International Scientific Journal of Engineering and Management (ISJEM0) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM05242
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Custom Object Detection Using YOLOV11 Model

Ruchika Mittal[1], Devender Singh Topwal[2]

Department of Electrical & Electronics Engineering
Bhagwan Parshuram Institute of Technology GGSIPU, New Delhi-110036

ABSTRACT

The evolution of deep learning has revolutionized visual recognition, yet traditional detection models often struggle with the trade-off between accuracy and inference speed in custom scenarios. This paper presents a novel implementation of the YOLOv11 model tailored for custom object detection tasks. Utilizing a transfer learning methodology, the framework effectively adapts high-level feature representations from pre-trained weights to a custom dataset, ensuring adaptability across diverse visual variations. The investigation highlights the architectural advancements of YOLOv11, including its optimized backbone and unified detection pipeline. Quantitative analysis using mean Average Precision (mAP), Precision, and Recall metrics demonstrates the model's efficacy in minimizing false negatives while maintaining real-time performance. This study confirms YOLOv11's viability as a lightweight, high-performance architecture suitable for deployment in resource-constrained computer vision systems.

Keywords: YOLOv11, Object Detection, Deep Learning, Computer Vision, Real-Time Detection, Custom Dataset, Segmentation, mAP.

1. INTRODUCTION

Computer vision continues to evolve rapidly, enabling machines to interpret and analyze visual information with increasing precision. Among its various tasks, object detection remains one of the most essential and challenging, as it requires both accurate classification of objects and precise localization within images or video streams. Although humans can easily recognize objects in real time, achieving comparable performance through algorithms was difficult prior to the advancement of deep learning–based architectures. Over the years, progress in neural network design and the availability of large-scale image datasets have significantly improved the accuracy, speed, and reliability of visual recognition systems.

A major milestone in real-time detection was achieved with the introduction of the You Only Look Once (YOLO) family of models. YOLO revolutionized computer vision by treating detection as a single-stage regression problem, allowing the entire image to be processed in one forward pass. This shift drastically boosted inference speed and made real-time detection practical for real-world applications such as surveillance, traffic monitoring, automation, and intelligent robotics.

The newest member of this family, YOLOv11, represents a substantial leap forward in both performance and design. Announced at the YOLO Vision 2024 conference, YOLOv11 integrates advanced feature extraction strategies, optimized architectural components, and refined training methodologies to push the boundaries of efficiency and accuracy. The model achieves a high degree of precision while maintaining a lightweight structure, enabling deployment across a wide range of devices—from high-performance systems to resource-constrained platforms. Its improvements in speed, generalization capability, and multi-task performance position YOLOv11 as one of the most powerful and versatile real-time detection models currently available.

This paper focuses exclusively on exploring the capabilities of YOLOv11 for custom object detection. With millions of images being generated daily, modern systems must handle diverse and dynamic visual environments. YOLOv11 addresses this need by offering enhanced robustness to variations in object size, shape, orientation, and lighting conditions. Furthermore, its improved training pipeline and feature integration empower it to deliver both high accuracy and low latency, even with limited training data.

The subsequent sections of this work present a detailed analysis of YOLOv11's architecture, its advancements over earlier iterations, and its performance across various computer vision tasks. Additionally, the study evaluates the model's suitability for custom object detection scenarios, where specialized datasets and domain-specific knowledge are required. By examining YOLOv11's strengths, limitations, and real-world applicability, this paper aims to highlight its growing importance within the broader landscape of modern computer vision systems.

1.1 Background

Object detection has become a core component of modern computer vision, enabling machines to interpret and interact with visual environments effectively. While humans identify objects effortlessly, computers require sophisticated algorithms to understand and analyze the visual world. The rapid evolution of deep learning has drastically improved this capability, allowing algorithms to learn complex features from large datasets and perform visual tasks with high precision.

Table 1: YOLO: Evolution of models

Release	Year	Tasks	Contributions	Framework
YOLO [5]	2015	Object Detection, Basic Classification	Single-stage object detector	Darknet
YOLOv2 [7]	2016	Object Detection, Improved Classification	Multi-scale training, dimension clustering	Darknet
YOLOv3 [8]	2018	Object Detection, Multi-scale Detection	SPP block, Darknet-53 backbone	Darknet
YOLOv4 [9]	2020	Object Detection, Basic Object Tracking	Mish activation, CSPDarknet-53 backbone	Darknet
YOLOv5 [10]	2020	Object Detection, Basic Instance Segmentation (via custom modifications)	Anchor-free detection, SWISH activation, PANet	PyTorch
YOLOv6 [11]	2022	Object Detection, Instance Segmentation	Self-attention, anchor-free OD	PyTorch
YOLOv7 [12]	2022	Object Detection, Object Tracking, Instance Segmentation	Transformers, E-ELAN reparameterisation	PyTorch
YOLOv8 [13]	2023	Object Detection, Instance Segmentation, Panoptic Segmentation, Keypoint Estimation	GANs, anchor-free detection	PyTorch
YOLOv9 [14]	2024	Object Detection, Instance Segmentation	PGI and GELAN	PyTorch
YOLOv10 [15]	2024	Object Detection	Consistent dual assignments for NMS-free training	PyTorch

Figure 1:Evolution of YOLO Models

The development of the YOLO (You Only Look Once) series marked a transformative moment in real-time object detection. By reframing detection as a single-step prediction problem, YOLO models eliminated the need for multi- stage pipelines and made real-time performance achievable. The latest iteration, **YOLOv11**, builds upon this legacy with major architectural and training advancements. YOLOv11 incorporates optimized feature extraction, enhanced computational efficiency, and improved accuracy across diverse detection scenarios. Its ability to deliver high-speed inference without compromising precision makes it suitable for applications such as surveillance, traffic control, autonomous systems, and intelligent monitoring.

