
ISSN: 2321-2152
IJMECE
International Journal of modern
electronics and communication engineering
E-Mail
editor.ijmece@gmail.com
editor@ijmece.com
www.ijmece.com

ISSN 2321-2152
www.ijmece .com
Vol 12, Issue.1 March 2024

OBJECT DETECTION USING DEEP LEARNING YOLOV8

Saiprapul Boyapati[1], SivaKrishna Gogireddy[2], Chandu Kariggi [3], Dr P Dileep [4]

[1,2,3]B.Tech Student, Department of CSE (Internet of Things), Malla Reddy College of Engineering and Technology, Hyderabad,India.

[4] Professor, Department of CSE (Internet of Things), Malla Reddy College of Engineering and Technology, Hyderabad, India.

ABSTRACT— Object detection is a challenging problem in the field of computer vision, and over the past decade, significant progress has been made thanks to the rapid evolution of deep learning. Researchers The evaluation of an object detector typically hinges on its detection accuracy and inference time. Generally, two-stage detectors tend to outperform single-stage detectors in terms of detection accuracy, while single-stage detectors offer faster inference times. However, recent advances, such as the YOLO (You Only Look Once) model have made substantial contributions to improving the performance of object detection and related tasks like object classification, localization, and segmentation using deep learning models. Object detectors can be broadly categorized into two groups: two-stage and single-stage detectors. Two-stage detectors primarily employ complex architectures that focus on selective region proposals, whereas single-stage detectors use simpler architectures to consider all spatial region proposals for potential object detection in a single shot. and its architectural successors, have significantly improved detection accuracy and sometimes outperform two-stage detectors. YOLO models have found wide-ranging applications, primarily due to their faster inference times, often prioritizing speed over detection accuracy. In our proposed system, we have implemented the state-of-the-art YOLOv8 model for real-time weapon detection. YOLOv8 offers improvements over YOLOv5 in terms of both speed and accuracy. To ensure efficient performance, we have quantized the weights of the YOLOv8 model. In our experiments, we assessed the performance of YOLOv8 and YOLOv5 models for weapon detection. We achieved a mean Average Precision (mAP) value of 90.1% with YOLOv8, surpassing the mAP value of 89.1% obtained with YOLOv5. Furthermore, by applying weight quantization to the YOLOv8 model, we managed to reduce the inference time by 15% compared to the original YOLOv8 configuration. This combination of improved accuracy and faster inference makes YOLOv8 an excellent choice for real-time weapon detection applications.

Keywords— Object detection, Single-stage detectors, Two-stage detectors, Deep learning, YOLO, mean

Average Precision (mAP).

I.INTRODUCTION

Object detection poses a formidable challenge in the realm of computer vision, involving the identification and precise localization of objects within images or videos. With the increasing availability of high-resolution cameras and the explosion of video data, object detection has become an indispensable tool with a multitude of applications. However, object detection remains a demanding task due to the complexities and variations found in real-world environments, including occlusions, clutter, and the need to simultaneously detect a large number of objects.

In recent years, the advent of deep learning techniques has ushered in a revolution in object detection, leading to remarkable enhancements in both accuracy and speed. Among the notable families of object detection algorithms, YOLO (You Only Look Once) stands out, utilizing deep Convolutional Neural Networks (CNNs) to achieve real-time object detection. The YOLO algorithm operates by partitioning the input image into a grid, subsequently identifying bounding boxes and confidence scores for each grid cell. What sets YOLO apart is its innovative loss function that seamlessly combines localization and classification errors, making it adept at handling overlapping objects and ultimately boosting detection accuracy.

Object detection is an ever-evolving field, characterized by ongoing research efforts aimed at augmenting the precision and efficiency of detection algorithms. The future of object detection may encompass the utilization of advanced deep learning architectures, the integration of additional sensor modalities, and the fusion of object detection with other computer vision tasks such as semantic segmentation and instance segmentation.

Recently, a novel iteration of the YOLO algorithm, known as YOLOv8, has been introduced, pushing the boundaries of object detection accuracy and speed even further. YOLOv8 introduces an inventive architecture grounded in a hybrid backbone network, amalgamating features from different abstraction levels. This innovative approach enables YOLOv8 to achieve precise object detection with remarkable efficiency. Furthermore, YOLOv8 incorporates numerous optimizations, including a redesigned anchor box scheme, a feature pyramid network, and enhanced training techniques, all contributing to its exceptional performance..

The performance of the proposed approach is evaluated using the testing set, and metrics such as accuracy, classification report, and confusion matrix are computed. Furthermore, examples of object detection are provided, demonstrating the

ISSN 2321-2152
www.ijmece .com
Vol 12, Issue.1 March 2024

model's capability to classify random data points, custom data points, and user input.

The outcomes of this research contribute to the advancement of object detection system, offering improved accuracy and performance. By leveraging the capabilities of deep learning networks.

