Performance Benchmarking of YOLOv11 Variants for Real-Time Delivery Vehicle Detection: A Study on Accuracy, Speed, and Computational Trade-offs

Original Research Article
.

ABSTRACT

The YOLOv series represents state-of-the-art technology for single-stage object detection, excelling in speed and accuracy. In many scenarios, it outperforms traditional two-stage detection frameworks, making it ideal for real-time applications. This study evaluates YOLOv11 model variants (n, s, m, i, x) on a custom dataset of 2,285 labelled images representing four delivery vehicle classes: FedEx, Other-Vehicles, UPS, and USPS-Truck. The dataset is meticulously curated to capture diverse delivery vehicle scenarios and split into training, validation, and test sets. Each variant was fine-tuned using uniform settings: 20 epochs, an input resolution of 640×640 pixels, and a batch size of 16.

Performance was assessed using metrics such as mean Average Precision (mAP, a standard metric measuring detection accuracy) across Intersection over Union (IoU) thresholds from 50% to 95% (a range defining the overlap between predicted and ground-truth bounding boxes), precision, recall, and inference speed on GPU and CPU. The results highlight trade-offs between model complexity and performance: smaller variants like YOLOv11-n achieved faster inference speeds (170.74 FPS on GPU and 5.86 ms on GPU), while larger models like YOLOv11-x excelled in detection accuracy and recall but at the cost of slower speeds (240.03 FPS on GPU and 4.17 ms on GPU). YOLOv11-s, for example, offered a balance with the highest FPS (1120.46 GPU FPS) but with moderate accuracy and recall. These findings demonstrate the adaptability of YOLOv11 variants to varying application requirements, from high-speed real-time systems to scenarios prioritizing detection accuracy.

This research advances object detection by providing a detailed performance benchmark for YOLOv11 variants. It offers practical insights for deploying YOLOv11 in diverse fields, including logistics, delivery tracking, and other domains requiring efficient and accurate object detection.

Keywords: YOLO, YoloV11, Object Detection Models, Deep Learning Computer Vision, Neural Networks, Dataset Evaluation, Model Performance Metrics, Image Processing Real-Time Applications

INTRODUCTION

Object detection is a critical task in computer vision, enabling machines to identify and localize objects in an image or video. In recent years, the YOLO (You Only Look Once) series has emerged as a cornerstone for real-time object detection, demonstrating state-of-the-art performance in various domains. The YOLO (You Only Look Once) series revolutionized object detection by reimagining it as a single regression problem. Traditional systems rely on complex pipelines, including region proposals, classification, and post-processing, which are computationally intensive and challenging to optimize. In contrast, YOLO uses a unified convolutional network to predict bounding boxes and class probabilities directly. This design enables real-time performance. YOLO's simplicity and efficiency have made it a benchmark in object detection. Each iteration improves speed, accuracy, and scalability. It trains on full images, optimizing detection performance end-to-end. YOLO has been widely applied, including in autonomous systems and robotics, where fast and accurate object detection is critical.

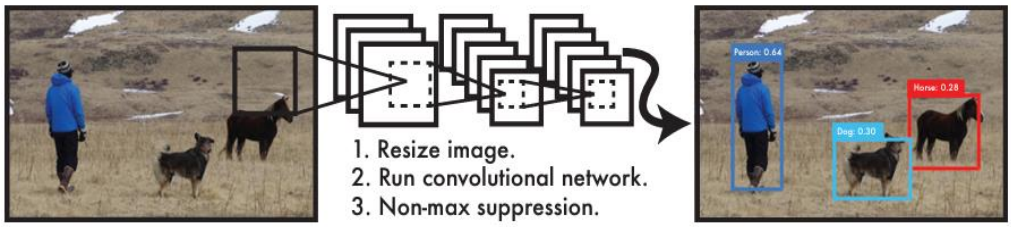

Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

This study investigates the latest iteration, YOLOv11, and its variants for object detection in the logistics industry, specifically for identifying delivery vehicles such as FedEx, UPS, USPS trucks, and other vehicles. By benchmarking YOLOv11 variants on a custom logistics dataset, we aim to provide insights into their suitability for domain-specific applications.

1.1 The YOLO Series: A Paradigm Shift in Object Detection

The YOLO series has transformed object detection with its single-stage detection pipeline, emphasizing speed without compromising accuracy. Unlike traditional two-stage approaches like Faster R-CNN, which separate object proposal generation and classification, YOLO combines these processes into a unified architecture, enabling real-time performance.

Key advancements through the YOLO series:

- YOLOv1: Introduced in 2015, YOLOv1 pioneered the single-shot detection framework, dividing the input image into a grid and predicting bounding boxes and class probabilities simultaneously.
- YOLOv2 and YOLOv3: Focused on improving accuracy with multi-scale detection, batch normalization, and feature pyramid networks.
- YOLOv4 and YOLOv5: Integrated advanced features like Cross Stage Partial (CSP) networks and Mosaic augmentation, offering a balance between speed and precision.
- YOLOv7 and YOLOv9: Pushed the boundaries with computational efficiency and transformer-based architecture, cementing YOLO's position as a leader in real-time object detection.
- YOLOv11, the latest iteration, builds on this legacy, incorporating cutting-edge innovations in attention mechanisms, anchor-free detection, and transformer modules.

