J-INTECH (Journal of Information and Technology)
Accredited Sinta 4 Ministry of Higher Education, Science and Technology Republic of Indonesia SK No. 10/C/C3/DT.05.00/2025
E-ISSN: 2580-720X || P-ISSN: 2303-1425

Detection and Classification of Organic and Inorganic Waste Using the Yolo Algorithm at Solo Technopark

Rafel Fernando[1*], Afu Ichsan Pradana[2], Sopingi[3]

[1,2,3]*Universitas Duta Bangsa Surakarta, Faculty Computer Science, Information System, Jl. Bhayangkara No.55, Tipes, Serengan District, Surakarta City, Central Java 57154, Indonesia*

Keywords
Computer Vision; Deep Learning; Object Detection; Waste; Yolo

Corresponding Author:
220101031@mhs.udb.ac.id

Abstract
The advancement of artificial intelligence (AI) technology, particularly in the area of computer vision, has encouraged the use of automatic object detection methods for various needs, including the classification of organic and inorganic waste. The problem of waste management in the Solo Technopark area which is still carried out manually causes the waste sorting process to not run optimally. This research focuses on developing and evaluating the performance of several YOLO models for detecting and classifying organic and inorganic waste types in real-time. The research dataset contains 6,758 waste images categorized into 10 object classes, obtained from Roboflow. The preprocessing stages include annotation, auto-orientation, and image resizing to 640×640 pixels. The dataset is then divided into 70% training data, 20% validation, and 10% testing. This study used three YOLO models, namely YOLOv11, YOLOv12, and YOLOv26 with epoch variations of 10, 30, 50, and 100. Model evaluation was carried out using precision, recall, mAP50, mAP50-95, and inference time metrics. The results showed that the best model was obtained on YOLOv26 epoch 100 with a precision value of 0.92, recall of 0.847, mAP50 of 0.892, mAP50-95 of 0.741, and inference time of 3.0 ms. These findings indicate that the YOLOv26 model has good capabilities in detecting and classifying organic and inorganic waste accurately and quickly, so it has the potential to be used as a basis for developing a real-time waste detection system.

1. Introduction

The rapid development of digital technology and artificial intelligence (AI) in recent years has been particularly prominent in the fields of computer vision and deep learning. One of the most widely used AI developments today is object detection technology, a method capable of automatically detecting and recognizing objects through digital images [1]. The applications of this technology span multiple industries, most notably within the security domain, health, industry, transportation, and environmental management. However, waste management remains a critical environmental issue in Indonesia, primarily due to the increasing volume of waste year after year in line with population growth and community activities [2]. Based

on data from the National Waste Management Information System (SIPSN) of the Ministry of Environment and Forestry in 2024, the amount of waste in Surakarta City reached more than 153,000 tons per year, yet much of this waste remains unmanaged. This situation demonstrates the need for a technology-based approach that can assist in the automatic identification and classification of waste [3].

A similar problem was also found in the Solo Technopark area, where organic and inorganic waste are often mixed. Furthermore, the lack of separate trash bins means that waste sorting is ineffective. This situation forces sanitation workers to collect waste quickly without an optimal classification process. Therefore, a detection model is needed that can automatically recognize and classify waste types to support a better sorting process. One approach widely applied In the area of computer vision for instantaneous object detection is YOLO (You Only Look Once) algorithm [4] [5] . YOLO is a Convolutional Neural Network (CNN)-based method that can classify and localize objects in a single processing stage, resulting in high inference speed and good accuracy [6].

The YOLO algorithm has undergone significant developments over several versions, from YOLOv1 to the latest versions such as YOLOv11, YOLOv12, and YOLOv26 [7] . Each version presents enhancements in accuracy, computational effectiveness, and the capability to generalize over various image conditions. YOLOv26, as one of the latest versions, uses an NMS-free architecture that enhances detection efficiency and reduces inference latency on devices with limited computational resources [8] . In addition, this model also has good capabilities in detecting small objects and overlapping objects, so it is considered suitable for use in the image-based organic and inorganic waste classification process.

Based on several previous studies, the use of the YOLO algorithm in waste detection shows quite good performance. A YOLOv8-based study detected organic and inorganic waste based on Android and obtained a precision value of 0.85 and an mAP50 of 0.69 [9]. Another study used YOLOv9 and obtained a precision value of 0.83 and mAP50 of 0.80 in classifying two main types of waste [10] . In addition, research using YOLOv11 based on transfer learning also showed good results with a precision value of 0.823 and an mAP50 value of 0.84 for six classes of recycled waste [11]. However, most previous studies still use a limited number of waste classes and older YOLO architecture resulting in not so high accuracy and mAP, while this study addresses the research horizon by comparing the latest YOLO models, namely YOLOv11, YOLOv12, and YOLOv26, on a more complex 10-class waste dataset using a comprehensive evaluation including precision, recall, mAP50, mAP50-95, and inference time to identify the most effective model for detecting waste in real-time.