II.LITERATURE REVIEW

A fast and simple approach to detecting real time images was introduced in this paper as You Only Look Once. The model was built to detect images accurately, fast and to differentiate between art and real images. In this literature review, we discussed studies conducted by Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi proposed "You Only Look Once: Unified, Real-Time Object Detection "A Deep learning based object detection using YOLO.

In conclusion, In comparison with Object detection techniques that came before YOLO, like R-CNN, YOLO introduced a single unified architecture for regression go image into bounding boxes and finding class probabilities for each box. This meant that YOLO performed much faster and also provided more accuracy. It could also predict art work correctly.

You Only Look Once: Unified, Real-Time Object Detection, by Joseph Redmon. Their prior work is on detecting objects using a regression algorithm. To get high accuracy and good predictions they have proposed YOLO algorithm in this paper.

Understanding of Object Detection Based on CNN Family and YOLO, by Juan Du. In this paper, they generally explained about the object detection families like CNN, R-CNN and compared their efficiency and introduced YOLO algorithm to increase the efficiency. Learning to Localize Objects with Structured Output Regression, by Matthew B. Blaschko. This paper is about Object Localization. In this, they used the Bounding box method for localization of the objects to overcome the drawbacks of the sliding window method.

III.METHODOLOGY

1. Data preprocessing:

- Gather a labeled dataset containing images or videos with annotated object bounding boxes and class labels.
- Preprocess the data, including resizing images, normalizing pixel values, and augmenting the dataset if necessary to increase diversity.

2. Data Splitting::

- Divide the dataset into three subsets: training, validation, and testing sets. Common splits are 70-80% for training, 10-15% for validation, and 10-15% for testing.

3. Model Selection:

- Choose YOLOv8 as the deep learning model for object detection. You can access pre-trained weights for YOLOv8 or train the model from scratch, depending on your dataset size and task.

4. Model Configuration:

- Configure the YOLOv8 model based on your specific requirements. Adjust parameters such as the number of anchor boxes, network architecture, and input size.

5. Training:

- Train the YOLOv8 model on the training dataset using the selected configuration.
- Utilize a suitable loss function, such as a combination of localization loss and classification loss, which is typically used in YOLO-based models.
- Monitor training progress by evaluating metrics on the validation dataset, such as mean Average Precision (mAP).

6. Model Configuration:

- Fine-tune hyperparameters, including learning rate, batch size, and training epochs, to optimize model performance.
- Implement techniques like learning rate scheduling to aid convergence.

7. Model Evaluation:

- Asses the trained YOLOv8 model on the testing dataset to evaluate its performance on unseen data.
- Custom Data Point Fraud Detection: Prompt the user to input the index of a data point they want to check, extract the corresponding data point from the dataset, make a predicition using the model, and compare it with actual label.

8. Inference:

- Deploy the trained YOLOv8 model for realtime object detection on new images or video streams.
- Measure the inference time to ensure that it meets real-time requirements.

9. Visualization:

- Visualize the object detection results by drawing bounding boxes around detected objects and labeling them with their corresponding class labels.

10. Deployment:

- Deploy the trained YOLOv8 model in your target application, whether it's for surveillance, autonomous vehicles, or any other object detection task.

11. Monitoring and Maintenance:

-

Continuously monitor the model's performance in the production environment and retrain it periodically with new data to maintain accuracy.

12. Documentation:

- Document the entire process, including data collection, model configuration, training parameters, and evaluation results, to facilitate reproducibility and future reference.

ISSN 2321-2152
www.ijmece .com
Vol 12, Issue.1 March 2024

IV.IMPLEMENTATION

1. Load the dataset that you want to train your model on. Make sure that the dataset contains images of objects that you want to detect.
2. Annotate the dataset with bounding boxes around the objects you want to detect. You can use tools like labeling, reactlabel for the step.
3. Preprocess the annotated dataset by resizing the images, normalizing the pixel values, and splitting the dataset into training and validation sets.
4. Train the YOLOv8 model on the preprocesses dataset using deep learning framework like pytouch. You can use pre-trained weights to speed up the training process.
5. Evaluate the trained model on the validation set to measure its accuracy and performance. You must use metrics like mean Average Precision (mAP) to evaluate model.
6. Optimize the trained model by fine-tuning its hyperparameters, adjusting the learning rate, or changing the architecture.
7. Create a new floder and place the data in it and your dataset is ready.
8. Deploy the model on a production environment and use it for object detection tasks.

V.RESULTS

During the initial training phase of the model, the following results were observed.