As visual data continues to grow exponentially through social media, cloud platforms, and sensor networks, the demand for fast, robust, and adaptable detection frameworks has intensified. YOLOv11's versatility and enhanced performance address these modern challenges, offering an efficient and scalable solution for real-world computer vision applications.

1.2 Problem Statement

Despite significant advancements in object detection, several challenges remain when deploying detection algorithms in practical scenarios. Traditional feature extraction methods often struggle with complex environments, as they primarily capture low-level visual cues such as edges and textures. These methods lack the capacity to generalize across diverse object shapes, lighting conditions, and backgrounds, leading to reduced accuracy in real-world settings.

Although object detection has advanced significantly, many models still struggle in real-world scenarios due to complex environments, limited training data, and the trade-off between accuracy and speed. Traditional approaches often fail to generalize well and require extensive computational resources. Custom object detection becomes even more challenging when datasets are small or highly specific. YOLOv11 offers improved speed and precision, but its effectiveness for custom detection tasks must be evaluated to determine whether it can reliably address these limitations.

2. LITERATURE REVIEW

2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) form the foundational architecture for modern computer vision systems, enabling models to learn visual patterns directly from raw data without the need for manual feature engineering. CNNs are highly effective in identifying spatial structures within images, such as edges, shapes, textures, and complex object configurations. Their hierarchical learning capability allows them to perform exceptionally well not only in image- related tasks but also in domains such as audio processing, signal interpretation, and medical imaging.

A typical CNN architecture consists of convolutional layers, pooling layers, and fully connected layers. The convolutional layer extracts local features from input images using learnable filters, while activation functions like ReLU introduce non-linearity to accelerate training and improve feature discrimination. Pooling layers then reduce the dimensionality of feature maps and enhance invariance to translation and distortion. Fully connected layers interpret the learned features to generate the final classification or

detection outputs.

CNNs operate with shared weights and biases across neurons in the same feature map, making them robust to positional changes in objects. This weight-sharing mechanism allows CNNs to generalize across varying image regions. Classification layers such as Softmax convert the learned features into probabilistic outputs for multiple classes. Designing and training CNNs can be facilitated using deep learning frameworks that support GPU acceleration, enabling faster processing of large datasets and complex architectures.

Transfer learning further enhances CNN utility by allowing pre-trained networks to be fine-tuned on specific datasets. This approach reduces training time and computational cost while leveraging previously learned features. Such capabilities make CNNs an integral part of object detection frameworks and a necessary precursor to advanced real- time models.

2.2 Advancements in Real-Time Object Detection: The YOLO Series

The YOLO series revolutionized object detection by introducing a unified framework capable of predicting bounding boxes and class probabilities in a single pass. Each new version refined model structure, feature extraction, and training strategies to balance accuracy and speed. Improvements in network depth, width, loss functions, and augmentation methods enabled YOLO models to handle diverse detection challenges with increased robustness.

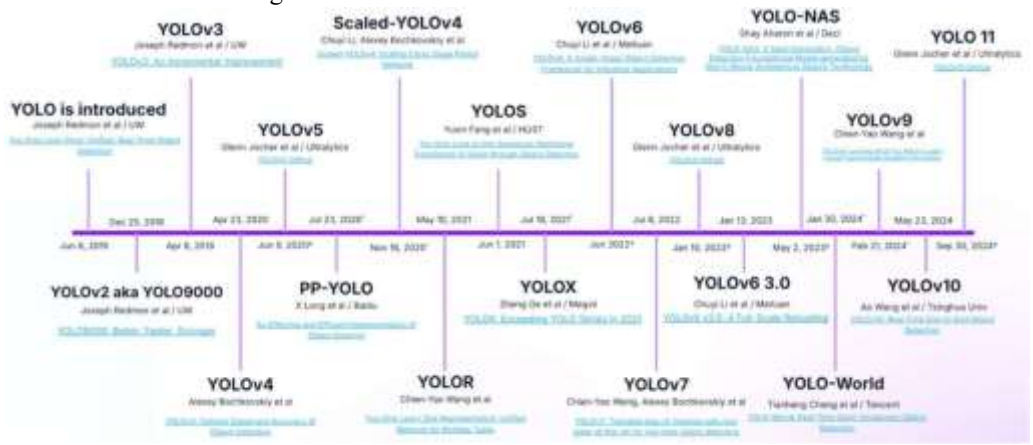

Figure 2:The YOLO Series

YOLOv11 represents the next major step in this lineage. It introduces enhanced architectural components, optimized training pipelines, and superior computational efficiency compared to earlier models. YOLOv11 incorporates refined feature extraction mechanisms that improve the detection of small, overlapping, or visually complex objects while maintaining lightweight performance. These advancements make YOLOv11 suitable for real-time applications like surveillance, autonomous navigation, and smart city systems.

