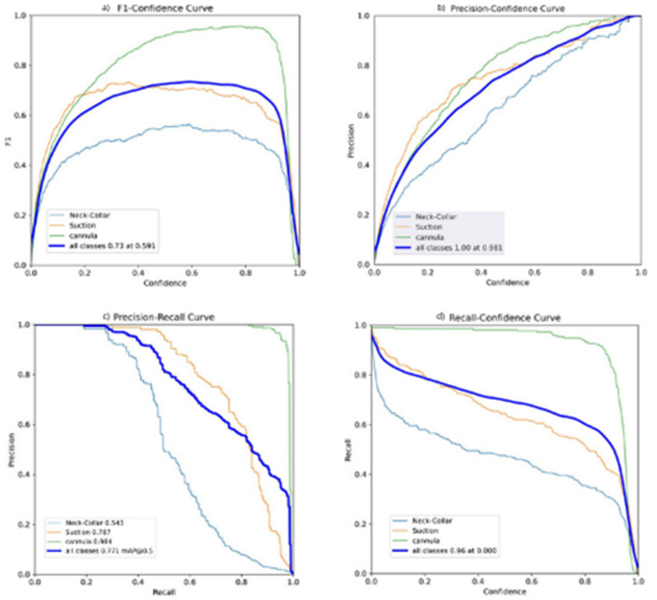


Fig. 11. YOLOv9-c: (a) F1-confidence, (b) Precision-Confidence curve, (c) Precision-Recall curve, and (d) Recall Confidence of YOLOv9-c.

D. Speed

The proposed solution can help ensure the availability of ambulance equipment and accelerate the process by implementing it in real-time. Table V illustrates the time distribution in two key phases: inference (passing the image through the neural network) and post-processing. YOLOv8n had the fastest time (3.3 ms) due to its lightweight nature, whereas YOLOv9n was the slowest (19.7 ms) due to its large size.

TABLE V. DETECTION TIME OF MODELS USING L4 GPU

Model	Inference (ms)	Postprocessing (ms)	Total time (ms)
YOLOv7-tiny	3.4	2.3	5.7
YOLOv8n	1.4	1.9	3.3
YOLOv9-c	16.0	3.7	19.7

E. Discussion

Figure 12 provides the results based on mAP for YOLOv7-tiny, YOLOv8n, and YOLOv9-c. The mAP50 result of YOLOv7-tiny (97.2%) was close to that of YOLOv8n (99.2%). On the other hand, YOLOv9 showed unsatisfactory mAP50 (59.3%) due to its large size (51.5 MB). Currently, there are only two versions of this model available, YOLOv9-c and YOLOv9-e. The smaller version was tested, but the remaining sizes of this model are not available yet.

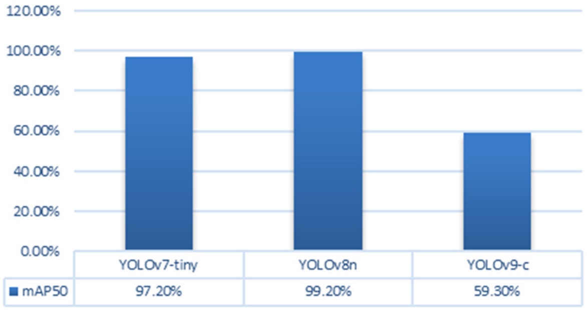
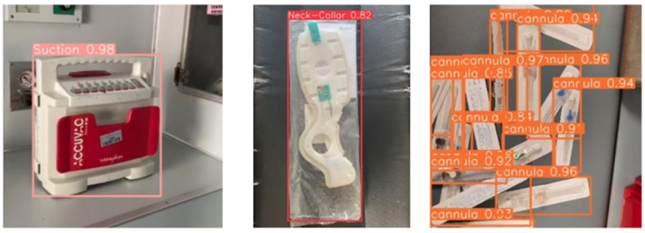


Fig. 12. Comparison of mAP50 for each model.