Python CLI

```python
from ultralytics import YOLO

# Load a model
model = YOLO('yolov8n.yaml')  # build a new model from YAML
model = YOLO('yolov8n.pt')  # load a pretrained model (recommended for training)
model = YOLO('yolov8n.yaml').load('yolov8n.pt')  # build from YAML and transfer weights

# Train the model
results = model.train(data='coco128.yaml', epochs=100, imgsz=640)
```

Fig 1: Trained Data

Python CLI

```python
from ultralytics import YOLO

# Load a model
model = YOLO('yolov8n.pt')  # load an official model
model = YOLO('path/to/best.pt')  # load a custom model

# Validate the model
metrics = model.val()  # no arguments needed, dataset and settings remembered
metrics.box.map    # map50-95
metrics.box.map50  # map50
metrics.box.map75  # map75
metrics.box.maps   # a list contains map50-95 of each category
```

Fig 2: Load a Model

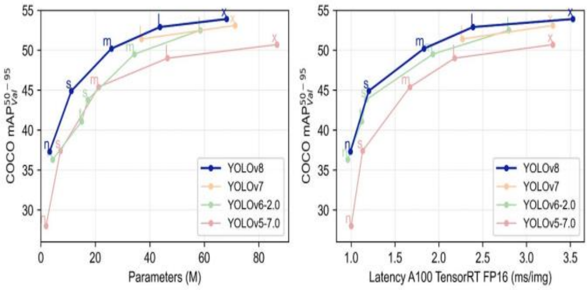

Fig 3: Comparison with previous YOLO Version

ISSN 2321-2152
www.ijmece .com
Vol 12, Issue.1 March 2024

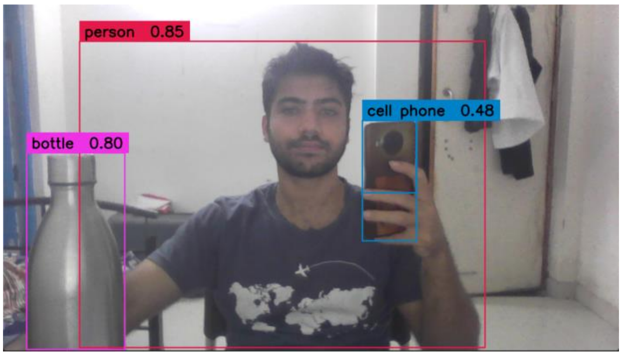

Fig 4: Applications For Object Detection

VI .CONCLUSION

In conclusion, object detection using deep learning with YOLOv8 represents a significant advancement in the field of computer vision.

YOLOv8, as a member of the YOLO (You Only Look Once) family of algorithms, has demonstrated remarkable capabilities in accurately and efficiently detecting objects within images and videos.

The adoption of deep learning strategies, such as YOLOv8, has paved the way for substantial improvements in both the speed and accuracy of object detection. YOLOv8's innovative architecture and loss functions have enabled it to handle complex real-world scenarios, including occlusions, clutter, and the simultaneous detection of numerous objects. This makes it a valuable tool for a wide range of applications where real-time object detection is crucial.

VII. FUTURE ENHANCEMENTS

As we continue to witness the rapid evolution of deep learning and object detection, YOLOv8 represents a significant milestone. It is poised to play a pivotal role in various real-world applications, including security, autonomous vehicles, surveillance, and more. With further research and refinement, YOLOv8 and similar deep learning-based approaches hold the promise of even greater advancements in the field of object detection, addressing the complex challenges of today's dynamic and ever-changing environments. Additionally, the quantization of YOLOv8's weights has demonstrated its ability to reduce inference times while maintaining high accuracy.

VIII.REFERENCES

1. M. Y. Zou, J. J. Yu, Y. Lv, B.Lu, W. Z. Chi and L. n. Sun. A Novel Day-to-Night Obstacle Detection Method for Excavators based on Image Enhancement and. IEEE Sensors Journal, 2023, pp. 1–11.

2. C. Li et al., YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. 2022, [Online]. Available: http://arxiv.org/abs/2209.02976.

3. A. D. Abadi, Y. Gu, I. Goncharenko, and S. Kamijo. Detection of Cyclist's Crossing Intention based on Posture Estimation for Autonomous Driving. 2023, IEEE Sensors Journal, pp. 1–1.

4. J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. 2018, arXiv preprint arXiv:1804.02767.

5. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2015, 28, Advances in neural information processing systems.

6. H. Liu, F. Sun, J. Gu, and L. Deng. SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode. 2022, Sensors, vol. 22, no. 15, pp. 1–14.

7. Cao Z, Liao T, Song W, Chen Z, Li C (2021) Detecting the shuttlecock for a badminton robot: a YOLO based approach. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113833

8. Albelwi S, Mahmood A (2017) A framework for designing the architectures of deep convolutional neural networks. Entropy 19(6):242.

9. Bhattacharya S, Maddikunta PKR, Pham QV, Gadekallu TR, Chowdhary CL, Alazab M, Piran MJ (2021) Deep learning and medical image processing for coronavirus (COVID-19) pandemic: a survey. Sustain Cities Soc 65:102589. https://doi.org/10.1016/j.scs.2020.102589

10. Quah, Agarwal S, Terrail JO, Jurie F (2018) Recent advances in object detection in the age of deep convolutional neural networks. arXiv preprint arXiv:1809.03193. https://doi.org/10.48550/arXiv.1809.03193