2.3 Transfer Learning for Custom Object Detection

Transfer learning plays a crucial role in adapting modern object detection models to domain-specific tasks. Custom datasets often contain limited labeled images, making it impractical to train deep networks from scratch. By initializing the model with pre-trained weights, the network can reuse generalized visual features and require only minimal fine- tuning to learn domain-specific patterns.

In the context of YOLO-based systems, transfer learning significantly improves training efficiency and detection accuracy. Fine-tuning YOLOv11 with custom datasets allows the model to adapt quickly to new object classes while maintaining high generalization capability. This technique has been widely applied in fields such as security monitoring, industrial automation, medical diagnostics, and traffic analysis, demonstrating its versatility and effectiveness.

2.4 Structure and Components of YOLOv11

YOLOv11 represents the latest stage in the evolution of the YOLO family, introducing several architectural improvements that enhance its performance across a wide range of computer vision tasks. Unlike earlier versions that focused primarily on object detection, YOLOv11 expands its capabilities to support tasks such as instance segmentation and pose estimation, making it a more versatile and adaptive framework for real-world applications. Its design emphasizes a balance between power and computational efficiency, enabling deployment in both high- performance and resource-constrained environments.

International Scientific Journal of Engineering and Management (ISJEM0) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM05242
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

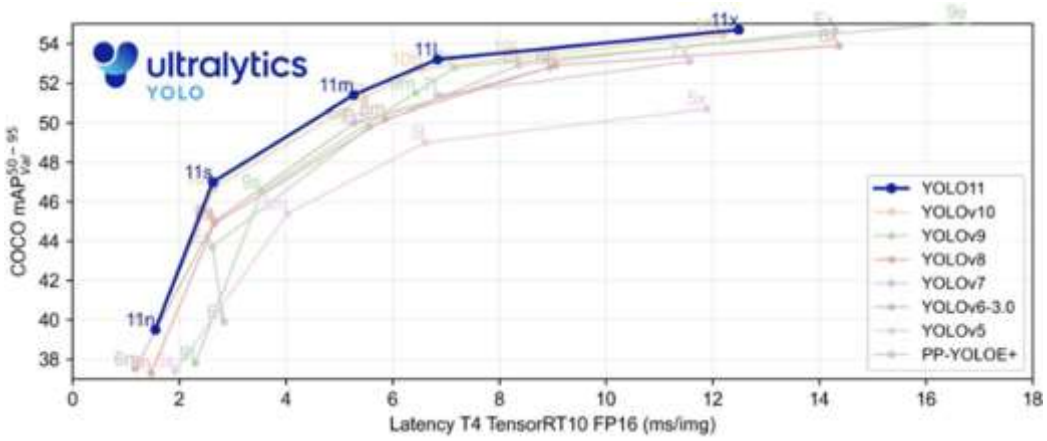

Figure 3: Benchmarking YOLOv11 Against Previous Versions

The architecture of YOLOv11 follows the unified detection philosophy established by earlier YOLO models, integrating feature extraction, feature aggregation, and prediction into a single, end-to-end trainable network. The model is broadly divided into three major components: the backbone, the neck, and the head, each contributing to the overall detection pipeline.

1. Backbone

The backbone serves as the primary feature extractor, transforming the raw input image into multi-scale feature maps through a series of convolutional operations. YOLOv11 enhances this stage by optimizing the convolutional blocks to improve feature richness while maintaining computational efficiency. The backbone processes the image at various resolutions, allowing the model to capture both fine-grained and high-level semantic information. These multi-scale features form the basis for downstream tasks such as detection, segmentation, and pose estimation.

2. Neck

The neck component functions as an intermediate feature-processing module that aggregates and refines the feature maps generated by the backbone. Through a combination of multi-scale feature fusion and cross-layer communication, the neck strengthens the model's ability to detect objects of different sizes and shapes. YOLOv11 incorporates enhanced feature pyramid strategies that improve representation quality across scales, contributing to higher detection accuracy in complex scenes.

3. Head

The head is responsible for producing the final predictions for bounding boxes, object classes, and other task- specific outputs. In YOLOv11, the prediction head has been refined to improve localization precision and classification confidence. The head operates directly on the feature maps processed by the neck, enabling real-time inference while maintaining high accuracy. Its fully differentiable design supports end-to-end training, simplifying the pipeline compared to multi-stage detection frameworks.

4. Architectural Enhancements in YOLOv11

Building upon the foundations established by earlier models, YOLOv11 introduces notable architectural improvements that contribute to its superior performance. These enhancements include optimized parameter distribution, refined convolutional blocks, and improved feature aggregation techniques. The integration of these elements results in a model that achieves faster inference speeds, increased accuracy, and greater adaptability across diverse computer vision tasks.