Release	Year	Tasks	Contributions	Framework
YOLO	2015	Object Detection, Basic Classification	Single-stage object detector	Darknet
YOLOv2	2016	Object Detection, Improved Classification	Multi-scale training, dimension clustering	Darknet

YOLOv3	2018	Object Detection, Multi-scale Detection	SPP block, Darknet-53 backbone	Darknet
YOLOv4	2020	Object Detection, Basic Object Tracking	Mish activation, CSPDarknet-53 backbone	Darknet
YOLOv5	2020	Object Detection, Basic Instance Segmentation (via custom modifications)	Anchor-free detection, SWISH activation, PANet	PyTorch
YOLOv6	2022	Object Detection, Instance Segmentation	Self-attention, anchor-free OD	PyTorch
YOLOv7	2022	Object Detection, Object Tracking, Instance Segmentation	Transformers, E-ELAN reparameterisation	PyTorch
YOLOv8	2023	Object Detection, Instance Segmentation, Panoptic Segmentation, Keypoint Estimation	GANs, anchor-free detection	PyTorch
YOLOv9	2024	Object Detection, Instance Segmentation	PGI and GELAN	PyTorch
YOLOv10	2024	Object Detection	Consistent dual assignments for NMS-free training	PyTorch

Table 1: Yolov series models with their release date and importance

The above illustrates the progression of YOLO models, from their inception to the most recent versions. Each iteration has brought significant improvements in object detection capabilities, computational efficiency, and versatility for handling various computer vision tasks. This evolution highlights the rapid advancements in object detection technologies, with each version introducing innovative features and expanding the range of supported tasks. From YOLO's ground-breaking single-stage detection to YOLOv10's NMS-free training, the series has continuously pushed the boundaries of real-time object detection. The latest version, YOLOv11, builds on this legacy, further enhancing feature extraction, efficiency, and multi-task capabilities. In the following analysis, we will examine YOLOv11's architectural innovations, including its improved backbone and neck structures, and assess its performance across tasks such as object detection, instance segmentation, and pose estimation.

1.2 YOLOv11 : State-of-the-Art Real-Time Object Detection

YOLOv11 represents a leap forward in object detection, combining architectural innovations to optimize both accuracy and efficiency. With its modular design, YOLOv11 offers a suite of model variants tailored to diverse computational environments and application needs. YOLOv11 distinguishes itself through its enhanced adaptability, supporting an expanded range of CV tasks beyond traditional object detection. Notable among these are posture estimation and instance segmentation, broadening the model's applicability in various domains. YOLOv11's design focuses on balancing power and practicality, aiming to address specific challenges across various industries with increased accuracy and efficiency.

Key Features of YOLOv11:

- Dynamic Convolution: Enhances feature representation by adapting convolutional kernels based on spatial and contextual cues.
- Anchor-Free Mechanism: Eliminates the need for predefined anchor boxes, simplifying model training and improving performance on irregular objects.
- Efficient Attention Mechanisms: Integrates lightweight attention layers to focus on critical regions of an image while reducing computational overhead.
- Improved Training Strategies: Incorporates advanced augmentation techniques and loss functions for better generalization.

Variants of YOLOv11:

- YOLOv11-n (Nano)
    - Ultra-lightweight model designed for low-power devices such as edge sensors and IoT applications.
    - Prioritizes speed over accuracy, making it suitable for environments with limited computational resources.
- YOLOv11-s (Small)

- Offers a balance between speed and accuracy, ideal for medium-scale applications like real-time drone surveillance.
- YOLOv11-m (Medium)
  - General-purpose model that provides a trade-off between precision and computational efficiency.
  - Suitable for scenarios requiring moderate accuracy and resource usage.
- YOLOv11-i (Intermediate)
  - Focuses on higher accuracy for detecting complex and smaller objects.
  - Suitable for use cases with moderate computational resources, such as on-premises GPUs.
- YOLOv11-x (Extreme)
  - The largest and most precise variant, designed for high-performance systems with ample computational power.
  - Excels in tasks requiring maximum accuracy, such as autonomous driving and high-resolution video analysis.

1.3 The YOLOv11 architecture

The YOLO framework revolutionized object detection by introducing a unified neural network that simultaneously handles both bounding box regression and object classification. This integrated approach represented a major shift from traditional two-stage detection methods, providing end-to-end training capabilities through its fully differentiable design. The YOLO architecture consists of three key components. First, the backbone, which extracts features from the raw image data using convolutional neural networks to create multi-scale feature maps. Next, the neck component, which aggregates and refines feature representations across different scales. Finally, the head component, which generates the final outputs for object localization and classification based on the processed feature maps. The basic framework architecture is illustrated in the below Figure 2.

Figure 2: Key architectural modules in YOLO11

1.4 Relevance of Object Detection in Logistics

The logistics industry relies heavily on accurate and efficient object detection systems to optimize operations. Applications such as package tracking, delivery vehicle identification, and real-time inventory monitoring are integral to maintaining streamlined supply chains. Detecting delivery vehicles, including FedEx, UPS, and USPS trucks, is particularly crucial for ensuring timely deliveries and operational efficiency.

Despite advancements in object detection, domain-specific challenges persist:

- Class Imbalance: Logistics datasets often contain skewed distributions of object classes, such as fewer examples of specific vehicle types.
- Real-Time Constraints: The need for rapid processing in dynamic environments like warehouses and delivery hubs.
- Variability in Appearance: Different lighting conditions, occlusions, and viewing angles can impact detection performance.

This study evaluates YOLOv11 variants on a custom logistics dataset to address these challenges, benchmarking their performance across key metrics, including mAP (50-95), precision, recall, and inference speed. The findings aim to guide practitioners in selecting the most suitable YOLOv11 variant for logistics-specific applications.

2. METHODOLOGY

In the section we discuss details of the dataset and models we have examined for the research.

