World Journal of Engineering Research
ISSN: 2959-9873
DOI: https://doi.org/10.61784/wjer3017

EXPLORATION OF THE DETECTION METHOD OF THE CONDITION OF PATIENTS ON THE AMBULANCE STRETCHER

XinLei Yang, JueXiao Chen*
School of Automotive Engineering, Tongji University, Shanghai 200092, China.
Corresponding Author: JueXiao Chen, Email: chenjuexiao@tongji.edu.cn

Abstract: To improve the efficiency of ambulance emergency task scheduling and address the issues of low efficiency and poor real-time performance in traditional manual entry of stretcher status, this paper proposes an automatic judgment method for ambulance emergency status based on deep learning. A self-built ambulance stretcher dataset is constructed by combining 567 images captured independently from ambulances and 545 images obtained via web crawling. Data augmentation techniques are used to enhance model robustness, and mainstream object detection algorithms such as TOOD, Faster R-CNN, and YOLOv8 are compared and analyzed. Experimental results show that YOLOv8 achieves an average precision (AP@0.5) of 0.779 at an IoU threshold of 0.5, with only 11.2 million parameters, significantly outperforming other models. This method can accurately and real-time determine the stretcher status, providing a reliable basis for emergency centers to dynamically dispatch ambulance resources. It effectively shortens the emergency response time and improves the success rate of emergency rescue.
Keywords: Ambulance emergency status; Object detection; Deep learning; Data augmentation

1 INTRODUCTION

Improving the utilization rate of ambulances in service is crucial for enhancing emergency rescue success. Upon receiving a task, ambulances must reach the scene promptly to provide treatment within the critical "golden time." For instance, cardiac arrest patients require a response time of less than 10 minutes, and early defibrillation maximizes cardiopulmonary resuscitation success[1]. Studies show that the survival rate for out-of-hospital cardiac arrest exceeds 10% in Europe and the United States, while it remains at 1% in China, primarily due to differences in emergency response times. Developed countries such as Denmark and Japan achieve response times of approximately 3 minutes, whereas the United States and Russia report 4–6 minutes[2]. For emergency dispatch centers, real-time knowledge of whether a stretcher is occupied is essential for assigning subsequent tasks. Once a stretcher is detected as unoccupied after patient transfer, the ambulance becomes available for new assignments. With advancements in IoT-enabled ambulances, onboard data can now be transmitted seamlessly to emergency centers[3].

Current domestic ambulance systems primarily rely on manual input of stretcher status, which is inefficient, subjective, and increases workload for emergency personnel. Recent developments in industrial automation and electronic technologies have facilitated the adoption of status detection methods. For example, Gao Wenxuan et al. proposed leveraging idle computing resources from vehicles to alleviate edge server load[4]. Common detection methods, such as gravity sensors and infrared detection, face challenges in installation complexity, high costs, and low accuracy. In contrast, deep learning-based detection offers advantages in speed, convenience, and cost-effectiveness, making it ideal for ambulance stretcher status detection.

This study focuses on deep learning algorithms for ambulance stretcher status detection, utilizing a self-constructed dataset to achieve automatic judgment of emergency status.

2 AMBULANCE STRETCHER DATASET CONSTRUCTION

The collection and construction of a dataset is a critical step in the deep learning process, as the scientific validity and representativeness of the dataset directly determine the accuracy and reliability of the final detection results. During the training and validation phases of a deep learning model, the dataset used must align with real-world application scenarios, especially for the task of detecting ambulance stretcher images. To ensure the model can operate effectively in real environments, the constructed ambulance stretcher dataset needs to exhibit diversity. This means the dataset should include multi-angle images to capture the features of stretchers from different perspectives; it should also cover various lighting conditions to simulate the effects of different illumination environments on images; additionally, the dataset must include diverse scenarios to reflect the real-world applications of ambulances in different environments.

The ambulance stretcher dataset used in this experiment consists of two parts. The first part is real-time collection of stretcher images through cameras and digital video recorders installed on ambulances. The second part involves using web crawler programs to obtain relevant ambulance stretcher images from the Internet. Combining these two methods can construct a more comprehensive and scientific dataset, providing a solid foundation for subsequent deep learning model training.