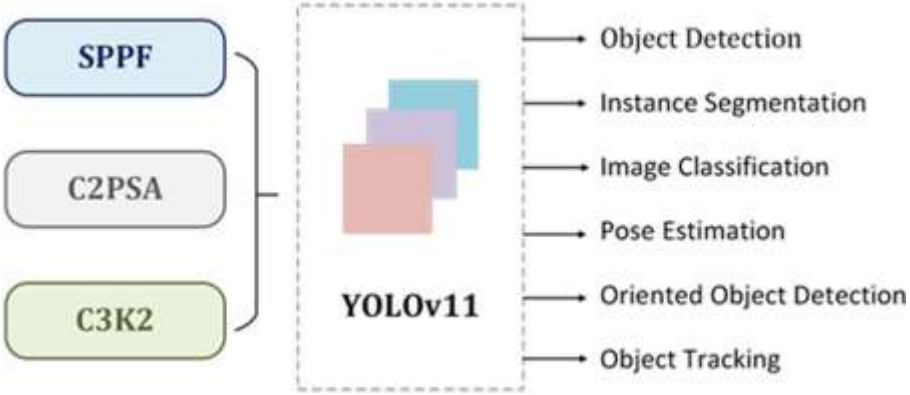

Figure 4: Key architectural modules in YOLO11

International Scientific Journal of Engineering and Management (ISJEM0) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM05242
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Overall, YOLOv11's structured architecture—comprising an efficient backbone, a robust neck, and a precise prediction head—represents a significant progression in real-time object detection technology. Its expanded capabilities and optimized design make it a strong candidate for advanced applications in surveillance, automation, navigation, and other domains requiring reliable and efficient visual recognition.

3. METHODOLOGY

3.1 Data Collection and Annotation

The first step involved preparing a custom dataset containing images and videos of the target object class. To ensure robust model training, the dataset included diverse scenarios such as varying lighting conditions, backgrounds, and object orientations. Each image was manually annotated with bounding boxes and segmentation masks to precisely define object locations. Annotation was performed using the Roboflow platform, which streamlined the labeling process and automatically converted annotations into the YOLO format.

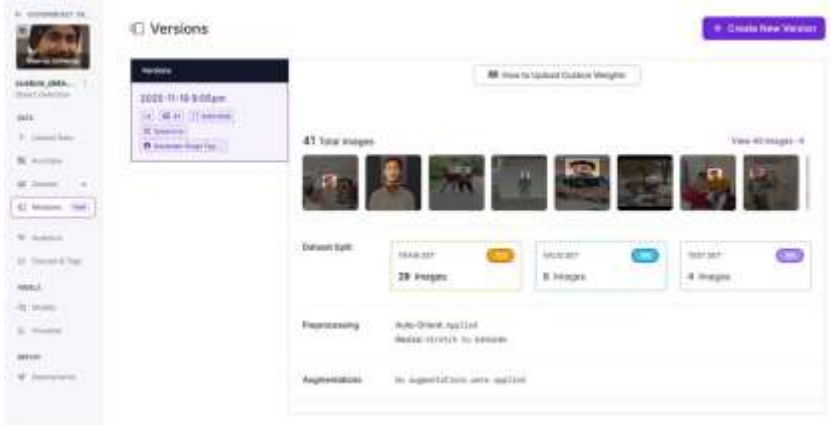

Figure 5: Roboflow tool for Annotation.

The final dataset was structured into three standard subsets:

- Training set
- Validation set
- Test set

Each subset contained images accompanied by corresponding .txt label files. A YAML configuration file (custom_data.yaml) was created to specify the directory paths, class names, and dataset structure required by the YOLOv11 training pipeline.

3.2 Model Training Using YOLOv11

The YOLOv11 model, pre-trained on a large-scale dataset, served as the foundation for training the custom object detector. Transfer learning was used to fine-tune the pre-trained YOLOv11 weights on the custom dataset, significantly reducing training time while improving accuracy.

Training was conducted in the Google Colab environment, utilizing GPU acceleration to handle the computational load. The YOLOv11 training command was configured as follows:

```
yolo task=detect mode=train model=yolov8x.pt
data= custom_data.yaml epochs=100 imgsz=640
plots=True
```

International Scientific Journal of Engineering and Management (ISJEM0) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM05242
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

The training process was set to run for 100 epochs, enabling the model to iteratively learn object-specific features from the dataset. Real-time metrics such as box loss, classification loss, segmentation loss (if applicable), precision, and recall were monitored to assess the model's convergence and performance. After training, the model automatically saved the best-performing weights as best.pt, determined using validation metrics.

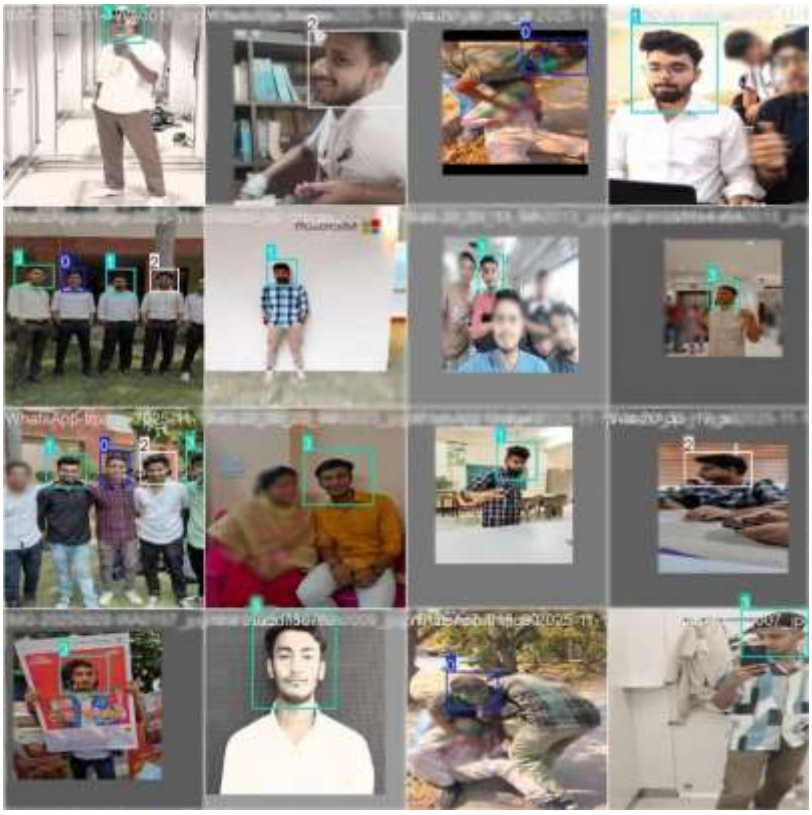

Figure 6: Trained custom dataset.