Based on the reviewed literature, several differences can be identified between previous studies and the proposed research. One study utilized YOLOv8 to classify only two waste categories, namely organic and inorganic waste, and evaluated the model using precision and mAP50 metrics [9] . Similarly, another study employed YOLOv9 for the same two-category waste classification problem and reported precision and mAP50 results [10] . Meanwhile, another study applied a transfer learning-based YOLOv11 model to classify six recyclable waste classes and evaluated its performance using precision and mAP50 [11] . Although these studies demonstrated promising results, they focused on a single YOLO architecture and did not provide a direct comparison between different YOLO generations under identical experimental conditions.

In contrast, this study evaluates three recent YOLO architectures, namely YOLOv11, YOLOv12, and YOLOv26, using a more complex dataset consisting of 10 waste classes. Furthermore, all models are trained using the same dataset, training parameters, and experimental settings to ensure a fair comparison. The evaluation is also conducted using more comprehensive metrics, including precision, recall, mAP50, mAP50-95, and inference time. Therefore, this study addresses the existing research gap by providing a systematic comparison of recent YOLO generations for multi-class waste detection and identifying the most effective model for real-time waste classification applications.

To the best of the authors' knowledge, this study is the first to systematically compare YOLOv11, YOLOv12, and YOLOv26 using the same dataset, training configuration, and evaluation framework for multi-class waste detection. The findings provide a comprehensive benchmark of recent YOLO architectures and contribute practical insights for selecting suitable models in real-time waste management systems [12].

2. Research Method

2.1 Yolo V11 Architecture

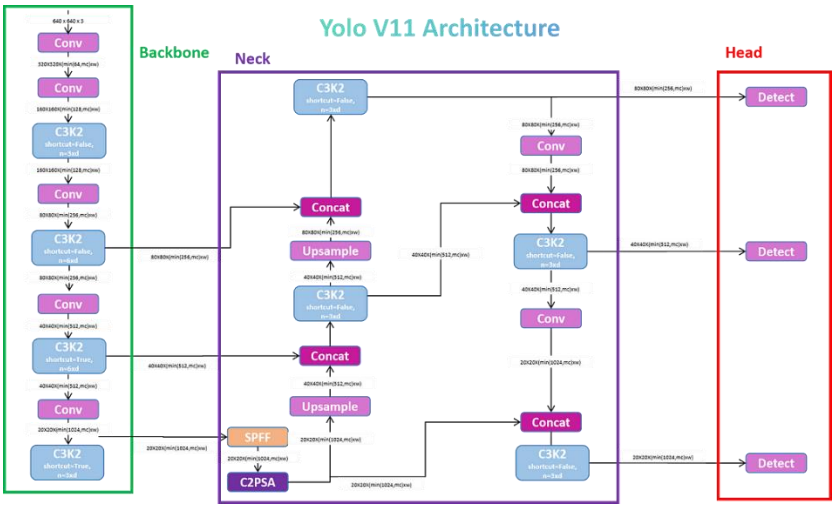

Figure 1. Yolo V11 Architecture

YOLOv11 is the newest generation of the YOLO algorithm family introduced by Ultralytics in 2024 as an extension of YOLOv8 [13] . It incorporates architectural enhancements from previous versions to improve detection efficiency. This model was officially revealed during the YOLO Vision 2024 event and comes in multiple size options: YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, and YOLOv11x, allowing it to be tailored to various needs and application scenarios [13]. YOLOv11's development focused on improving performance in various computer vision tasks, especially in terms of speed of detection, precision, and computational effectiveness in the object recognition procedure [14]. Compared with two-stage detectors such as Faster R-CNN, YOLO performs object localization and classification in a single forward pass, enabling faster real-time detection [15]].

The YOLOv11 architecture enhances feature extraction capabilities to better recognize object details [13] . This model utilizes several key components, including modules such as Convolutional Block with Parallel Spatial Attention (C2PSA), Spatial Pyramid Pooling Fast (SPPF), and Cross-Stage Partial Kernel Size 2 (C3k2), which improve the quality of feature extraction during the detection process. Furthermore, the training pipeline and architecture design were optimized to obtain a balance between model accuracy and performance [13].

