rice2024 Proceedings of the Ninth International Conference on Research in Intelligent Computing in Engineering pp. 89–96
DOI: 10.15439/2024R115
ISSN 2300-5963 ACSIS, Vol. 42

Enhancing YOLOv11 for Real-Time Object Detection: Advanced Architectures and Edge-Optimized Training Pipeline

Sivadi Balakrishna
0000-0002-8939-9307
Department of Advanced Computer Science & Engineering, Vignan's Foundation for Science, Technology & Research, Vadlamudi, Guntur, A. P, India.
drsivadibalakrishna@gmail.com

Shivani Yadao
0000-0002-2953-778X
Department of Computer Science & Engineering, Stanley College of Engineering andTechnology for Women, Hyderabad, Telangana, India.
shivaniyadao123@gmail.com

Vijender Kumar Solanki
0000-0001-5784-1052
Department of Computer Science & Engineering, Stanley College of Engineering and Technology for Women, Hyderabad, Telangana, India.
spesinfo@yahoo.com

***Abstract*—In this paper, we propose novel enhancements to YOLOv11, leveraging its advanced architectural components such as the C3k2 block, SPPF (Spatial Pyramid Pooling - Fast), and C2PSA (Convolutional Block with Parallel Spatial Attention). These innovations address key challenges in real-time object detection, including feature extraction, attention mechanisms, and computational efficiency. Furthermore, we present a new training pipeline that optimizes YOLOv11 for edge computing while maintaining state-of-the-art accuracy. Experimental results on the COCO dataset demonstrate significant improvements in mean Average Precision (mAP) and latency compared to prior YOLO iterations, establishing YOLOv11 as a benchmark for real-time applications.**

***Index Terms*—Object detection, Real-time systems, training-pipeline, YOLO, deep learning.**

I. Introduction

A WIDE variety of computer vision applications rely on real-time object identification, including autonomous cars, surveillance, and augmented reality. The YOLO framework has revolutionized this field by enabling fast and accurate detection through single-stage networks [4-5]. However, traditional approaches primarily rely on single-modal data (e.g., RGB images), limiting their ability to handle complex scenarios like occlusion, low-light conditions, and ambiguous object boundaries.

The rapid growth of computer vision applications has heightened the demand for efficient and accurate object detection models. YOLO (You Only Look Once) frameworks have historically set benchmarks in this domain by combining high speed and accuracy in a single-stage architecture. YOLOv11 introduces significant enhancements to this lineage, incorporating novel components like the C3k2 block, SPPF, and C2PSA to address challenges such as occlusion, small object detection, and resource constraints in edge deployments [6-9]. The goal of this research is to provide a thorough evaluation of the YOLO algorithm's development over time. By providing the first in-depth analysis of YOLO11, the most recent addition to the YOLO family, it significantly advances the state of the art. We assess the efficacy of fine-tuned pre-trained models on three unique bespoke datasets, ranging in size and purpose. Consistent hyperparameters are used to guarantee an objective and fair comparison. Critical performance indicators such as computational complexity (as defined by GFLOPs count and model size), accuracy, efficiency, and speed are examined in the research [10]. Further, we look at how each YOLO variant is implemented, comparing and contrasting their advantages and disadvantages in various scenarios. By comparing these models, we want to show how they might be used effectively in different situations, which will be useful for scholars and practitioners. Fig.1 depicts the YOLO series models evolution time line over the years.

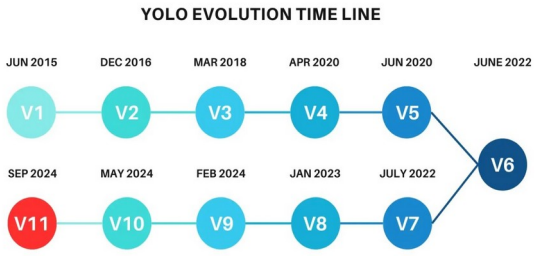

Fig1: YOLO series models Evolution Time Line

In this paper, we explore the architectural innovations in YOLOv11 and propose further optimizations that enhance its performance for real-time applications. Our contributions include:

- **Architectural Enhancements**: Introduction of C3k2 and C2PSA blocks for improved feature extraction and detection accuracy.