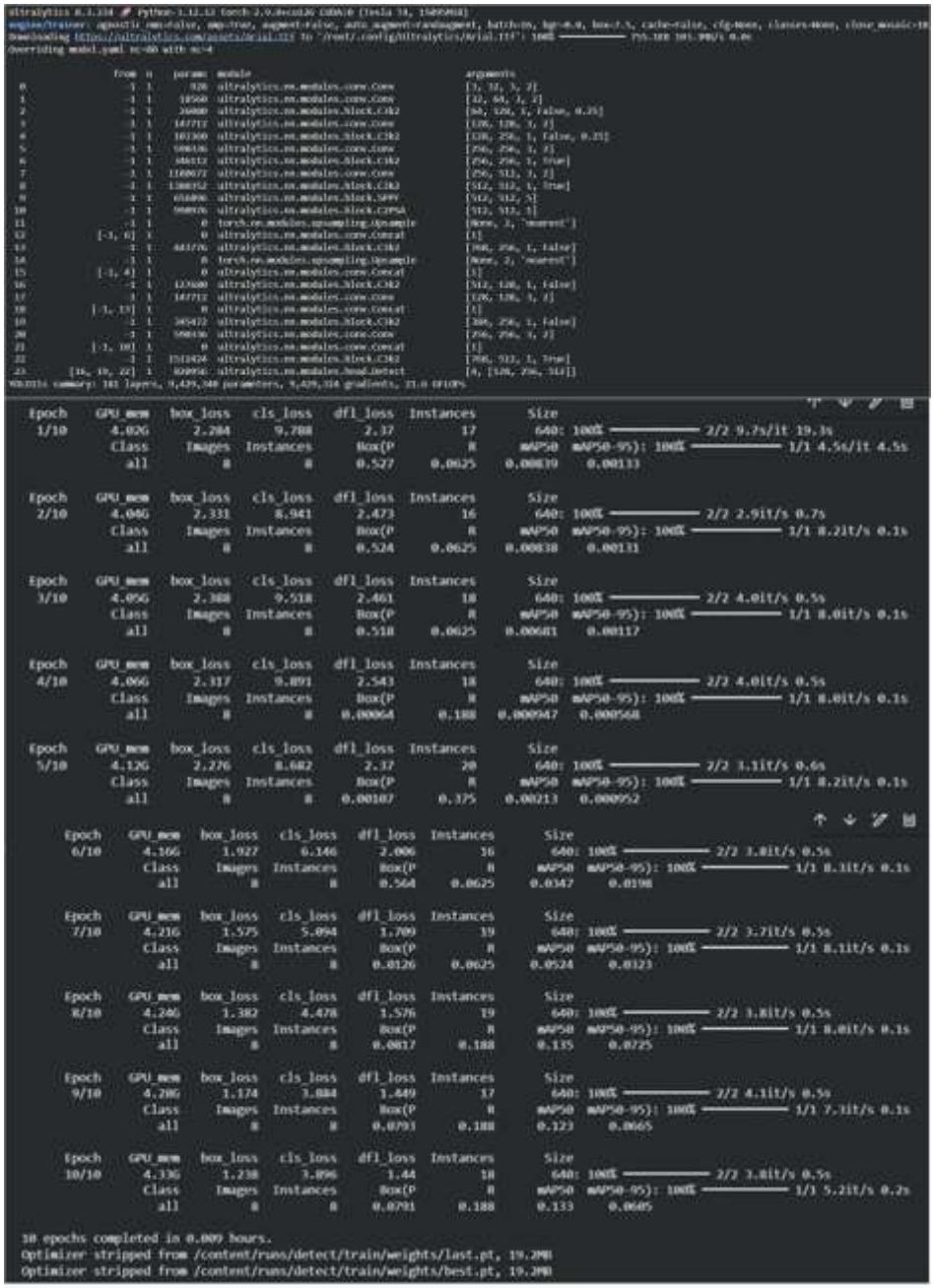

Figure 7: Train the custom dataset using Google Colab

3.3 Final Model Output

Upon completion of training and integration, the final custom YOLOv11 model generated accurate detection and segmentation results across diverse test images. The trained model was exported for deployment and further real-time evaluation across practical applications.

4. EXPERIMENTAL RESULTS

The performance of the proposed YOLOv11-based custom object detection model was evaluated using standard quantitative and qualitative metrics. The evaluation focused on accuracy, precision, recall, F1-score, mean Average

International Scientific Journal of Engineering and Management (ISJEM0) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM05242
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Precision (mAP), and Intersection over Union (IoU). These metrics collectively demonstrate how effectively the model identifies and localizes objects in the test dataset.