Table VI displays the performance of YOLOv8n in each class. YOLOv8n is the optimal model for light equipment detection, as its lightweight (6.3 MB) and fast detection speed (3.3 ms) make it suitable for mobile applications. It achieved a mAP50 result of 99.2%. The model exhibited high results in all three classes, as they differ greatly in terms of sizes, colors, and shapes, which positively affects its performance. The model was trained and tested on one form of equipment in each class, currently used by the Red Crescent Authority, contributing to its performance. In the Suction class, it outperformed the others with an mAP50 score of 99.5%, attributed to its larger and clearer characteristics compared to the Neck-collar and Cannula classes. Both of the latter are usually contained in plastic bags, which could explain the relatively lower results compared to those obtained in the former class. Figure 13 illustrates a sample of the test results.

TABLE VI. PERFORMANCE OF YOLO8N IN EACH CLASS

Class	P (%)	R (%)	mAP50(%)
All	0.987	0.987	0.992
Neck-Collar	0.978	0.98	0.991
Suction	0.997	1	0.995
Cannula	0.986	0.981	0.991



(a) (b) (c)

Fig. 13. Sample of testing results using YOLOv8n: (a) Suction, (b) Neck-collar, (c) Cannula.

IV. CONCLUSION AND FUTURE WORK

This study presented a solution to help ambulance crews confirm the availability and count of essential ambulance equipment to accelerate the checklist review process using computer vision and deep learning. A dataset was built, containing 2099 images with 3000 instances of equipment in an ambulance vehicle, each labeled as Neck-collar, Suction, or Cannula. 1463 images (70%) were utilized for training, 430 (20%) for validation, and 206 (10%) for testing. Three deep-learning object detection models were evaluated, namely YOLOv9, YOLOv8n, and YOLOv7-tiny. The results showed that YOLOv8n achieved the best performance, with 99.2% in mAP50, a compact model size of 6.3 MB, and a fast inference speed of 3.3 ms, making it suitable for mobile applications. YOLOv7-tiny achieved close results to YOLOv8n, with 97.2% in mAP50, a compact model size of 12.3 MB, and an inference speed of 5.7 ms. YOLOv9-c was less favorable due to its lower detection accuracy of 59.3% mAP50, a model size of 51.5 MB, which is considered heavy, and a slower inference speed of 19.7 ms. Despite the successful implementation of the proposed solution, there are a few limitations:

- The models were trained on only three types of equipment and one form of ambulance vehicle equipment. Therefore, the models' current functionality is limited to detecting and counting the three classes of equipment they were trained on. Expanding the classes to include more ambulance equipment would be desirable.
- The dataset size was small, which is considered a limitation that can affect the performance of the models. Increasing the dataset size could improve their accuracy.

Future work could involve expanding the dataset to include most or all ambulance equipment in various forms to improve the efficiency of the system. This model could be used under human supervision, being integrated into a mobile application to scan ambulance equipment through the camera to assess availability and quantity, and so assist staff. On the other hand, to ensure that the scan process runs safely and avoid potential risks, a confirmation message should appear for any undetected device, prompting the user to re-scan it. Paramedics can manually inspect equipment and include notes (e.g., expiration dates, etc.). They can also access the history of all checklist processes and review them. Furthermore, enhancing the model could involve adding a feature to verify the readiness status of the equipment.

REFERENCES

[1] A. Vahab, M. S. Naik, P. G. Raikar, and S. R. Prasad, "Applications of object detection system," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 4, pp. 4186–4192, Apr. 2019.

[2] R. Dockter, R. Sweet, and T. Kowalewski, "A fast, low-cost, computer vision approach for tracking surgical tools," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, Sep. 2014, pp. 1984–1989, https://doi.org/10.1109/IROS.2014.6942826.

[3] C. Cao *et al.*, "Deep Learning and Its Applications in Biomedicine," *Genomics, Proteomics & Bioinformatics*, vol. 16, no. 1, pp. 17–32, Feb. 2018, https://doi.org/10.1016/j.gpb.2017.07.003.

[4] V. A. Rajendran and S. Shanmugam, "Automated Skin Cancer Detection and Classification using Cat Swarm Optimization with a Deep Learning Model," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12734–12739, Feb. 2024, https://doi.org/10.48084/etasr.6681.