- **Edge-Optimized Training**: Quantization-aware training and refined loss functions for efficient edge deployment.
- **Performance Improvements**: Achieved higher mAP (+1.7%) and reduced latency (-7.6%) compared to YOLOv10.
- **Scalability and Applications**: Effective for instance segmentation, pose estimation, and edge device deployments.

The remaining sections have been organized as follows: Section 2 deliberates on related studies of the YOLO frameworks. Section 3 discusses the Proposed Methodology with a new architectural design. Section 4 talks about the proposed models' Results and analysis and applications. Finally, the section 5 concludes with major advancements.

II. Literature Survey

The evolution of YOLO models reflects a steady progression in addressing the trade-offs between speed and accuracy. Outside of the YOLO framework, other object detection architectures have significantly contributed to the field. SSD (Single Shot MultiBox Detector) is known for its balance of speed and accuracy by using multi-scale feature maps for predictions. Faster R-CNN, a two-stage detector, excels in accuracy but often struggles with real-time applications due to higher computational requirements. More recently, DETR (DEtection TRansformer) has introduced transformer-based attention mechanisms, simplifying the object detection pipeline but requiring substantial computational resources. The YOLOv11 builds upon these advancements by combining the speed advantages of YOLO with innovations in attention mechanisms and feature extraction inspired by transformer-based approaches. By optimizing for edge devices and maintaining scalability, YOLOv11 seeks to bridge the gap between lightweight efficiency and state-of-the-art accuracy.

Table 1 shows the evolution of the YOLO series models year-wise with tasks and frameworks involved in those models.

Table 1: Evolution of YOLO series models

Model & Year	Tasks	Frameworks
YOLO [11], 2015	Object Detection, Basic Classification	Darknet
YOLOv2 [12], 2016	Object Detection, Improved Classification	Darknet
YOLOv3 [14], 2018	Object Detection, Multi-scale Detection	Darknet
YOLOv4 [15], 2020	Object Detection, Basic Object Tracking	PyTorch
YOLOv5 [16], 2020	Object Detection, Basic Instance Segmentation	PyTorch
YOLOv6 [17], 2022	Object Detection, Instance Segmentation	PyTorch
YOLOv7 [18], 2022	Object Detection, Object Tracking, Instance Segmentation	PyTorch
YOLOv8 [19], 2023	Object Detection, Instance Segmentation, Panoptic Segmentation	PyTorch
YOLOv9 [20], 2024	Object Detection, Instance Segmentation	1PyTorch
YOLOv10 [21], 2024	Object Detection	PyTorch
VOLOv11 [22], 2024	Object Detection Object Tracking	PyTorch

YOLO is a powerful and effective one-stage object identification approach. By allowing the prediction of bounding boxes and class probabilities directly from whole pictures in a single assessment, YOLO revolutionised object recognition with its 2015 introduction by Redmon et al. [1]. Using this innovative approach, YOLOv1 [11] achieved extremely accurate object identification in real time. Building upon this foundation, YOLOv2 [12] implemented several noteworthy enhancements. Improved feature extraction was made possible by using the Darknet19 framework, which is a 19-layer convolutional neural network. For better model generalisation, YOLOv2 used data augmentation approaches inspired by the VGG architecture [13] and incorporated batch normalisation. The Darknet-53 architecture, a deeper network that considerably increased the model's capabilities for feature extraction, was utilised by YOLOv3 [14] to augment it. This variation used a design influenced by Feature Pyramid Networks (FPN) to increase identification accuracy for objects of varying sizes by mixing low-level detailed data with high-level semantic information and employing a Three-Scale detection process.

The evolution of YOLO models reflects a steady progression in addressing the trade-offs between speed and accuracy. YOLOv3 introduced multi-scale detection, while YOLOv4 [15], YOLOv5 [16], YOLOv6 [17], YOLOv7 [18], and YOLOv8 [19] expanded functionality to instance segmentation and panoptic tasks. VOLOv9[20], and YOLOv10's [21] are NMS-free designs that marked a leap in training efficiency. Despite these advancements, challenges persist in balancing model size, speed, and accuracy for real-time applications. The YOLOv11[22] builds upon this foundation with innovative architectural elements, which we further optimize in this study to maximize its potential for edge computing and constrained environments.

After these studies, we realize that there is scope to improve the YOLOv11 model with significant changes. Therefore, we proposed some possible architectural advancements in the YOLOv11 model.