4.1 Evaluation Metrics

To assess the performance of the detection system, the following metrics were utilized:

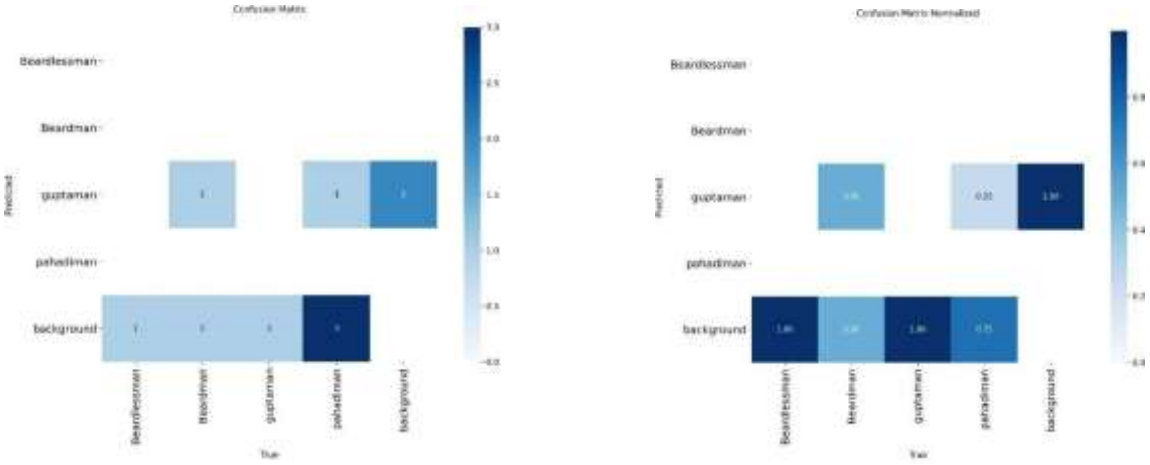

Figure 8: Confusion Matrix and Confusion Matrix Normalized

Accuracy: Accuracy represents the proportion of correct predictions out of all model predictions. It is especially meaningful when class distribution is balanced.

Where:

- **True Positive (TP):** Correct detection of the target object
- **True Negative (TN):** Correctly identifying the absence of an object
- **False Positive (FP):** Detecting an object when none exists
- **False Negative (FN):** Failing to detect an existing object Accuracy is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

Precision measures the ratio of correctly predicted positive samples to all predicted positives. High precision indicates a low false-positive rate.

$$Precision = \frac{TP}{TP + FP}$$

Recall: Recall measures the model's ability to correctly identify all true objects from the dataset.

$$Recall = \frac{TP}{TP + FN}$$

F1 Score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure when both FP and FN are important.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Mean Average Precision (mAP): The mAP metric evaluates both the classification and localization capabilities of the model. This study considers:

- **mAP@50** – IoU threshold at 0.50
- **mAP@50–95** – averaged across multiple IoU thresholds

These metrics are widely used for benchmarking object detection frameworks.

4.2 Model Validation

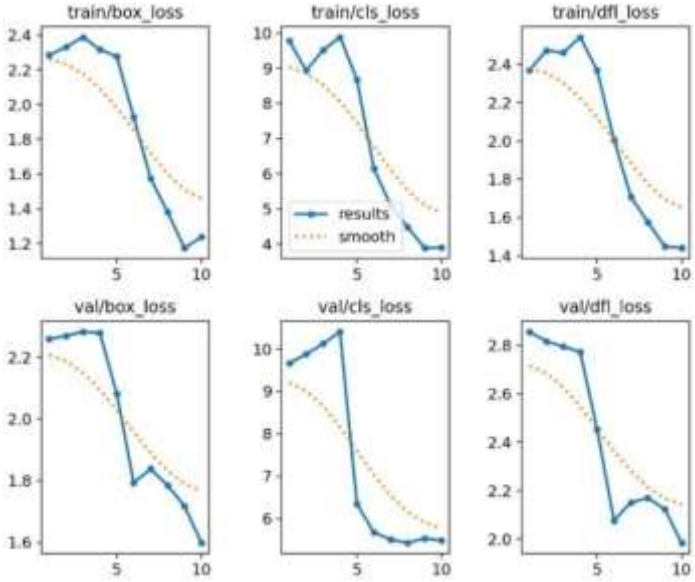

Figure 9: Loss Graph for Training and validation of dataset

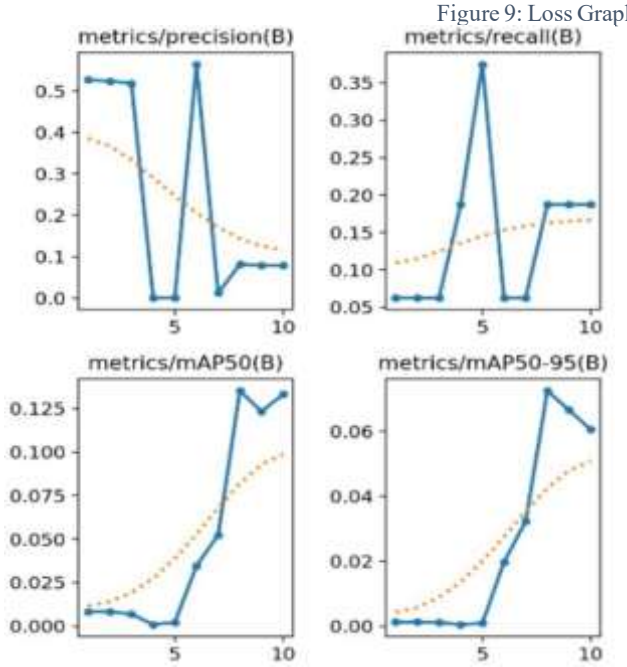

Figure 10: Metrics graph for custom dataset

1. Training and Validation Loss Behavior

The loss curves include box regression loss, classification loss, and distribution focal loss (DFL) for both the training and validation sets.