2.1 Dataset Overview

For this study, we utilized the USPS-Merge dataset, which was sourced from the Roboflow Universe platform. This dataset was specifically curated for object detection tasks and contains images of vehicles associated with various logistics and postal services. The dataset includes four distinct vehicle classes, and it is divided into three main subsets:

- Training Set: The training subset contains 1637 labelled images, used to train the object detection models. These images contain vehicles belonging to the specified classes and are annotated with bounding boxes. These labels enable the models to learn the spatial location of each object in the image.
- Validation Set: The validation subset contains 443 images, used during the training process for model evaluation and hyperparameter tuning. The validation set helps in adjusting the model parameters to avoid overfitting and to select the best-performing model during training.
- Test Set: The test subset consists of 205 images. This dataset is kept separate and is used for final model evaluation and inference, ensuring that the evaluation results are unbiased and reflective of the model's real-world performance.

The dataset comprises the following four classes:

- USPS-Truck: This class includes images of vehicles used by the United States Postal Service (USPS), often characterized by a white and blue color scheme. These vehicles are crucial in the context of postal delivery and provide a diverse dataset for training the model to detect different logistics vehicles.
- UPS: This class consists of images of UPS (United Parcel Service) delivery trucks. These vehicles are typically brown and are used for parcel delivery in various regions globally. They are another important class to distinguish in logistics-based object detection tasks.
- Other-Vehicles: This class encompasses a variety of vehicles not related to specific postal or delivery services, such as personal cars, trucks, buses, and motorcycles. It serves to introduce diversity and background vehicles in the dataset, making the task more challenging for the object detection models.
- FedEx: This class includes images of FedEx delivery trucks, commonly used in logistics for parcel delivery. FedEx vehicles are often characterized by their unique color schemes, including purple and orange branding.

Figure 3: Sample image of dataset: USPS-Truck, UPS, Other-Vehicles, FedEx (from left to right)

2.2 Training Configuration

To ensure a consistent evaluation of different variants of the YOLOv11 architecture, we specified the following uniform configuration parameters for all models. The training configuration is designed to balance computational efficiency, model performance, and overfitting mitigation.

- Epochs: The model was trained for 20 epochs, which were chosen based on empirical results from previous studies and the model's convergence behaviour during preliminary experiments. The relatively low number of epochs was adequate for fine-tuning the model on the USPS-Merge dataset, given the dataset's size.
- Image Size: The images were resized to a consistent resolution of 640 x 640 pixels for both training and validation. This image size strikes a balance between maintaining sufficient detail for the object detection task and ensuring the model can be trained efficiently without excessive computational requirements. The choice of image size is consistent with typical YOLO configurations, which can process images at this resolution while maintaining real-time detection performance.
- Batch Size: A batch size of 16 was selected as it is a common default for object detection tasks, offering a balance between memory usage and gradient stability. For YOLOv11-x, a batch size of 8 was employed to accommodate computational cost constraints. While smaller batch sizes can result in noisier gradients, larger batch sizes may significantly increase memory consumption without providing proportional improvements in model performance.
- Optimizer: The AdamW optimizer was selected for training. AdamW is an extension of the Adam optimizer that includes weight decay, which helps regularize the model and prevents overfitting by penalizing large weight values. This choice was made to improve convergence speed and model generalization. The learning rate was initially set to 0.00125, with a momentum parameter of 0.9. The optimizer automatically adjusted the learning rate during training, based on the loss function's gradient.

- Learning Rate: The learning rate was automatically adjusted during training using the AdamW optimizer. This dynamic adjustment helps the model converge faster while maintaining stability. The initial learning rate of 0.00125 was chosen based on preliminary experiments and literature recommendations for similar tasks.
- Uniform Configuration Across Models: All variants of YOLOv11 (n, s, m, i, x) were trained with identical hyperparameters to ensure that any performance differences observed during evaluation were due to architectural variations rather than changes in training configuration. This uniform approach allows for a fair comparison of the models' performance across different scales and capabilities.

2.3 Data Augmentation

To enhance the model's generalization capabilities and reduce overfitting, we employed a set of data augmentation techniques during training. These augmentations were applied randomly to the training images, introducing variability to the dataset and helping the model learn robust features. The following augmentations were applied:

- Blur: This augmentation applied slight Gaussian blur to the images with a probability of 0.01. This helps the model to become more robust to motion blur or low-quality images during inference.
- MedianBlur: Similar to Gaussian blur but using median blurring with a probability of 0.01. This transformation can help the model deal with noise or sharp edges in images.
- ToGray: This transformation converts the images to grayscale with a probability of 0.01, allowing the model to focus on spatial features rather than color, which can be useful in scenarios where color information is less important or unreliable.
- CLAHE (Contrast Limited Adaptive Histogram Equalization): This augmentation improves the contrast of the image with a probability of 0.01, helping the model focus on important features by making the images more visually distinguishable.

These augmentations were chosen to simulate common variations encountered in real-world object detection tasks, where lighting conditions, image quality, and environmental factors can affect the visibility and appearance of objects.

2.4 Evaluation Metrics

To assess the performance of the trained models, several key evaluation metrics were used, including those specifically designed for object detection tasks. These metrics are essential for understanding how well each model variant performs in real-world conditions, balancing both accuracy and efficiency.

