IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Ambulance Detection Using Yolov8

Aakansha Kumar[#1],
Dr. Manisha Bharti[#2],
[#]Department of Electronics and Communication
National Institute of Technology Delhi

ABSTRACT- The escalating urban population in India has given rise to a surge in traffic congestion within cities, posing challenges for ambulances to navigate through densely populated streets. This issue is exacerbated by a general lack of public awareness regarding the critical importance of yielding to emergency vehicles. To address this problem, our research focuses on training the YOLOv8 model for effective ambulance identification amidst other vehicles on the road. YOLO, distinguished for its efficacy in object detection, notably excels in swift processing speeds and exceptional accuracy. This project emphasizes the utilization of YOLOv8, which demonstrates an 84.62% precision, a 75.93% recall, and an F1-score of 79.98% for ambulance detection and the application of deep learning methodologies for image segmentation, aiming to enhance emergency vehicle navigation in congested urban environments.

INDEX TERMS- Emergency vehicles, YOLOv8, object detection, deep learning, image segmentation.

I. INTRODUCTION

The burgeoning urban population in India has given rise to a significant upsurge in traffic congestion within cities, posing formidable challenges for ambulances to navigate through densely populated streets. Compounding this issue is a pervasive lack of public awareness regarding the crucial importance of yielding to emergency vehicles. This is particularly crucial in line with the "Golden Hour" theory, endorsed by the World Health Organization, which asserts that transporting road accident victims or heart disease patients to healthcare facilities within an hour significantly enhances their chances of recovery by 70 to 80 percent.

Despite the global adherence to the "Golden Hour" principle, India faces alarming statistics, with approximately 168,491 people reported killed in road accidents in 2023, and a disheartening 30% of these fatalities attributed to ambulance services experiencing delays.[4] Additional government data indicates that more than 50% of heart attack cases arrive at the hospital after significant delays, with traffic congestion being a major contributor.

In response to this critical issue, this research project undertakes the task of training a model to identify ambulances amid other vehicles on the road, employing the YOLOv8 model for object recognition. Yolov8,

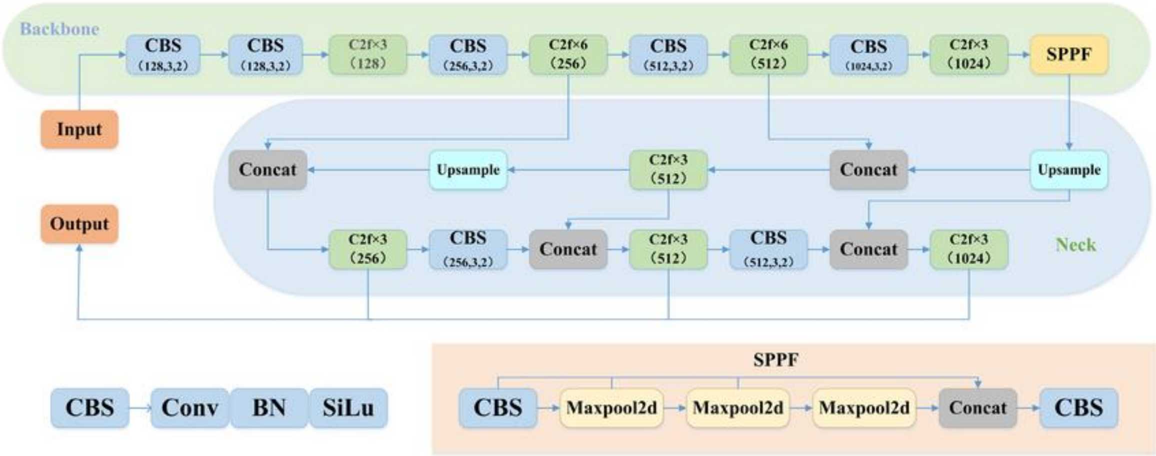

Figure 1. YOLOv8 network structure diagram

renowned for its efficacy in object detection, stands out as a preferred model architecture and algorithm due to its exceptional accuracy and rapid processing speeds. This study focuses on harnessing YOLOv8 for object detection and leveraging deep learning methodologies for image segmentation.[2]

The subsequent sections of this paper are organized as follows: Section II provides a review of the literature, while Section III outlines the theory and methodology proposed. Section IV comprises a discussion of the results and section V offers the study's concluding remarks.