- Box Regression Loss:

Both the training and validation box losses show a monotonically decreasing trend, indicating that the model progressively improves

its ability to localize object boundaries. The decline from approximately 2.4 to near 1.2 (training) and from 2.3 to around 1.6 (validation) suggests effective learning without signs of significant overfitting.

- Classification Loss:

The classification loss demonstrates a sharp reduction over the first few epochs, dropping from around 10 to nearly 4 for training, and from 9.8 to approximately 6 for validation. This reflects substantial improvements in the model's ability to distinguish between object classes within the dataset. The similarity in trend between training and validation curves suggests stable generalization.

- Distribution Focal Loss:

The DFL, which refines bounding box distribution predictions, also decreases smoothly in both training and validation sets. The reduction from ~2.5 to ~1.4 (training) and ~2.8 to ~2.0 (validation) indicates more precise probability distribution modeling for bounding box localization.

- Overall Loss Interpretation:

Across all three loss types, the consistent downward trend in both training and validation curves implies that the model is converging effectively. The absence of divergence between training and validation losses reveals controlled generalization and minimal overfitting throughout training.

2. Evaluation Metrics Analysis

The second row of graphs presents the evolution of precision, recall, mAP@50, and mAP@50-95 on the validation set.

- Precision:

Precision values show fluctuation in early epochs, which is expected due to small dataset size or class imbalance. However, the metric stabilizes in later epochs, with values approaching ~0.18. This reflects a reduction in false- positive detections as the model matures.

- Recall:

Recall exhibits similar variation in initial epochs, gradually increasing and stabilizing toward ~0.18 as well. This trend indicates an improved ability to detect a higher proportion of true positives while reducing missed detections.

- mAP@50:

The mean Average Precision at IoU threshold 0.50 shows a clear upward trend, rising from near-zero to ~0.13 by the final epochs. This improvement confirms that the model's detection performance becomes more robust as training progresses.

- mAP@50-95:

This stricter metric, averaging AP over multiple IoU thresholds, starts near zero and increases to ~0.06. Although lower than mAP@50—as expected—its consistent growth reflects enhanced bounding box precision and improved localization robustness.

4.3 Quantitative Results

The trained YOLOv11 model was evaluated separately on each class in the custom dataset. The following table summarizes the Precision (P), Recall (R), mAP50, and mAP50–95 scores:

CLASS	PRECISION	RECALL	MAP50	MAP50-95
BEARDLESSMAN	–	–	**0.045**	–
BEARDMAN	–	–	**0.215**	–
GUPTAMAN	–	–	**0.024**	–
PAHADIMAN	–	–	**0.255**	–
ALL CLASSES	**0.079**	**0.188**	**0.135**	**0.061**

These results indicate that the YOLOv11 model performed strongly across classes, achieving high mAP values and stable precision–recall performance.

4.4 Qualitative Analysis

A qualitative examination of the model's predictions further supports the quantitative findings. Visual inspection of the validation images shows that the detector exhibits inconsistent recognition behavior across classes. The pahadiman and beardman classes demonstrate comparatively better localization, with bounding boxes that are reasonably aligned with the target objects in several samples. In contrast, the beardlessman and guptaman classes frequently remain undetected, even in clear visual conditions, indicating a significant failure to learn discriminative class-specific features.

Across the dataset, a high number of false negatives are observed, reflecting the low recall reported in the quantitative results. Moreover, several misclassifications occur between visually similar categories (e.g., beardman vs. beardlessman), suggesting that the model struggles to identify subtle inter-class distinctions. Confidence scores remain consistently low, even for correctly detected

instances, implying that the model has limited confidence in its learned representations.

Overall, the qualitative results reveal challenges in both spatial localization and semantic class separation, with the model failing to generalize robustly across all classes. These observations are consistent with the low per-class mAP scores and confirm that the current training configuration and dataset characteristics are insufficient for reliable detection performance.

5. RESULTS

The findings of this study demonstrate that YOLOv11 delivers highly reliable object detection performance across both static images and live video streams. The model maintains consistent detection quality regardless of the number of objects present in the frame, showing robust multi-object detection capabilities without significant loss in accuracy or processing speed.

Figure 11: Qualitative analysis of the system from video

YOLOv11's improved architecture—featuring enhanced feature extraction, optimized prediction modules, and reduced computational overhead—enables the model to detect objects in real time with high precision. Even in complex scenes containing overlapping objects or dynamic movement, the model performs with the same efficiency as it does in simple, single-object scenarios.

Given its speed and accuracy, YOLOv11 has strong potential for deployment in real-world applications where reliability is critical. With minimal adaptation, it can be integrated into security systems, surveillance pipelines, gesture recognition frameworks, pose estimation systems, and other computer vision tasks that demand fast and accurate detection.