- mAP (Mean Average Precision at IoU 50-95): Mean Average Precision (mAP) is one of the most widely used metrics for evaluating object detection performance. It calculates the average precision at multiple Intersection over Union (IoU) thresholds, ranging from 0.5 to 0.95. This provides a comprehensive view of model performance across various levels of localization accuracy. A higher mAP score indicates better performance in both detecting objects and accurately localizing them.
- Precision: Precision measures the accuracy of the positive predictions made by the model, defined as the ratio of true positives (TP) to the sum of true positives and false positives (FP). High precision ensures that the model does not misidentify other objects as part of the target classes. Precision is crucial when false positives could lead to unnecessary alerts or actions, as seen in autonomous driving or security applications.
- Recall: Recall measures how well the model can detect all relevant objects, defined as the ratio of true positives to the sum of true positives and false negatives (FN). High recall ensures that the model does not miss any objects in the image. It is particularly important in applications where missing an object could lead to critical failures, such as detecting delivery vehicles in real-time.
- Inference Speed (CPU ONNX - ms): Inference speed is measured in milliseconds per image and is crucial for real-time applications. Faster inference speeds are necessary for use cases where the model must operate within time constraints, such as autonomous vehicles or surveillance systems. The inference speed was measured using ONNX (Open Neural Network Exchange) format, which is optimized for fast deployment across various hardware platforms.
- Confusion Matrix: The Confusion Matrix was used to evaluate the classification performance of the models in detail. It provides a summary of the true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for each class. This allows for a deeper understanding of where the model is making errors, such as misclassifying UPS trucks as FedEx or missing objects from the USPS-Truck class. From the confusion matrix, several additional metrics can be derived, including:
    - F1-Score: The harmonic mean of precision and recall, offering a balanced measure of the model's performance.
    - Accuracy: The overall proportion of correct predictions made by the model across all classes.
    - Specificity: The ability of the model to correctly identify non-object regions, ensuring that the model does not make excessive false positive predictions.

2.5 Training Process Overview
The training process was carried out using TensorBoard for real-time monitoring of key performance metrics such as loss, mAP, and precision-recall curves. TensorBoard also provided insights into the model's architecture, allowing us to visualize the training process and adjust hyperparameters as needed. Additionally, Automatic Mixed Precision (AMP) was used during training, which enables faster training by using lower precision for certain operations without sacrificing model accuracy. This results in reduced memory usage and improved training throughput, especially on GPUs.
By following this methodology, we ensured that the results of the study were robust and comparable across all YOLOv11 model variants. The next phase involves the detailed evaluation and comparison of the model variants based on the performance metrics outlined above.

3. EVALUATION

In this section we have discussed outcomes of different variants of yolov11 models and compared performance matrices.

Yolov11-n-model
The evaluation result shows both training and validation losses show consistent reduction, indicating better model convergence and performance metrics such as mAP, precision, and recall exhibit consistent growth, highlighting robust improvement in detection and classification capabilities over the 20 epochs as shown below figure 4.

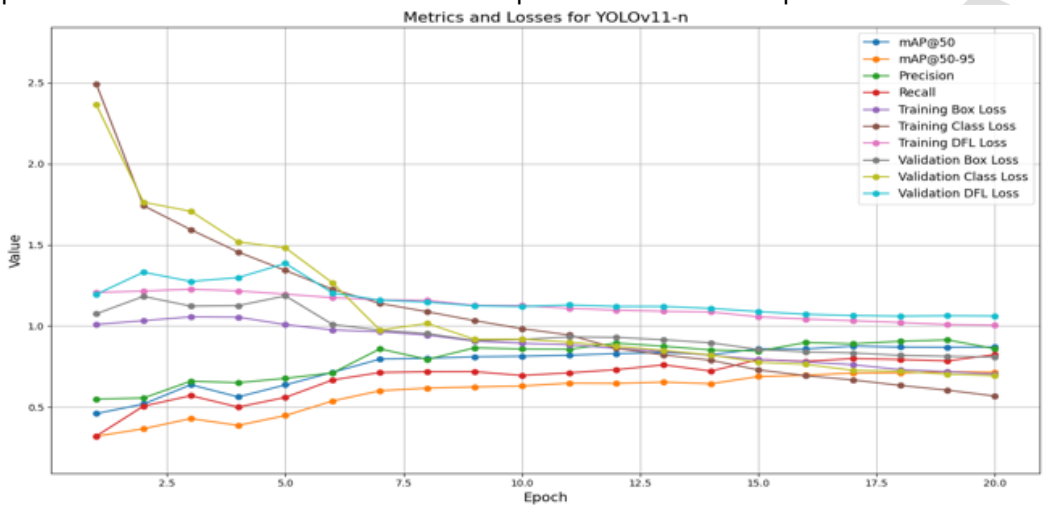

Figure 4: Yolov11-n training result.

Loss Metrics
Training Loss:

- Box Loss: Gradual decline from 1.01035 (Epoch 1) to 0.70374 (Epoch 20), indicating improved bounding box localization.
- Class Loss: Significant reduction from 2.49392 (Epoch 1) to 0.56846 (Epoch 20), reflecting enhanced object classification accuracy.
- DFL Loss: Slight reduction from 1.20724 (Epoch 1) to 1.00522 (Epoch 20), suggesting better consistency in distribution focal loss.

Validation Loss:

- Box Loss: Decreases from 1.07546 (Epoch 1) to 0.80903 (Epoch 20), aligning with training loss improvements.
- Class Loss: Drops notably from 2.36577 (Epoch 1) to 0.69375 (Epoch 20), showcasing better generalization.
- DFL Loss: Gradual reduction from 1.19559 (Epoch 1) to 1.06223 (Epoch 20). Performance Metrics mAP (Mean Average Precision):

Performance Metrics:

- mAP50: Significant improvement from 0.46003 (Epoch 1) to 0.87071 (Epoch 20), demonstrating better detection at 50% IoU threshold.
- mAP50-95: Steady increase from 0.32051 (Epoch 1) to 0.71511 (Epoch 20), reflecting enhanced detection across IoU thresholds.
- Precision: Improves from 0.54975 (Epoch 1) to 0.86050 (Epoch 20), indicating fewer false positives over time.
- Recall: Gains from 0.32155 (Epoch 1) to 0.82393 (Epoch 20), reflecting better object detection coverage across images.