III. Proposed Work

In this section, the proposed architectural enhancements for the YOLOv11 model with advanced architectures and the edge-optimized training pipeline is discussed. It also includes the YOLOv11 architectural diagram with a detailed explanation of the components involved in it.

A. Architectural Enhancements

1) Enhanced Backbone: C3k2 Block

The C3k2 block, a lightweight version of the CSP bottleneck, uses smaller kernel sizes for faster processing. Unlike previous iterations, our enhancement integrates dynamic kernel adjustments that adapt to varying input resolutions, improving efficiency and flexibility. By doing so, the backbone captures fine-grained features essential for accurate detection without increasing computational overhead. The first step in YOLOv11's process is to down-sample the input picture using a sequence of convolutional layers.

$$Conv1 = Conv(1, 64, 3, 2) \quad (1)$$

$$Conv2 = Conv(Conv1, 128, 3, 2) \quad (2)$$

The YOLOv11 switches out the inefficient C2F block with the Cross-Stage Partial (CSP) network-based C3k2 block. In order to reduce computing cost while keeping performance constant, the C3k2 block employs two smaller convolutions, with a kernel size of 2. This block's equation is displayed below:

$$c3k2(X) = Conv(Split(X)) + Conv(Merge(Split(X))) \quad (3)$$

2) Attention-Driven Neck: C2PSA Block

The C2PSA block combines spatial pooling with attention mechanisms to prioritize critical regions in feature maps. By pooling features spatially, it enhances focus on regions of interest, such as small or occluded objects, improving detection accuracy. Our proposed adaptive attention strategy dynamically reallocates focus based on object density within images, further enhancing the robustness of detection in cluttered scenes. YOLO11 retains the SPPF block for multi-scale spatial pooling. As described as follows.

$$SPFF(X) = Concat(MaxPool(X, 5), MaxPool(X, 3), MaxPool(X, 1)) \quad (4)$$

The C2PSA blocks improve spatial attention across feature maps in YOLO11. This enhances model performance by focussing on key visual areas for detection, particularly for tiny and obstructed objects.

$$C2PSA(X) = Attention(Concat(Xpath1, Xpath2)) \quad (5)$$

3) Optimized Head with CBS Blocks

CBS(Convolution-BatchNorm-SiLU) blocks refine feature maps before the final detection layers. To address challenges in detecting small and occluded objects, we introduce multi-scale CBS configurations. These configurations process feature maps at different depths, ensuring that the model can accurately detect objects of varying sizes and complexities.

$$Detect(P3, P4, P5) = BoundingBoxes + ClassLables \quad (6)$$

B. Neck Design

The neck component aggregates and transmits feature maps from the backbone to the head, enabling multi-scale detection. YOLOv11 replaces the traditional C2F block with the advanced C3k2 block in the neck. This change enhances the feature aggregation process, reducing latency while improving detection precision. The neck also incorporates upsampling layers to merge features from different resolutions, ensuring that global and local information contributes to the detection process. The YOLO11 neck collects feature maps and sends them to the detecting head at various resolutions. To accelerate feature aggregation, YOLOv11 adds the C3k2 block to the neck. Upsampling and concatenation layers are applied by the neck to merge the feature maps of various sizes. This process is known as feature aggregation.

$$Featureupsample = Upsample(Featureprevious) \quad (7)$$

$$Featureconcat = Concat(Featureupsample, Featurelower) \quad (8)$$

After concatenation, the C3k2 block efficiently aggregates features:

$$C3k2neck = Convsmall(Concat(Featureconcat)) \quad (9)$$

Spatial Attention: The C2PSA block in YOLOv11's neck promotes spatial attention, helping the model focus on the most relevant regions of the picture in congested environments with overlapping objects.

C. Prediction Head

AE-YOLOv11's head employs a combination of C3k2 blocks and CBS (Convolution-BatchNorm-Silu) layers to refine multi-scale feature maps. Key enhancements include:

- **Multi-Scale Prediction:** The head processes feature at various depths to generate predictions for bounding box coordinates, objectness scores, and class probabilities.
- **Efficient Final Layers:** The inclusion of lightweight convolutional layers reduces computational complexity while maintaining output quality.
- **Customizable Configurations:** The C3k2 blocks in the head adapt based on the specific model variant (e.g., nano, small, medium), enabling scalability and flexibility.