2.1 Dataset Shooting

Cameras selected for ambulances are usually small in size, mainly spherical cameras, which save space and better adapt to the internal environment of ambulances. These cameras must also have night vision functions to ensure clear capture of surrounding images under low-light conditions. Ambulances themselves need to have video storage capabilities to record and save key monitoring video data, ensuring that relevant materials can be retrieved at any time in emergency situations.

Considering the actual shooting scenario of ambulances, the self-collected image data mainly uses Dahua brand network cameras and digital video recorders. Two camera models were selected: DH-IPC-HDBW3233F-M-AS and DH -IPC-HDW1020C. Both models have high resolution and good low-light performance, providing clear image quality in various environments. For digital video recorders, the DH-MNVR4104-GFW model was used, which supports multi-channel video input and has strong video storage capacity to meet the need for long-term video recording. Through these high-performance devices, 567 stretcher images during ambulance missions were effectively collected, providing reliable data for subsequent analysis and model training.

2.2 Dataset Crawling

Limited by equipment, vehicles, and operating environments, self-shot ambulance stretcher images are relatively single and have certain limitations in application. To obtain more extensive stretcher images from various ambulances, regions, and camera types as data sources, web crawlers were used to acquire the required images.

Web crawlers are automated tools that search for target information on the Internet through specific rules and algorithms. Driven by the continuous upgrading of Internet infrastructure, web crawler technology has evolved exponentially, forming a diversified technical ecosystem. Search engine crawlers focus on collecting and indexing web content, allowing users to quickly find needed information through search engines; image crawlers specifically target image data to obtain specific types of visual information. The design and implementation of these crawlers are based on user requirements. Developers customize programs according to specific user needs to ensure that crawlers can efficiently and accurately acquire the required data. Due to the urgent demand for dataset acquisition, web crawling technology has been widely adopted. Some developers have customized crawler tools into applications such as WebHarvy, Larbin, and Web Scraper. These tools possess distinct functionalities and features, enabling users to perform data scraping in diverse scenarios. However, such fixed-format crawler tools often fail to directly meet the needs of specific research projects, particularly when handling complex crawling targets. This may result in the inclusion of non-target image categories in the scraped results, thereby compromising data quality and reliability.

To address these needs, a flexible crawler architecture was independently developed using Python 3.8 and its request module. The request module is a library for sending HTTP requests, with a relatively simple API and good stability. This approach enables more flexible design of crawler programs to adapt to different needs and complex crawling targets. Through web crawlers, 545 ambulance stretcher images were successfully captured from major search engines such as Baidu and Google.

2.3 Data Augmentation

To improve the final model's performance, stability, and adaptability to new data, data augmentation technology was used to expand the original training set. The Augly framework was selected as the implementation tool, which is a data augmentation library focused on adversarial robustness, providing a wide range of augmentations for multiple modalities (audio, image, text, and video) [5].

The Augly framework was developed by the AI team at Facebook (now Meta). This framework enhances model generalization by generating extensive variant data, enabling models to learn broader patterns. During data augmentation, Augly employs a diverse range of transformation strategies, including image blurring, random noise addition, grayscale conversion, random rotation, and random adjustments to brightness and contrast. These transformations are applied stochastically during training, producing a wide array of images from varying angles and scales, thereby better simulating real-world image diversity.

For instance, in ambulances, cameras installed on the ceiling of the medical compartment are not fixed in position—they may be mounted at any of the four corners. After installation, workers typically adjust the camera angles to capture as much of the compartment’s occupants and equipment as possible. Consequently, the positioning and angles of footage vary across different ambulances.

Ambulance electrical systems are designed with complexity to accommodate diverse power configurations for emergency scenarios. Typically, these systems include a 12V primary vehicle power supply, which serves as the foundational energy source for onboard devices. Additionally, ambulances are equipped with a 24V converted power supply to support high-power equipment, ensuring stable operation during emergencies. A 220V inverter power supply is also integrated to convert direct current (DC) to alternating current (AC), powering critical medical devices and lighting systems. These intertwined multi-power systems inevitably introduce interference to camera footage, leading to variations in image quality.The Augly framework can perform operations such as random rotation and noise addition on ambulance stretcher images to simulate the diversity of images captured by real cameras in complex working environments. These augmentation techniques help researchers better understand and address challenges in real-world model applications, effectively improving the model's performance under various conditions. This makes Augly an

indispensable tool in the fields of machine learning and computer vision, assisting researchers in achieving better results in data processing and model training. The effect of data augmentation is shown in Figure 1.