Yolov11-s model

The model exhibits steady improvements across all metrics and losses, indicating stable training with a well-behaved loss curve. Both training and validation losses show a consistent downward trend, and the performance metrics (Precision, Recall, mAP) show a strong upward trend, suggesting that the model is progressively improving both in fitting the data and generalizing well on unseen data as shown in below figure 5.

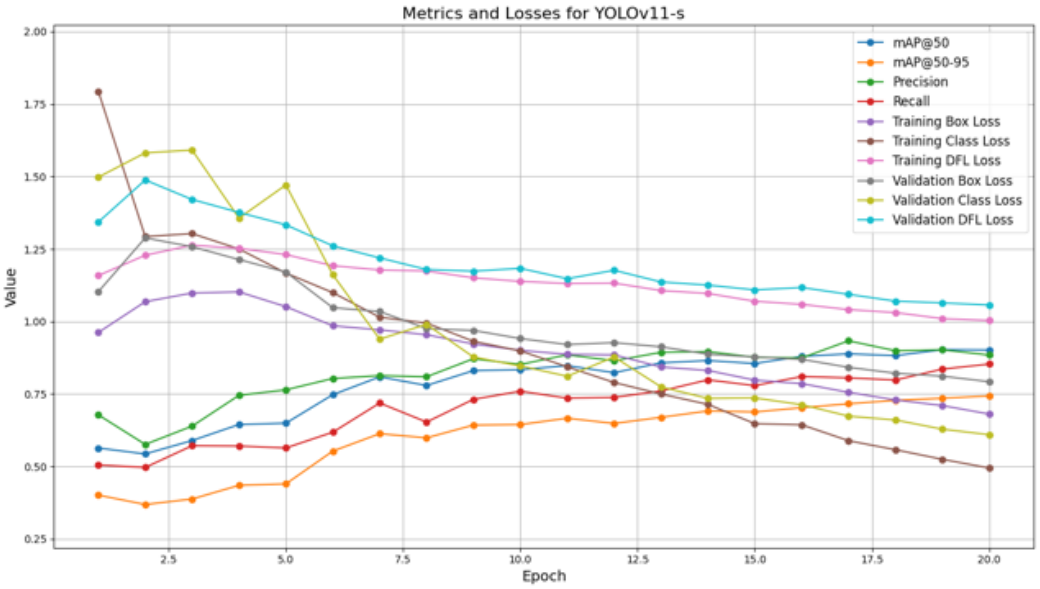

Figure 5: Yolov11-s training result.

Loss Metrics

Training Loss:

- Box Loss: The training box loss steadily decreases from 0.9623 in epoch 1 to 0.6809 in epoch 20, showing a consistent improvement in the model's ability to fit the bounding boxes.
- Class Loss: The training class loss decreases from 1.7943 in epoch 1 to 0.4948 in epoch 20, indicating better classification performance as training progresses.
- DFL Loss: The training DFL (distribution focal loss) also shows a decreasing trend, from 1.1586 in epoch 1 to 1.0037 in epoch 20, suggesting the model is improving its decision-making across epochs.

Validation Loss:

- Box Loss: The validation box loss decreases from 1.1029 in epoch 1 to 0.6097 in epoch 20, confirming that the model's generalization ability is improving over time.
- Class Loss: The validation class loss decreases from 1.4981 in epoch 1 to 1.0571 in epoch 20, showing progress in classification performance.
- DFL Loss: The validation DFL loss decreases from 1.3432 in epoch 1 to 1.0644 in epoch 20, indicating improvements in handling difficult examples.

Performance Metrics:

- mAP@50: The mAP at 50% intersection-over-union (IoU) improves from 0.5637 in epoch 1 to 0.9020 in epoch 20, indicating strong improvement in detection performance.
- mAP@50-95: This metric shows a steady rise from 0.4013 in epoch 1 to 0.7435 in epoch 20, reflecting better performance at multiple IoU thresholds.
- Precision: The precision increases from 0.6781 in epoch 1 to 0.8845 in epoch 20, reflecting an improvement in the model's ability to correctly identify positive samples.
- Recall: Recall shows a similar trend, rising from 0.5051 in epoch 1 to 0.8534 in epoch 20, which suggests better recall as training progresses.

Yolov11-m model

The training of the "m" variant of the yolov11 model exhibits both the training and validation losses showing a steady decrease, which is a sign of stable learning and effective generalization. Key performance metrics like Precision, recall, and mAP metrics all show strong improvement across the epochs, suggesting that the model is not only learning but also generalizing well to unseen data as shown in below figure 6.

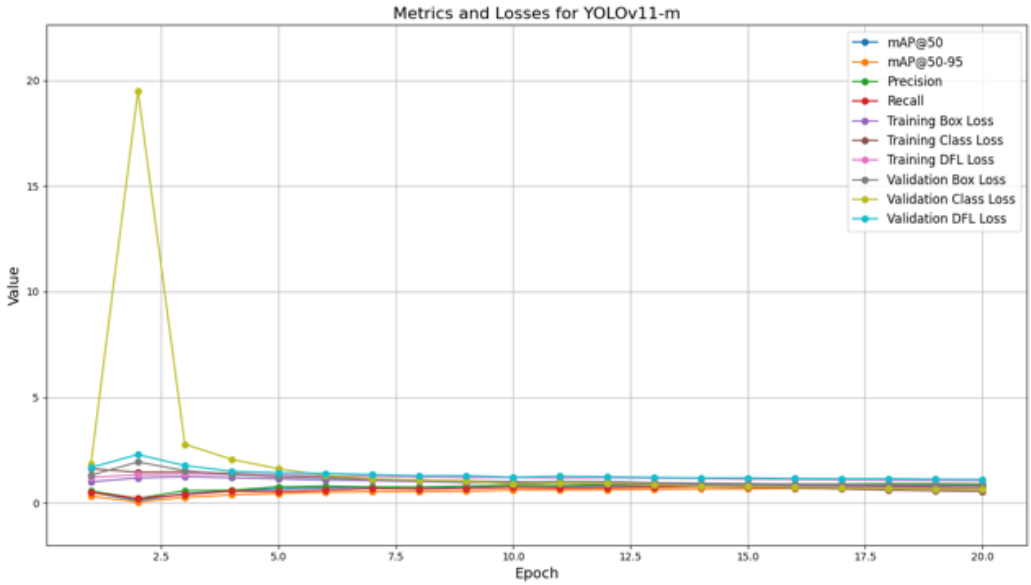

Figure 6: Yolov11-m training result.