II. LITERATURE REVIEW

Object recognition and detection using deep learning, particularly with models like YOLO (You Only Look Once), have shown remarkable performance in various real-world applications.[3] However, challenges such as image noise, blurring, and rotation jitter significantly impact the accuracy of detection systems, particularly in scenarios like traffic detection.

Studies have addressed these challenges by developing image degradation models that simulate real-world shooting conditions.[9] By combining traditional image processing methods with deep learning methodologies utilizing YOLO networks, researchers have been able to evaluate the effects of different degradation models on object detection accuracy. These studies highlight the importance of robust models capable of handling image degradation for tasks like traffic detection in real scenes.

In the context of smart city applications, like parking capacity utilization, the use of deep neural networks requires fine-tuning to adapt to specific scenarios.[1] While existing models for object detection like YOLO offer good precision and real-time performance, customization with proprietary data can lead to further improvements in accuracy and efficiency.

Recent progress has resulted in the creation of adapted deep object detection models utilizing architectures such as YOLOv5 and YOLOv8. These models aim to detect objects of different sizes, including large, small, and tiny objects like vehicles.[13] An approach entails incorporating a multi-scale mechanism to autonomously learn discriminative feature representations at various scales. This alteration not only decreases the number of trainable parameters in comparison to the original YOLO-v5 architecture but also notably enhances precision. For instance, experiments have shown marginal parameter reduction and enhanced detection speed, with notable improvements in tiny vehicle detection performance.[5]

In light of these developments, there is a need for further research to enhance object recognition systems based on YOLO v8. This includes exploring novel techniques to optimize model architectures for specific smart city applications and evaluating performance on diverse datasets.[5] By building upon existing literature and leveraging advancements in deep learning, future studies can contribute to the continued evolution of object recognition technologies for smart city environments.

III. THEORY AND METHODOLOGY

The YOLO network maintains a significant advantage in speed over other networks in the one-stage category. Furthermore, it has shown comparable performance to state-of-the-art methods while preserving accuracy, leveraging the global context of the input image for predictions.[3] Hence, our proposed model is based on the YOLOv8 architecture as its foundation.

An inherent strength of YOLOv8 lies in its ability to accurately identify and locate objects within an image, addressing a wide range of object sizes and categories. Unlike its predecessor YOLOv5, YOLOv8 is not constrained to specific object categories or contexts. This makes it adaptable to diverse real-world applications such as security, surveillance, autonomous vehicles, and smart city implementations.[2] Moreover, YOLOv8 boasts improvements in precision and recall, ensuring a more reliable and accurate detection process. The enhanced multi-scale mechanism of the model contributes to better feature discrimination at various scales, ultimately enhancing overall performance.

The utilization of anchor-free detection and novel convolutional layers enhances prediction accuracy in YOLOv8. Its notable trait lies in its extensibility, designed as a framework compatible with all preceding YOLO iterations.[6] This allows for a seamless transition between models and facilitates performance evaluation across versions. Consequently, YOLOv8 stands out as the optimal choice for individuals seeking to leverage the latest YOLO advancements while ensuring the continued functionality of their existing YOLO models.[7] Moreover, YOLOv8 strikes a better balance between training time and precision.

Model Size	YOLOv5 mAP	YOLOv8 mAP	Difference
Nano	28	37.3	+33.21%
Small	37.4	44.9	+20.05%
Medium	45.4	50.2	+20.57%
Large	49	52.9	+7.96%
Extra Large	50.7	53.9	+6.31%

Table 1. Object detection performance comparison (YOLOv8 vs YOLOv5) where the image size is 640.