[5] R. Gurumoorthy and M. Kamarasan, "Breast Cancer Classification from Histopathological Images using Future Search Optimization Algorithm and Deep Learning," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12831–12836, Feb. 2024, https://doi.org/10.48084/etasr.6720.

[6] B. Chan et al., "Computer vision for object detection; machine learning-based identification of surgical equipment," *Archives of Disease in Childhood*, vol. 104, no. 4, Nov. 2019 https://doi.org/10.1136/archdischild-2019-gosh.75.

[7] B. Ran, B. Huang, S. Liang, and Y. Hou, "Surgical Instrument Detection Algorithm Based on Improved YOLOv7x," *Sensors*, vol. 23, no. 11, Jan. 2023, Art. no. 5037, https://doi.org/10.3390/s23115037.

[8] "Roboflow: Computer vision tools for developers and enterprises," https://roboflow.com/.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.

[10] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," *Machines*, vol. 11, no. 7, Jul. 2023, Art. no. 677, https://doi.org/10.3390/machines11070677.

[11] A. Alayed, R. Alidrisi, E. Feras, S. Aboukozzana, and A. Alomayri, "Real-Time Inspection of Fire Safety Equipment using Computer Vision and Deep Learning," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13290–13298, Apr. 2024, https://doi.org/10.48084/etasr.6753.

[12] L. Zhang, N. Xiong, X. Pan, X. Yue, P. Wu, and C. Guo, "Improved Object Detection Method Utilizing YOLOv7-Tiny for Unmanned Aerial Vehicle Photographic Imagery," *Algorithms*, vol. 16, no. 11, Nov. 2023, Art. no. 520, https://doi.org/10.3390/a16110520.

[13] S. Li, F. Chen, Z. Sun, Z. Zhu, L. Zhou, and K. Tang, "Research on YOLOv8 Object Detection Algorithm in UAV Scenarios." Mar. 01, 2024, https://doi.org/10.21203/rs.3.rs-3995816/v1.

[14] K. Lan, X. Jiang, X. Ding, H. Lin, and S. Chan, "High-Efficiency and High-Precision Ship Detection Algorithm Based on Improved YOLOv8n," *Mathematics*, vol. 12, no. 7, Jan. 2024, Art. no. 1072, https://doi.org/10.3390/math12071072.

[15] C. T. Chien, R. Y. Ju, K. Y. Chou, and J. S. Chiang, "YOLOv9 for Fracture Detection in Pediatric Wrist Trauma X-ray Images." arXiv, May 27, 2024, https://doi.org/10.48550/arXiv.2403.11249.

[16] C. Y. Wang, I. H. Yeh, and H. Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information." arXiv, Feb. 28, 2024, https://doi.org/10.48550/arXiv.2402.13616.

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

[18] M. Abouelyazid, "Comparative Evaluation of SORT, DeepSORT, and ByteTrack for Multiple Object Tracking in Highway Videos," *International Journal of Sustainable Infrastructure for Cities and Societies*, vol. 8, no. 11, pp. 42–52, Nov. 2023.

[19] L. D. Quach, K. Nguyen, A. N. Quynh, and H. T. Ngoc, "Evaluating the Effectiveness of YOLO Models in Different Sized Object Detection and Feature-Based Classification of Small Objects," *Journal of Advances in Information Technology*, vol. 14, no. 5, pp. 907–917, Sep. 2023, https://doi.org/10.12720/jait.14.5.907-917.

[20] A. Sharma, A. Z. Ansari, R. Kakulavarapu, M. H. Stensen, M. A. Riegler, and H. L. Hammer, "Predicting Cell Cleavage Timings from Time-Lapse Videos of Human Embryos," *Big Data and Cognitive Computing*, vol. 7, no. 2, Jun. 2023, Art. no. 91, https://doi.org/10.3390/bdcc7020091.

[21] G. Gledec, M. Sokele, M. Horvat, and M. Mikuc, "Error Pattern Discovery in Spellchecking Using Multi-Class Confusion Matrix Analysis for the Croatian Language," *Computers*, vol. 13, no. 2, Feb. 2024, Art. no. 39, https://doi.org/10.3390/computers13020039.