Loss Metrics

Training Loss:

- Box Loss: The training box loss decreases steadily from 1.0043 in epoch 1 to 0.7271 in epoch 20, suggesting that the model is learning to improve its bounding box predictions over time.
- Class Loss: The training class loss shows a consistent reduction from 1.6252 in epoch 1 to 0.5540 in epoch 20, indicating improvements in the model's classification accuracy.
- DFL Loss: The DFL loss reduces from 1.2131 in epoch 1 to 1.0549 in epoch 20, reflecting gradual refinement in the model's distribution-based predictions.

Validation Loss:

- Box Loss: The validation box loss decreases from 1.3202 in epoch 1 to 0.8132 in epoch 20, indicating improved generalization as the model trains.
- Class Loss: Validation class loss declines from 1.8594 in epoch 1 to 0.6478 in epoch 20, showing the model is better at classifying objects in unseen data.
- DFL Loss: Validation DFL loss also drops from 1.6829 in epoch 1 to 1.1142 in epoch 20, suggesting that the model is successfully learning to predict object distributions more effectively.

Performance Metrics:

- mAP50 (B): The mAP at 50% IoU increases from 0.5143 in epoch 1 to 0.8919 in epoch 20, reflecting better detection accuracy.
- mAP50-95 (B): mAP at 50-95% IoU improves from 0.3234 in epoch 1 to 0.7288 in epoch 20, indicating the model's improved performance across a range of IoU thresholds.
- Precision (B): Precision rises significantly from 0.5679 in epoch 1 to 0.8972 in epoch 20, indicating that the model is getting much better at correctly identifying objects.
- Recall (B): Recall also improves from 0.4982 in epoch 1 to 0.8272 in epoch 20, suggesting a reduction in false negatives.

Yolov11-i model

Over the 20 epochs, the model demonstrates steady improvement in all critical areas: Losses: Both training and validation losses consistently decrease, indicating that the model is effectively learning and generalizing. Metrics: mAP, precision, and recall all show substantial gains, suggesting the model's increasing accuracy in detecting and classifying objects. Learning Rate: The gradual decrease in learning rate suggests the model is stabilizing its learning process, allowing for more refined training as it progresses as shown in below figure 7.

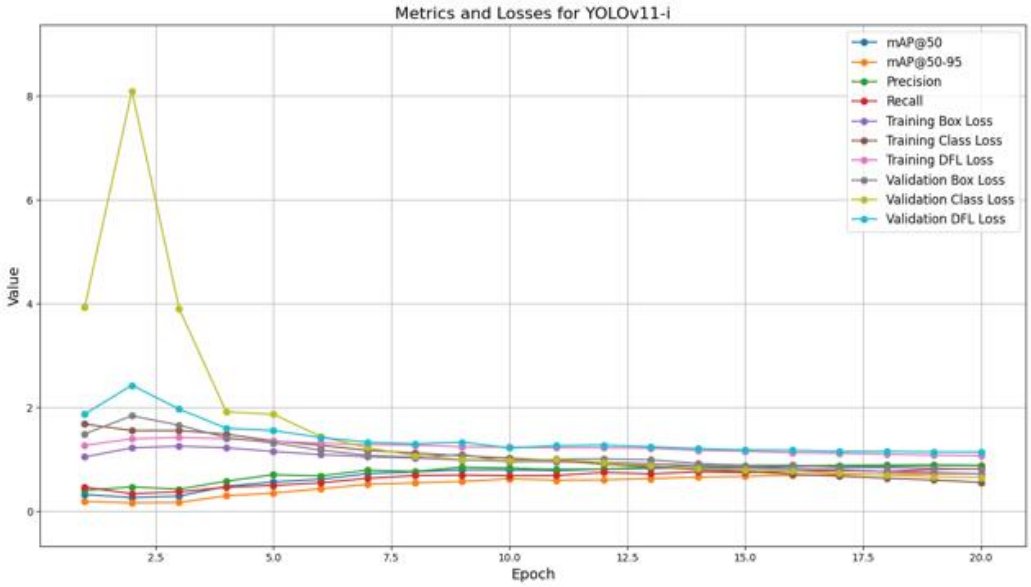

Figure 7: Yolov11-i training result.

Loss Metrics

Training Loss:

- Box Loss: The training box loss decreases steadily from 1.0498 in epoch 1 to 0.7242 in epoch 20, indicating that the model is progressively refining its ability to predict bounding boxes accurately.
- Class Loss: The class loss also shows a consistent reduction from 1.68318 in epoch 1 to 0.5594 in epoch 20, reflecting improved classification accuracy as the model trains.
- DFL Loss: The DFL loss decreases from 1.2726 in epoch 1 to 1.07337 in epoch 20, suggesting ongoing improvements in the model's ability to predict the distribution of object centres.

Validation Loss:

- Box Loss: The validation box loss decreases from 1.49338 in epoch 1 to 0.8147 in epoch 20, indicating the model is improving in generalization, with better bounding box predictions on unseen data.
- Class Loss: Validation class loss reduces from 3.93951 in epoch 1 to 0.6588 in epoch 20, showing the model's enhanced ability to classify objects correctly in validation samples.
- DFL Loss: The validation DFL loss drops from 1.87535 in epoch 1 to 1.14352 in epoch 20, suggesting that the model's distribution prediction capability is improving as it trains on more data.