Average Precision (AP) is a commonly utilized metric for assessing the accuracy of object detectors. It measures the area under the precision-recall curve, providing a unified metric that encapsulates both the precision and recall performance of the model.

By computing the average AP values across various object classes, Mean Average Precision (mAP) expands upon the notion of AP.

There are five models in each category of YOLOv8 models for detection, segmentation, and classification. Among them, YOLOv8 Nano stands out as the fastest and smallest, whereas YOLOv8 Extra Large (YOLOv8x) is recognized for its superior accuracy despite being the slowest.

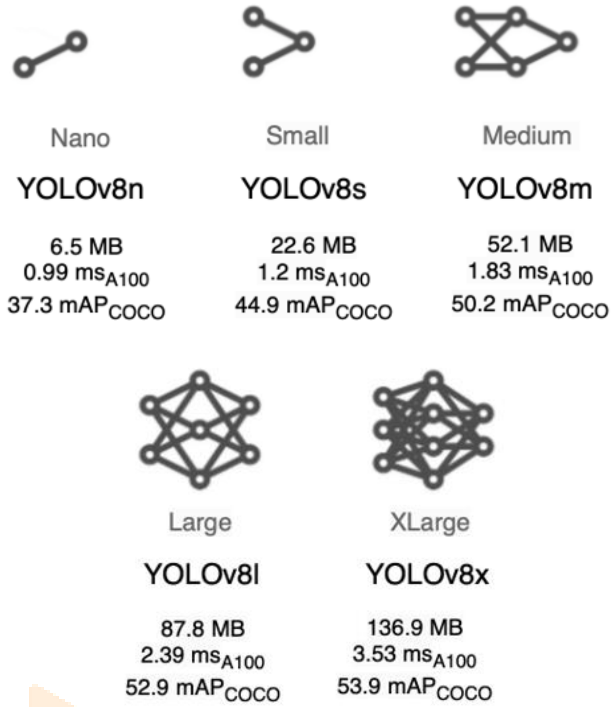

Figure 2. Models Available in YOLOv8

The model creation process for recognizing ambulances in traffic involves various steps. The first is creating the dataset. Before developing an object recognition model, the initial step involves selecting a dataset, specifically focusing on cars and ambulances for this project. To train a computer vision model effectively, it is imperative to have labeled data for the training process. The accuracy of these labels, or annotations, significantly influences the model's performance, with higher precision achievable through meticulous labeling.

To facilitate this process, we configured the environment. This setup ensures that the necessary tools and dependencies are in place, laying the foundation for subsequent stages in the development of the object recognition model. Creating a dataset involves further two steps, downloading images and annotating the data.

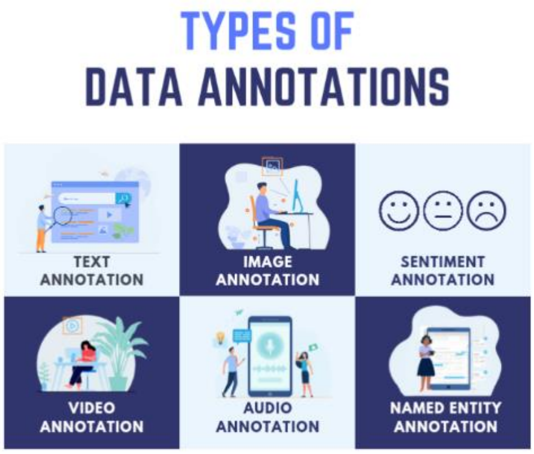

Figure 3: Types of data annotations.

We proceed to acquire the images for our dataset. This step involves downloading the necessary visual data to construct a comprehensive dataset for training the object recognition model.

Annotated images are text files derived from regular images, wherein each file corresponds to an image containing a single object of interest. The annotation format consists of a single row of digits.