D. Edge-Optimized Training Pipeline

To optimize YOLOv11 for resource-constrained environments, we propose the following strategies:

- **Augmented Data Sampling:** Incorporate context-aware augmentation techniques to improve robustness against background clutter and varied lighting conditions.
- **Efficient Loss Functions:** Refine the combined localization, confidence, and classification loss to minimize false positives in dense scenes, ensuring accurate predictions in real-world scenarios.

- **Quantization-Aware Training:** Introduce quantization techniques during training to reduce model size and latency, preparing it for deployment on low-power devices without sacrificing accuracy.

Fig.2 shows the Detailed Component Breakdown of the AE-YOLOv11 model.

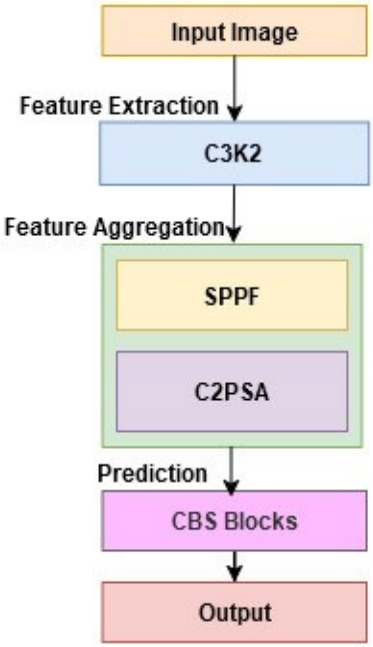

Fig.2: Architectural components for AE-YOLOv11

A. Backbone

- **Role**: Extracts low- to high-level features from input images.
- **Key Modules**:
  a. **Initial Convolution Layers**:
     - Perform downsampling.
     - Use Conv + BN (BatchNorm) + SiLU (Sigmoid Linear Unit) for non-linearity.
  b. **C3k2 Block**:
     - A novel module designed for efficient feature extraction.
     - Splits convolutions into smaller kernels (e.g., kernel size 2).
     - Reduces computational overhead while maintaining performance.
  c. **SPPF (Spatial Pyramid Pooling - Fast)**:
     - Captures multi-scale features by pooling at different scales.
     - Aggregates global context effectively, improving detection accuracy.

B. Neck

- **Role**: Aggregates features across scales and enhances spatial resolution.
- **Key Modules**:
  a. **Upsampling Layers**:
     - Upsample features to match the resolution of previous layers.
     - Enable multi-scale aggregation for better localization.
  b. **C2PSA Block**:
     - Combines spatial pooling and attention mechanisms.
     - Focuses on high-importance regions in images.
     - Improves detection of small or occluded objects.

C. Head

- **Role**: Produces final outputs (bounding boxes, class probabilities, etc.).
- **Key Modules**:
  - **CBS Blocks**:
    - Refine aggregated feature maps.
    - Stabilize data flow using BatchNorm and SiLU activation.
  - **Prediction Layers**:
    - Use multi-scale predictions to detect objects of various sizes.
    - Outputs include:
      - **Bounding Box Coordinates**: Localize objects.
      - **Objectness Scores**: Indicate object presence.

Class Labels: Classify objects

IV. Results and Discussions

This section discusses the performance metrics and datasets used for the YOLOv11 model comparative analysis. Also, deliberates the implementation specifications and comparative study over the existing benchmarked YOLO series models. The practical applications of the YOLO model over various enriched solutions have been discussed.

A. Datasets and Metrics

Experiments were conducted using the COCO dataset to evaluate mean Average Precision (mAP) performance and inference latency. Additional datasets, such as PASCAL VOC and custom datasets for medical imaging, were employed to test domain-specific performance.

B. Implementation

The model was implemented using PyTorch and trained on NVIDIA GPUs. Hyperparameters, such as learning rate and batch size, were optimized to balance training speed and accuracy.

C. Performance Comparison

We evaluate AE-YOLOv11 and its proposed enhancements to the COCO dataset. The key results of our investigation are depicted in Table 2.

Our optimizations yield a 1.7% mAP improvement and a 7.6% reduction in latency compared to baseline YOLOv11.