Performance Metrics:

- mAP50 (B): The mAP at 50% IoU improves from 0.32562 in epoch 1 to 0.8846 in epoch 20, reflecting significant improvements in detection accuracy over time.
- mAP50-95 (B): The mAP at 50-95% IoU rises from 0.18964 in epoch 1 to 0.72672 in epoch 20, showing that the model is becoming more robust in detecting objects across varying levels of IoU.
- Precision (B): Precision increases from 0.40676 in epoch 1 to 0.88615 in epoch 20, indicating that the model is getting much better at identifying objects correctly and reducing false positives.
- Recall (B): Recall improves from 0.46699 in epoch 1 to 0.82067 in epoch 20, showing a reduction in false negatives and an increase in the model's ability to detect all relevant objects.

Yolov11-x model

Throughout the 20 epochs, the model demonstrates significant improvement in all key metrics. Both training and validation losses consistently decrease, indicating effective learning and generalization. mAP, precision, and recall show substantial gains, reflecting the model's increasing accuracy and ability to detect and classify objects. Additionally, the gradual decrease in the learning rate suggests that the model is nearing convergence, making finer adjustments as it progresses as shown in below figure 8.

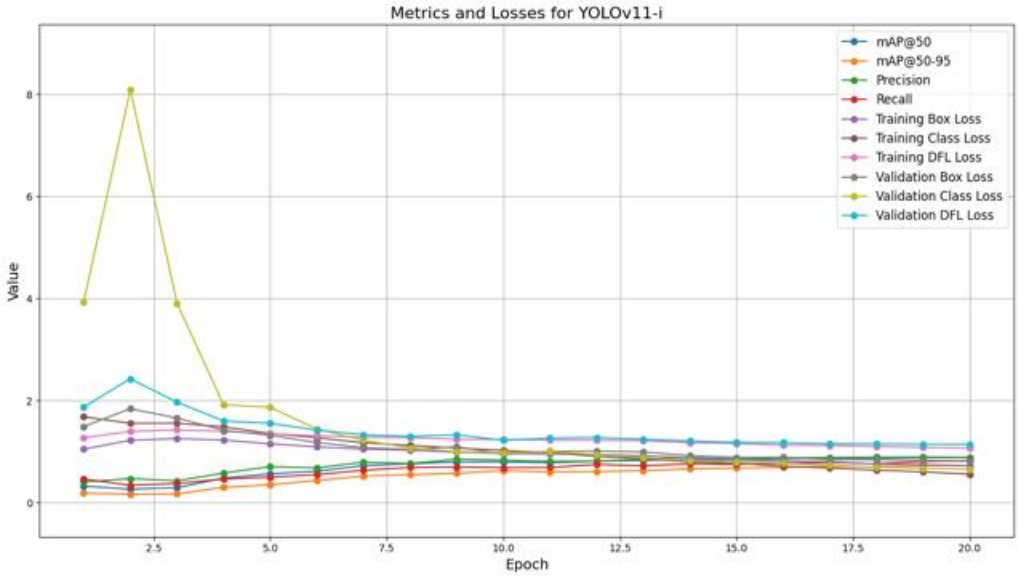

Figure 8: Yolov11-x training result.

Loss Metrics

Training Loss:

- Box Loss: The training box loss decreases steadily from 1.16451 in epoch 1 to 0.62205 in epoch 20, indicating continuous improvement in the model's ability to predict bounding boxes more accurately.
- Class Loss: The class loss shows a consistent reduction from 1.82984 in epoch 1 to 1.10992 in epoch 20, reflecting a significant improvement in classification accuracy over time.
- DFL Loss: The DFL loss decreases from 1.38319 in epoch 1 to 0.88079 in epoch 20, demonstrating the model's ongoing refinement in predicting distribution-based object centre predictions.

Validation Loss:

- Box Loss: The validation box loss decreases from 1.62893 in epoch 1 to 0.81976 in epoch 20, suggesting that the model is generalizing well and making better bounding box predictions on unseen data.
- Class Loss: Validation class loss decreases from 7.77179 in epoch 1 to 0.69336 in epoch 20, indicating that the model has improved in classifying objects correctly in the validation set.
- DFL Loss: The validation DFL loss drops from 2.1399 in epoch 1 to 1.1616 in epoch 20, reflecting improved predictions of object distributions in the validation phase.

Performance Metrics:

- mAP50 (B): The mAP at 50% IoU improves significantly from 0.22541 in epoch 1 to 0.88079 in epoch 20, highlighting a strong improvement in detection accuracy as the model trains.
- mAP50-95 (B): The mAP at 50-95% IoU shows substantial improvement from 0.37487 in epoch 1 to 0.87436 in epoch 20, demonstrating that the model is improving across different IoU thresholds, enhancing its detection capabilities.
- Precision (B): Precision increases from 0.24387 in epoch 1 to 0.70684 in epoch 20, suggesting a reduction in false positives and a significant improvement in correctly identifying objects.
- Recall (B): Recall improves from 0.12998 in epoch 1 to 0.70684 in epoch 20, indicating a reduction in false negatives and that the model is becoming better at detecting all relevant objects.

4. PREDICTION

The performance of different YOLOv11 model variants was evaluated on the USPS-Merge-38 dataset using a total of 205 test images. The evaluation focused on key metrics, including inference speed (in milliseconds for both GPU and CPU), model parameters (in millions), and FPS (Frames Per Second) for GPU execution.