0 0.212322 0.232432 0.093231 0.781622
A b c d e

Figure 4. Structure of the annotation format. Here, "a" represents the class label, while "b," "c," "d," and "e" denote the bounding box coordinates.

Annotating images allows the precise identification of object classes and the definition of corresponding bounding box dimensions. The annotation process is crucial for providing labeled training data, enabling the model to learn and recognize the specified objects accurately.

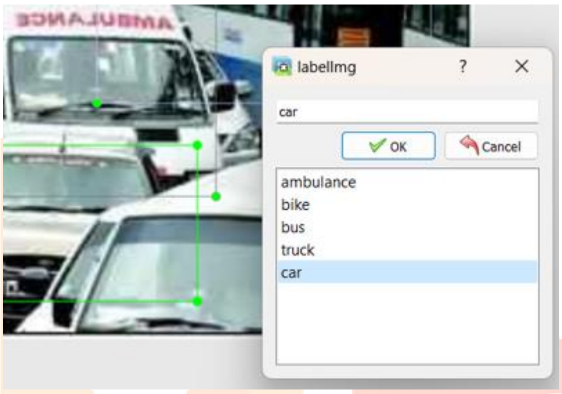

Figure 5: Labelling the different classes in the image.

After creating the dataset, the next crucial phase is preparing YOLOv8 for deployment. This is achieved by executing a set of commands that initialize and configure the YOLOv8 model. These commands play a pivotal role in defining the model architecture, loading the dataset, and configuring various parameters necessary for training and inference. The proper setup ensures that the YOLOv8 model is ready to be trained on the provided dataset, facilitating effective object detection based on the specified classes, such as ambulances in a traffic scenario.

Following the setup of YOLOv8, the next step involves training the model to recognize objects, particularly ambulances in this case. During training, the model refines its parameters based on the labeled dataset, learning to accurately identify and locate ambulances within images. The training process is iterative and involves adjusting the model's weights and biases to optimize its performance. Successful completion of this step is essential for the model to achieve high accuracy and effectiveness in subsequent tasks of object detection.

After the training of the model, the next phase is prediction, where the trained YOLOv8 model is applied to new images to identify and localize objects. This step is crucial for assessing the model's performance on real-world data and verifying its ability to generalize from the training dataset to new, unseen examples. The prediction results provide insights into the model's effectiveness in detecting ambulances within diverse image contexts.

IV. RESULTS AND DISCUSSION

Initially, the presentation of labels serves as an outcome that elucidates the object classes and defines the corresponding bounding box dimensions. This representation is a direct consequence of the annotation process, a pivotal step in furnishing labeled training data. This annotated data is instrumental in facilitating the model's learning process, ensuring its ability to accurately recognize and categorize specified objects.

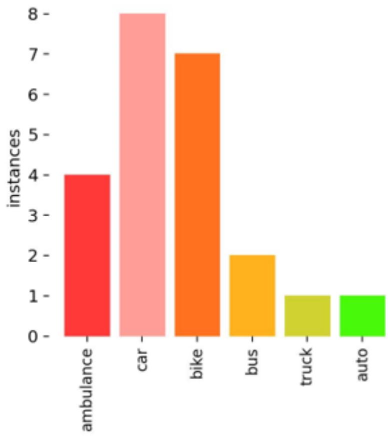

Figure 6. Labels of the object classes used in the model for training.

Figure 7. Batch-wise training

Batch-wise training in YOLOv8 updates the model's parameters using a subset or batch of training samples rather than the entire dataset in a single iteration. This approach helps improve the efficiency of the training process by reducing computational demands and memory requirements.

During each training iteration, a batch of images and their corresponding annotations are processed through the neural network, and the model's parameters are updated based on the computed loss. The use of batches allows for parallel processing of multiple samples, enabling faster convergence and effective utilization of available computational resources.

Finally, after the completion of the model training and evaluation process, predictions of the model regarding the presence or detection of ambulances within the given input dataset have been obtained. As seen in Figure 8, these results are important for assessing the performance and efficacy of the developed model in the context of ambulance detection.