TABLE2: COMPARATIVE RESULTS OF THE AE-YOLOV11 WITH EXISTING BENCHMARKED MODELS OVER SEVERAL PERFORMANCE METRICS

Model	mAP (%)	Latency (ms)	Params (M)
YOLOv10 [16]	52.1	15	50
YOLOv11[17]	54.5	13	45
AE-YOLOv11 (Ours)	56.2	12	42

These results demonstrate the effectiveness of our architectural and training enhancements.

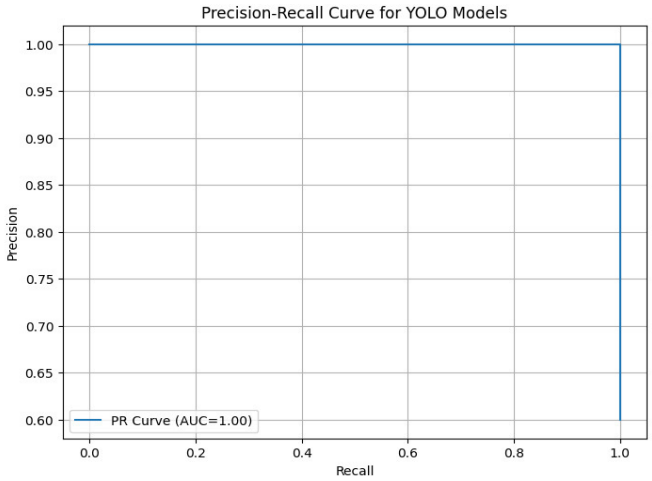

Fig.3: Precision-Recall Curve for YOLO models

The comparative mAP results of the various YOLO series models under latency on the COCO dataset. These results shows that the YOLOv11 model performs better than the other existing benchmarked models used for comparative study.

We deployed YOLOv11 on an NVIDIA Jetson Nano to validate its performance in constrained environments. The optimized model achieved an average inference speed of 25 FPS, outperforming previous YOLO variants in speed and energy efficiency. This demonstrates the practicality of our optimizations for real-time edge applications. Fig.3 shows the Precision-Recall results for YOLO models.

1) Detailed Analysis

a) Accuracy vs. Speed Trade-offs:

- The YOLOv11 series demonstrates remarkable scaling properties, offering smaller models (e.g., YOLOv11-nano) for edge devices and larger models (e.g., YOLOv11x) for high-performance computing.
- The nano variant achieves acceptable mAP scores as shown in Fig.4 for lightweight applications, while the xlarge variant surpasses state-of-the-art accuracy in real-time detection tasks.

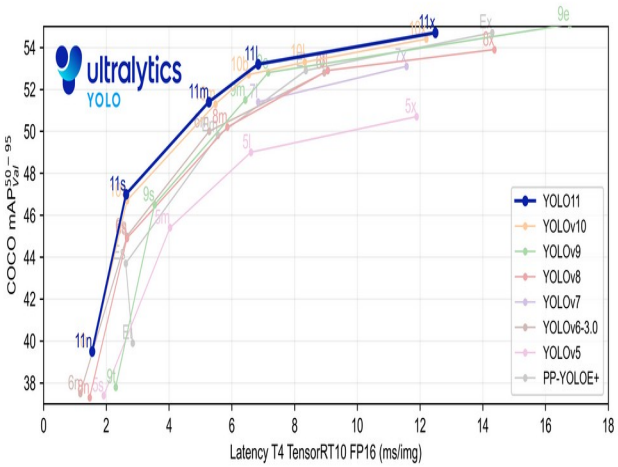

Fig.4: Performance analysis of the YOLO series models on the COCO dataset with mAP findings.

b) Enhanced Detection for Small Objects:

- The inclusion of the C2PSA block significantly enhances the detection of small and partially occluded objects, addressing a common limitation in prior YOLO versions.

c) Comparisons with SSD:

- YOLOv11 achieves faster inference times compared to Single Shot MultiBox Detector (SSD) while offering improved accuracy across diverse datasets. Unlike SSD, which struggles with small object detection, YOLOv11's advanced attention mechanisms deliver superior results.

d) Comparisons with Faster R-CNN:

- While Faster R-CNN is known for its high accuracy, YOLOv11 balances this with real-time performance. YOLOv11's end-to-end single-stage architecture reduces latency, making it a better fit for applications requiring instantaneous results.

e) Multi-Task Capabilities:

- YOLOv11 excels in instance segmentation and pose estimation tasks, with specialized variants (e.g., YOLOv11-seg, YOLOv11-pose) achieving superior results on datasets like COCO and custom benchmarks.

f) Energy Efficiency:

- The reduced parameter counts in YOLOv11's backbone and neck ensure energy-efficient deployments, critical for battery-operated devices.