The results are summarized as follows:

YOLOv11-n achieved the highest GPU FPS of 170.74, with a relatively low GPU inference time of 5.86 ms, though it comes with 2.59 million parameters. In contrast, its CPU inference time was significantly higher at 154.71 ms.

YOLOv11-s exhibited the fastest inference on GPU with 1120.46 FPS, but its CPU inference time was the highest among the tested models at 347.36 ms. It has 9.43 million parameters.

YOLOv11-m demonstrated a balanced performance, with 430.40 FPS on GPU and an inference time of 2.32 ms for GPU, along with a CPU inference time of 889.04 ms. It contains 20.06 million parameters.

YOLOv11-i showed an FPS of 304.91 on GPU, with a 3.28 ms GPU inference time. However, it faced a 1118.48 ms delay on CPU, and its model size was 25.31 million parameters.
YOLOv11-x had the largest model size with 56.88 million parameters and the lowest GPU FPS of 240.03. Its GPU inference time was 4.17 ms, and the CPU inference time was 2349.83 ms.

Custom Mode	Speed GPU (ms)	Speed CPU (ms)	Params (M)	GPU FPS
yolov11-n	5.86	154.71	2.59	170.74
yolov11-s	0.89	347.36	9.43	1120.46
yolov11-m	2.32	889.04	20.06	430.40
yolov11-i	3.28	1118.48	25.31	304.91
yolov11-x	4.17	2349.83	56.88	240.03

Table 2: Prediction results on custom models.

5. CONCLUSION

This research investigated the performance of various YOLOv11 model variants (n, s, m, i, x) for object detection on the USPS-Merge-38 dataset, which includes four distinct vehicle classes: FedEx, UPS, USPS-Truck, and Other-Vehicles. Our study aimed to evaluate the trade-offs between model complexity, accuracy, and inference speed in real-world logistics and delivery tracking scenarios.
The results highlight that smaller YOLOv11 variants, such as YOLOv11-n and YOLOv11-s, deliver superior inference speed, making them ideal candidates for real-time applications where speed is critical. On the other hand, larger variants, like YOLOv11-i and YOLOv11-x, offer improved accuracy and recall, making them suitable for tasks that prioritize detection quality over processing time. The findings underscore the adaptability of the YOLOv11 architecture, demonstrating that different model configurations can be selected depending on specific application requirements, balancing between computational resources and performance outcomes.
Furthermore, the model evaluation metrics, including mAP, precision, recall, and inference speed, provide a comprehensive understanding of the strengths and limitations of each variant. The performance of the models was rigorously evaluated across different hardware configurations, with GPU and CPU inference times measured for each model variant. These evaluations offer valuable insights into the practical deployment of YOLOv11 in real-world logistics systems, where both accuracy and efficiency are paramount.
In conclusion, the YOLOv11 family of models presents a versatile solution for object detection tasks, with each variant offering distinct advantages based on the desired balance between speed and accuracy. This research contributes to the growing body of knowledge on real-time object detection in logistics and delivery tracking, providing a benchmark for selecting the optimal YOLOv11 model based on specific operational requirements.

Disclaimer (Artificial intelligence)

Author(s) hereby declares that no generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript.

REFERENCES

Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi “You Only Look Once: Unified, Real-Time Object Detection”. Access mode: https://arxiv.org/abs/1506.02640.

Mujadded Al Rabbani Alif "YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain". Access mode: https://arxiv.org/html/2406.10139v1

Rahima Khanam, Muhammad Hussain “YOLOv11: An Overview of the Key Architectural Enhancements”. Access mode: https://arxiv.org/pdf/2410.17725

David Rimer “USPS Merge Computer Vision Project”. Access mode: https://universe.roboflow.com/david-rimer-3ch55/usps-merge

Rahima Khanam, Muhammad Hussain, Richard Hill, and Paul Allen “A comprehensive review of convolutional neural networks for defect detection in industrial applications” IEEE Access, 2024. Access mode: https://ieeexplore.ieee.org/document/10589380

S. Raghavan. “2020 AI predictions from IBM research”. Access mode: https://www.ibm.com/blogs/research/2019/12/2020-ai-predictions.

Joseph Redmon, Ali Farhadi, “YOLO9000: Better, Faster, Stronger”. Access mode: https://arxiv.org/pdf/1612.08242.

Joseph Redmon Ali Farhadi, “YOLOv3: An Incremental Improvement”. Access mode: https://arxiv.org/pdf/1804.02767.

Alexey Bochkovskiy,Chien-Yao Wang,Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection". Access mode: https://arxiv.org/pdf/2004.10934.

Trupti Mahendrakar, “Performance Study of YOLOv5 and Faster R-CNN for Autonomous Navigation around Non-Cooperative Targets”. Access mode: https://arxiv.org/pdf/2301.09056.

Chuyi Li, “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications”. Access mode: https://arxiv.org/pdf/2209.02976

Chien-Yao Wang, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. Access mode: https://arxiv.org/pdf/2207.02696.

Rejin Varghese, “YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness”. Access mode: https://ieeexplore.ieee.org/document/10533619.

Chien-Yao Wang, “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information”. Access mode: https://arxiv.org/pdf/2402.13616.

Ao Wang1, “YOLOv10: Real-Time End-to-End Object Detection”, Access mode: https://arxiv.org/pdf/2405.14458.

Juan Du, “Understanding of object detection based on cnn family and yolo”. Access mode: Conference Series, volume 1004, page 012029. IOP Publishing, 2018.

Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye, “Object detection in 20 years: A survey”. Proceedings of the IEEE, 111(3):257–276, 2023.

Momina Liaqat Ali, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection". Access mode: https://www.preprints.org/manuscript/202410.1785/v.

Muhammad Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection". Access mode: https://www.mdpi.com/2075-1702/11/7/677