Figure 1 Data Augmentation Examples

3 EXPERIMENTAL COMPARISON AND ANALYSIS

3.1 Experimental Environment and Parameter Configuration

The hardware and software configurations for all model training and testing in this paper are shown in Table 1, using the self-made dataset described in this chapter. CUDA 12.1 was used for computational acceleration to improve efficiency.

Table 1 Experimental Environment

Parameter	Value
CPU Model	Intel(R) Xeon(R) CPU E5-2680
GPU Model	Nvidia GeForce RTX 2060
Memory	24G
Video Memory	6G
Operating System	Windows 10 Enterprise
Programming Language	Python 3.9
Deep Learning Framework	PyTorch 1.13.1

For the self-made ambulance stretcher dataset, the mmDetection framework was selected as the tool for model performance comparison during multi-model contrast experiments. This framework supports various advanced object detection algorithms, including two-stage and one-stage methods such as TOOD [6], Faster R-CNN [7], and YOLO [8], which are representative in the field of object detection and have good comprehensive detection performance for comparative analysis.

The experimental CPU is Intel(R) Xeon(R) CPU E5-2680, equipped with a GeForce RTX 2060 graphics card with 6G video memory, using CUDA 12.1 for acceleration and Python 3.9 as the programming language. The Stochastic Gradient Descent (SGD) algorithm was used for training with the following parameters: 400 epochs, input image size of 640×640, batch size of 8, initial learning rate of 0.01, and optimizer momentum of 0.937.

3.2 Experimental Result Analysis

In this experiment, each object detection algorithm was rigorously tested to ensure the best-performing version during training. This process is crucial because the effectiveness of the algorithm largely depends on the performance of the selected model. To accurately evaluate the performance of multiple object detection algorithms, the COCO metric system widely recognized in the field of object detection was adopted for evaluation. This evaluation criterion demonstrates good universality and scientific rigor, ensuring the objectivity of the assessment results. Performance is primarily measured using different metrics, including the average precision (AP) across intersection-over-union (IoU) thresholds from 0.5 to 0.95 (AP0.5:0.95) and the average precision at an IoU threshold of 0.5 (mAP0.5). Additionally, the number of parameters in each model must be considered, as it reflects the model's complexity and directly impacts the algorithm's computational efficiency and resource consumption in practical applications—an aspect particularly critical for mobile devices or real-time monitoring systems.Among the compared algorithms, Cascade R-CNN [9] and Mask R-CNN [10] typically achieve higher recognition accuracy, while YOLOv3 [11] and YOLOv8 [12] are more

suitable for real-time applications. Emerging algorithms such as TOOD, GFL [13], and ATSS [14] show good performance in specific scenarios. Each algorithm was trained on the self-made dataset until convergence, and the test results are shown in Table 2. Figure 2 visually displays the accuracy comparison during training.

Table 2 Comparison of Experimental Results for Each Algorithm

Model	AP@0.5	AP@0.5:0.95	Parameters
TOOD	0.747	0.476	32.018M
GFL	0.754	0.477	32.258M
ATSS	0.758	0.491	32.113M
Faster R-CNN	0.774	0.463	41.753M
Cascade R-CNN	0.759	0.499	69.152M
Mask R-CNN	0.754	0.493	43.971M
YOLOv3	0.712	0.464	19.524M
YOLOv8	0.779	0.486	11.2M