V. CONCLUSION

In this paper, a model to detect ambulances using YOLOv8 is proposed, following the meticulous phases of dataset preparation, model configuration, training, and subsequent prediction. The results, presented in the form of images, serve as a visual testament to the YOLOv8 model's adeptness in successfully detecting ambulances, with a precision of 84.62%, a recall of 75.93%, and an F1-score of 79.98%. The images vividly showcase the model's precision in identifying and highlighting specified objects within a diverse array of image scenarios. This project also emphasizes the practical importance of the developed model in real-world situations, particularly in addressing challenges related to ambulance detection in diverse and dynamic environments, thereby fulfilling the objectives.

Figure 8. Ambulance is detected among other vehicles on the road according to the various object classes provided to the model

REFERENCES-

1. D. Padilla Carrasco, H. A. Rashwan, M. Á. García and D. Puig, "T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks," in IEEE Access, vol. 11, pp. 22430-22440, 2023, doi: 10.1109/ACCESS.2021.3137638.

2. Lou, Haitong & Duan, Xuehu & Guo, Junmei & Liu, Haiying & Gu, Jason & Bi, Lingyun & Chen, Haonan. (2023). DC-YOLOv8: Small Size Object Detection Algorithm Based on Camera Sensor.10.20944/preprints202304.0124.v1

3. C. Liu, Y. Tao, J. Liang, K. Li and Y. Chen, "Object Detection Based on YOLO Network," 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2018, pp. 799-803, doi: 10.1109/ITOEC.2018.8740604.

4. Tariq, Saba & Tariq, Sundus & Malik, Akram. (2021). Give way to the ambulance; What we all need to do to facilitate ambulance in traffic?. Journal of University Medical & Dental College. 12.10.37723/jumdc.v12i4.696.

5. A. Afdhal, K. Saddami, S. Sugiarto, Z. Fuadi and N. Nasaruddin, "Real-Time Object Detection Performance of YOLOv8 Models for Self-Driving Cars in a Mixed Traffic Environment," 2023 2nd International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE), Banda Aceh, Indonesia, 2023, pp. 260-265, doi: 10.1109/COSITE60233. 2023.10249521.

6. M. dos Santos de Arruda, L. P. Osco, P. R. Acosta, D. N. Gonçalves, J. M. Junior, A. P. M. Ramos, E. T. Matsubara, Z. Luo, J. Li, J. de Andrade Silva, and W. N.

Gonçalves, "Counting and locating high-density objects using convolutional neural network," 2021, arXiv:2102.04366.

7. J. -H. Kim, N. Kim and C. S. Won, "High-Speed Drone Detection Based On Yolo-V8," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-2, doi: 10.1109/ICASSP49357. 2023.10095516.

8. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

9. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," 2015, arXiv:1506.01497

10. J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, arXiv:1804.02767.

11. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, arXiv:2004. 10934.

12. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," 2020, arXiv:2011.08036.

13. G. Jocher, A. Stoken, J. Borovec, L. Changyu, A. Hogan, A. Chaurasia, L. Diaconu, Doug, Durgesh, F. Ingham, Frederik, Guilhen, A. Colmagro, H. Ye, Jacobsolawetz, J. Poznanski, J. Fang, J. Kim, K. Doan, and L. Yu, "Ultralytics/yolov5: V4.0—Nn.SiLU() activations, weights & biases logging, PyTorch hub integration," Ultralytics, Los Angeles, CA, USA, Jan. 2021, doi: 10.5281/zenodo.4418161.

14. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2016, arXiv:1612.03144.

15. C. Kumar B., R. Punitha, and Mohana, "YOLOv3 and YOLOv4: Multiple object detection for surveillance applications," in Proc. 3rd Int. Conf. Smart Syst. Inventive Technol. (ICSSIT), Aug. 2020, pp. 1316–1321.