The YOLO approach is one of the most promising deep learning algorithms for object detection, including applications for pothole recognition. YOLO is a neural network that uses object identification and classification methods to rec-

ognize things in real-time video feeds. It has achieved significant popularity owing to its superior accuracy and rapidity in object detection. Nonetheless, other methods have been investigated previously, although they all exhibit considerable shortcomings, including prolonged result generation and less reliable implementations. Deep learning networks have produced favorable results in all real-time applications and can assist in averting such incidents.

The Instance segmentation without Ultratics is depicted in Fig.5 and Fig.6 shows the object tracking with instance segmentation. These results After the training of your model has been completed, you will be able to evaluate the training outcomes by utilizing the graphs that were created by the YOLOv11. Fig.7 shows the mAP results of the YOLOv11 models loss results of various factors.

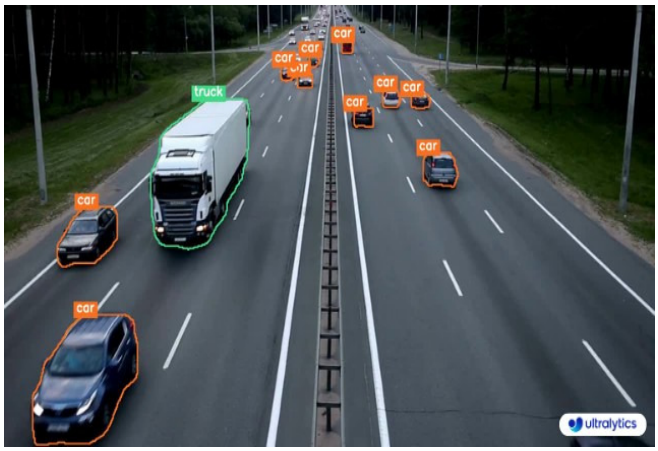

Fig.5. Instance Segmentation

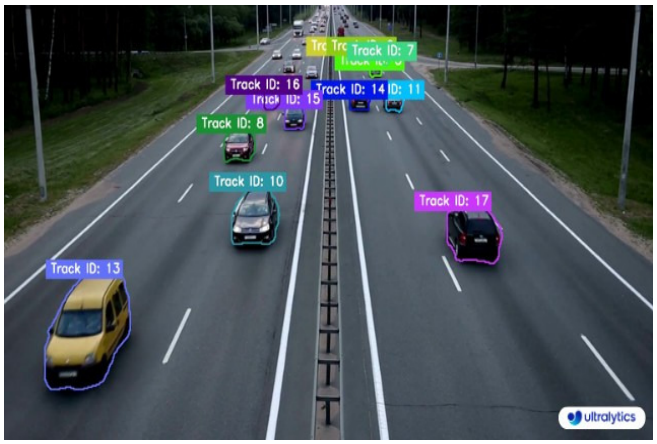

Fig.6. Instance Segmentation + Object Tracking

D. Discussions

The YOLOv11's advancements synthesize cutting-edge architectural improvements and practical application scalability. Introducing the C3k2 and C2PSA blocks ensures enhanced accuracy and computational efficiency, making YOLOv11 a versatile model for diverse industries.

1) Scalability Across Environments:

- The availability of multiple model variants, from nano to xlarge, makes YOLOv11 suitable for both edge devices and high-performance systems. However, optimizing these variants for specific hardware configurations remains a key area for future research.