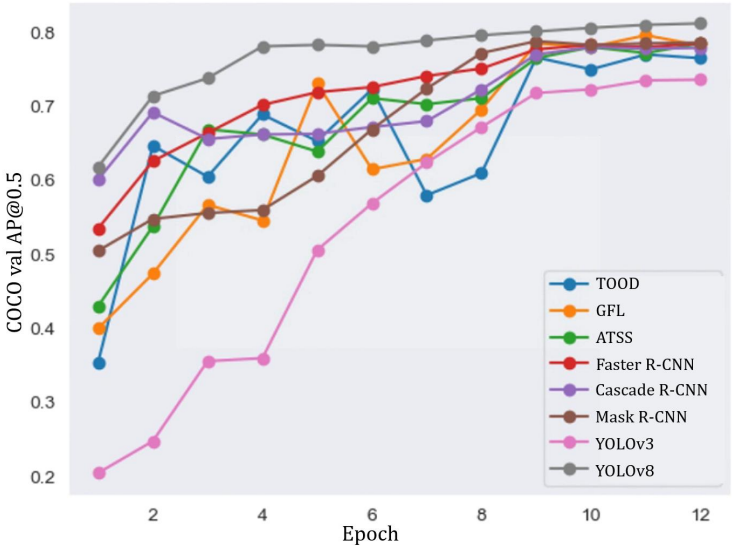

Figure 2 Precision Comparison during Training

Analyzing AP@0.5 (IoU threshold of 0.5), YOLOv8 performs best with an AP@0.5 of 0.779, followed by Faster R-CNN (0.774) and ATSS (0.758). This indicates that Faster R-CNN and YOLOv8 have higher detection accuracy under lower IoU requirements. In terms of parameter quantity, Cascade R-CNN has the most parameters (69.152M), consistent with its complex network structure and multi-stage detection process. In contrast, YOLOv8 has the fewest parameters (11.2M), demonstrating its ability to maintain high performance while significantly reducing model complexity. For AP@0.5:0.95 (averaged over IoU thresholds from 0.5 to 0.95), Cascade R-CNN performs best at 0.499, followed by Mask R-CNN (0.493) and ATSS (0.491), indicating good generalization ability in handling object detection tasks with different IoU thresholds.

Cascade R-CNN is optimal for high-precision scenarios requiring strict IoU thresholds, while YOLOv8 balances high performance with low model complexity, making it more suitable for real-time detection tasks. Therefore, YOLOv8 was selected as the final algorithm for ambulance emergency status judgment.

4 CONCLUSION

This paper addresses the need for ambulance stretcher status detection by constructing a multi-scenario dataset and implementing an automatic emergency status judgment method based on deep learning. Experiments show that YOLOv8 achieves a good balance between accuracy and efficiency, suitable for real-time detection scenarios, providing an efficient technical support for emergency task dispatch. This method significantly reduces manual intervention, improves the utilization of emergency resources, and has important practical significance for optimizing emergency processes. Future research will further enhance the model's robustness in complex lighting and occlusion scenarios and explore lightweight deployment solutions to adapt to more IoT ambulance devices.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Hu Jingchun, Liu Xiaomei, Yang Cheng, et al. Influence of Differences in Pre-hospital Emergency Cardiopulmonary Resuscitation Conditions on the Efficacy of Patients with Cardiac Arrest. Journal of Anhui Health Vocational & Technical College, 2013, 12(01): 27-28.
[2] Zhang Wenwu, Dou Qingli, Liang Jinfeng, et al. Government-led Public First Aid Training: The Practice in Bao'an, Shenzhen. Chinese Journal of Emergency Medicine, 2019(01): 126-128.
[3] S M, T R, N V, et al. Adaptive ambulance monitoring system using IOT. Measurement: Sensors, 2022, 24.
[4] Gao Wenxuan, Yang Xinjie. A Computation Offloading Scheme for Energy Consumption Optimization in Internet of Vehicles. Telecommunications Science, 2023, 39(10): 29-40.
[5] Papakipos Z, Bitton J. Augly: Data augmentations for robustness. arxiv preprint arxiv:2201.06494, 2022.
[6] Feng C, Zhong Y, Gao Y, et al. Tood: Task-aligned one-stage object detection. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE Computer Society, 2021: 3490-3499.
[7] Ren S. Faster r-cnn: Towards real-time object detection with region proposal networks. arxiv preprint arxiv:1506.01497, 2015.
[8] Redmon J. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[9] Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6154-6162.
[10] He K, Gkioxari G, Dollár P, et al. Mask r-cnn. Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
[11] Redmon J. Yolov3: An incremental improvement. arxiv preprint arxiv:1804.02767, 2018.
[12] Terven J, Córdova-Esparza D M, Romero-González J A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Machine Learning and Knowledge Extraction, 2023, 5(4): 1680-1716.
[13] Li X, Wang W, Wu L, et al. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Advances in Neural Information Processing Systems, 2020, 33: 21002-21012.
[14] Zhang S, Chi C, Yao Y, et al. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9759-9768.