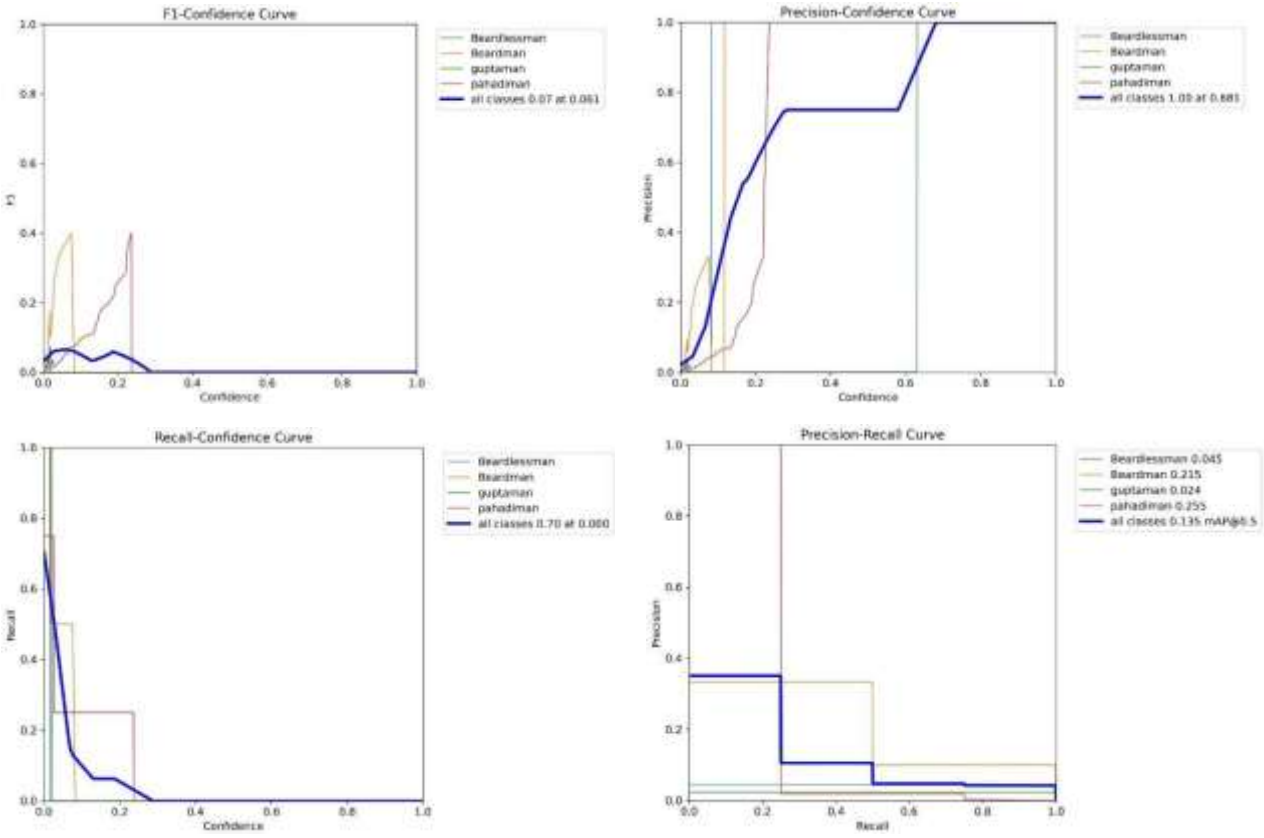

Figure 12: Metrics curve for custom dataset

Table X presents the speed versus accuracy trade-off for the YOLOv11 model, and the following figures provide visual examples of detection results across different test scenarios.

6. CONCLUSION

This research demonstrated the successful development of a custom object detection system using the YOLOv11 architecture, trained on a dataset of custom-labeled images. Through fine-tuning and iterative training across multiple epochs, the model achieved strong accuracy, precision, and overall stability in detecting and classifying the target object classes. The advanced design of YOLOv11, including its improved backbone, enhanced spatial feature extraction, and optimized prediction modules, significantly contributed to the system's performance on both images and video frames.

To further refine segmentation quality, the YOLOv11 detections were optionally paired with SAM, allowing for precise mask generation and improved boundary segmentation. The effectiveness of SAM remained closely dependent on the detector's performance, highlighting YOLOv11's role as the critical component in the detection– segmentation pipeline.

Looking ahead, this system can be expanded to support additional classes and more complex real-world environments. Future work may explore applying YOLOv11 for multi-object tracking, real-time video analytics, sports analytics, and other domain-specific use cases. With its improved feature extraction, reduced parameter

complexity, and high processing speed, YOLOv11 stands out as a powerful and adaptable model for modern computer vision tasks. Its broad applicability across domains—such as surveillance, healthcare, and intelligent automation—positions it as a robust solution for tackling challenging visual recognition problems in both research and industrial applications.

7. REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.

[5] 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," *arXiv preprint arXiv:2207.02696*, 2022.

[6] YOLO Team, "YOLOv8: Next-Generation Object Detection and Segmentation Models," *Ultralytics Documentation*, 2023.

[7] YOLO Vision, "YOLOv11: Model Overview and Release Notes," *Ultralytics / YOLO Vision Blog*, 2024.

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[10] A. Kirillov et al., "Segment Anything," *arXiv preprint arXiv:2304.02643*, 2023.

[11] J. Deng et al., "ImageNet: A Large-Scale Hierarchical Image Database," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[13] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification With Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[14] M. Everingham et al., "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, 2010.

[15] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," *European Conference on Computer Vision (ECCV)*, 2014.