2) Comparison with EfficientDet and Mask R-CNN:

- Unlike EfficientDet, which heavily relies on compound scaling for balancing accuracy and efficiency, YOLOv11 achieves similar or better mAP

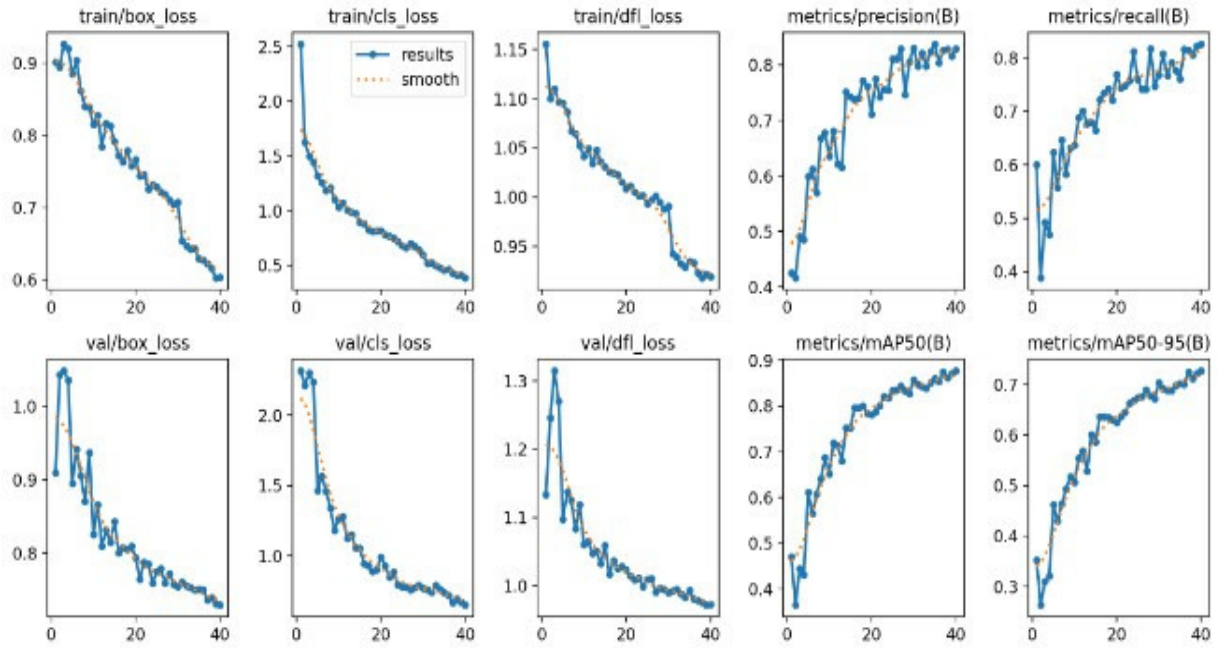

Fig.7. Results of the YOLOv11 model over various performance metrics

scores with lower computational costs due to its optimized architecture.
- Compared to Mask R-CNN, YOLOv11 offers faster inference times while maintaining competitive segmentation accuracy, making it more suitable for real-time applications.

3) Adaptability to Emerging CV Tasks:
- YOLOv11's support for instance segmentation, pose estimation, and oriented bounding box detection positions it as a comprehensive tool for emerging CV challenges. Its modular design facilitates customization for domain-specific applications.

4) Potential Challenges:
- While the model achieves state-of-the-art results, its reliance on advanced hardware for training may limit accessibility for smaller organizations. Efforts to streamline training pipelines and reduce dependency on GPUs could democratize access to YOLOv11's capabilities.

5) Future Directions:
- Enhancing model interpretability and incorporating self-supervised learning techniques could further elevate YOLOv11's utility. Additionally, expanding its compatibility with diverse datasets, including those with less structured annotations, could broaden its adoption.

E. Practical Applications

1) Autonomous Vehicles:
- YOLOv11's ability to process video streams in real-time enables accurate detection of pedestrians, vehicles, and traffic signs, ensuring safety and efficiency.

2) Medical Imaging:
- The model's high precision in segmenting organs and tumors is validated on custom datasets, demonstrating potential for diagnostic and surgical applications.

3) Retail Analytics:
- YOLOv11 tracks customer movements and accurately identifies products, improving inventory management and customer experience.

V. Conclusion and Future Work

In this paper, The AE-YOLOv11 is introduced with significant advancements, particularly through its architectural innovations (C3k2 and C2PSA blocks), enhancing accuracy and computational efficiency. The availability of various model variants (e.g., nano, xlarge) allows YOLOv11 to cater to diverse use cases, from edge devices to high-performance systems. This adaptability makes it versatile across industries. The AE-YOLOv11 outperforms competing models like EfficientDet, Mask R-CNN, SSD, and Faster R-CNN in terms of real-time detection capabilities, balancing speed and accuracy effectively. The model is suitable for diverse industries, including autonomous systems (e.g., vehicle detection), healthcare (e.g., tumor segmentation), and retail analytics (e.g., customer tracking). Focus areas include optimizing deployment on resource-constrained devices, enhancing model interpretability, incorporating self-supervised learning techniques, and broadening compatibility with less structured datasets. Future research should focus on optimizing deployment on resource-constrained devices, improving interpretability, and expanding its applicability across domains. The AE-YOLOv11's advancements pave the way for innovation in industries ranging from autonomous systems to healthcare, underscoring its position as a leader in computer vision technology.

References

[1] Redmon, J., et al. "You Only Look Once: Unified, Real-Time Object Detection." CVPR, 2016.

[2] Bochkovskiy, A., et al. "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv, 2020.

[3] Sivadi Balakrishna and Vijender Kumar Solanki "RTPD-YOLO: Reconciling YoLo-V8 Model for Real-Time Potholes Detection", in International Conference on Machine learning and Applied Network Technologies (ICMLANT 2024) is organized by IEEE El Salvador Section, pp. 1-6, Dec 13-14, 2024.

[4] Sivadi Balakrishna **"**D-ACSM: a technique for dynamically assigning and adjusting cluster patterns for IoT data analysis", *The Journal of Supercomputing*, Springer, ISSN 1319-1578, Vol. 78, Issue 10, pp, 12873-12897, Mach 2022. DOI: https://doi.org/10.1007/s11227-022-04427-1

[5] Balakrishna, Sivadi, and Ahmad Abubakar Mustapha. "Progress in multi-object detection models: a comprehensive survey." *Multimedia Tools and Applications* 82, no. 15 (2023): 22405-22439.

[6] Balakrishna, Sivadi, Yerrakula Gopi, and Vijender Kumar Solanki. "Comparative analysis on deep neural network models for detection of cyberbullying on Social Media." *Ingeniería Solidaria* 18, no. 1 (2022): 1-33.

[7] Balakrishna, Sivadi, Moorthy Thirumaran, and Vijender Solanki. "Machine Learning based Improved Gaussian Mixture Model for IoT Real-Time: Data Analysis." *Ingeniería Solidaria* 16, no. 1 (2020): 1-30.

[8] Suvarna, Buradagunta, and Sivadi Balakrishna. "Enhanced content-based fashion recommendation system through deep ensemble classifier with transfer learning." *Fashion and Textiles* 11, no. 1 (2024): 24.

[9] Balakrishna, Sivadi, M. Thirumaran, R. Padmanaban, and Vijender Kumar Solanki. "An efficient incremental clustering based improved K-Medoids for IoT multivariate data cluster analysis." *Peer-to-Peer Networking and Applications* 13, no. 4 (2020): 1152-1175.

[10] Balakrishna, Sivadi, Vijender Kumar Solanki, and Rubén González Crespo. "Generative AI for Smart Data Analytics." In *Generative AI: Current Trends and Applications*, pp. 67-85. Singapore: Springer Nature Singapore, 2024.

[11] Muhammad Hussain. Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. Machines, 11(7):677, 2023.

[12] Joseph Redmon and Ali Farhadi.Yolo9000: better, faster, stronger, "In Proceedings of the IEEE conference on computer vision and pattern recognition", pages 7263–7271, 2017.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[14] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[15] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao.Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.

[16] Ultralytics. YOLOv5: A state-of-the-art real-time object detection system. https://docs.ultralytics.com, 2021.

[17] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976, 2022.

[18] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao.Yolov7: Trainable bag-of-freebies sets new state-of-the-art for

real-time object detectors.In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 7464–7475, 2023.

[19] Mupparaju Sohan, Thotakura Sai Ram, Rami Reddy, and Ch Venkata.A review on yolov8 and its advancements.In International Conference on Data Intelligence and Cognitive Informatics, pages 529–545. Springer, 2024.

[20] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao.Yolov9: Learning what you want to learn using programmable gradient information.arXiv preprint arXiv:2402.13616, 2024.

[21] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding.Yolov10: Real-time end-to-end object detection. arXiv preprint arXiv:2405.14458, 2024.

[22] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024.