YOLOv11 is capable of delivering better results on benchmark datasets thanks to its more optimal training techniques [16] [14] . On the COCO dataset, YOLOv11m achieved a high mAP value with 22% fewer parameters than YOLOv8m [13]. Furthermore, YOLOv11 also boasts offers a faster inference speed compared to YOLOv10, making it more enhancing its suitability for real-time applications that need quick and precise detection [13] . This demonstrates that YOLOv11 marks a major improvement in artificial intelligence-based object detection technology.

2.2 Yolo V12 Architecture

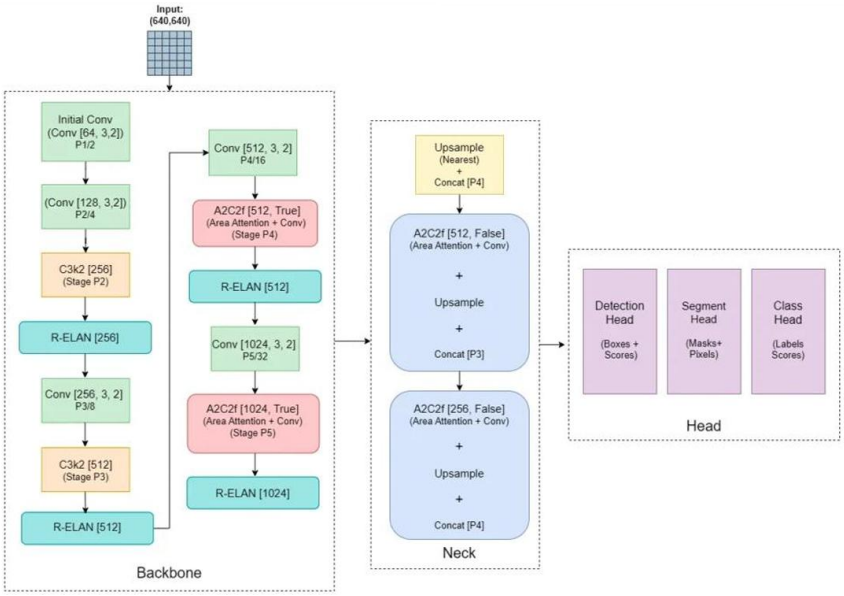

Figure 2. Yolo V12 Architecture

YOLOv12 represents the newest advancement in the YOLO algorithm family, released in 2025. It utilizes an attention-based approach to improve real-time object detection performance [17]. Unlike previous versions of YOLO, which predominantly used a Convolutional Neural Network (CNN) architecture [18] , YOLOv12 integrates an efficient attention mechanism without compromising inference speed, adapting it for use on devices that have restricted computational capabilities. This attention-based design is inspired by the Transformer architecture introduced by Vaswani et al. [19]], which demonstrated the effectiveness of attention mechanisms in learning contextual relationships. YOLOv12 comes in various model versions: YOLOv12n, YOLOv12s, YOLOv12m, YOLOv12l, and YOLOv12x, allowing models to be tailored to performance requirements and hardware capacity.

The YOLOv12 architecture is composed of three primary components: the backbone, neck, and head. In the backbone segment, YOLOv12 utilizes a Residual Efficient Layer Aggregation Network (R-ELAN) architecture integrated with FlashAttention-driven Area Attention (A^2) to enhance feature extraction capabilities and computational efficiency. The residual learning concept adopted in modern backbone architectures originates from ResNet, which demonstrated that residual connections facilitate the training of deeper neural networks and improve feature representation [20]]. Furthermore, this model utilizes depthwise separable convolution with 3x3 and 7x7 kernels to increase the receptive field while maintaining computational complexity at a relatively low level. The neck fuses multi-level attributes from the backbone through an enhanced feature pyramid architecture combined with an attention mechanism, while the head produces predictions consisting of bounding boxes, confidence scores, and class labels.

YOLOv12 was developed to balance elevated precision and processing effectiveness for various object detection applications. The YOLOv12n and YOLOv12s variants are designed for edge devices and low-latency real-time applications, while YOLOv12m, YOLOv12l, and YOLOv12x focus on improving accuracy on devices with higher computational capabilities. By integrating the attention mechanism and architectural optimizations, YOLOv12 improves object detection quality, particularly for small objects, stacked objects, and complex environments. This makes YOLOv12 one of the most effective and significant contemporary object detection models for various computer vision-based implementations.

2.3 Yolo V26 Architecture

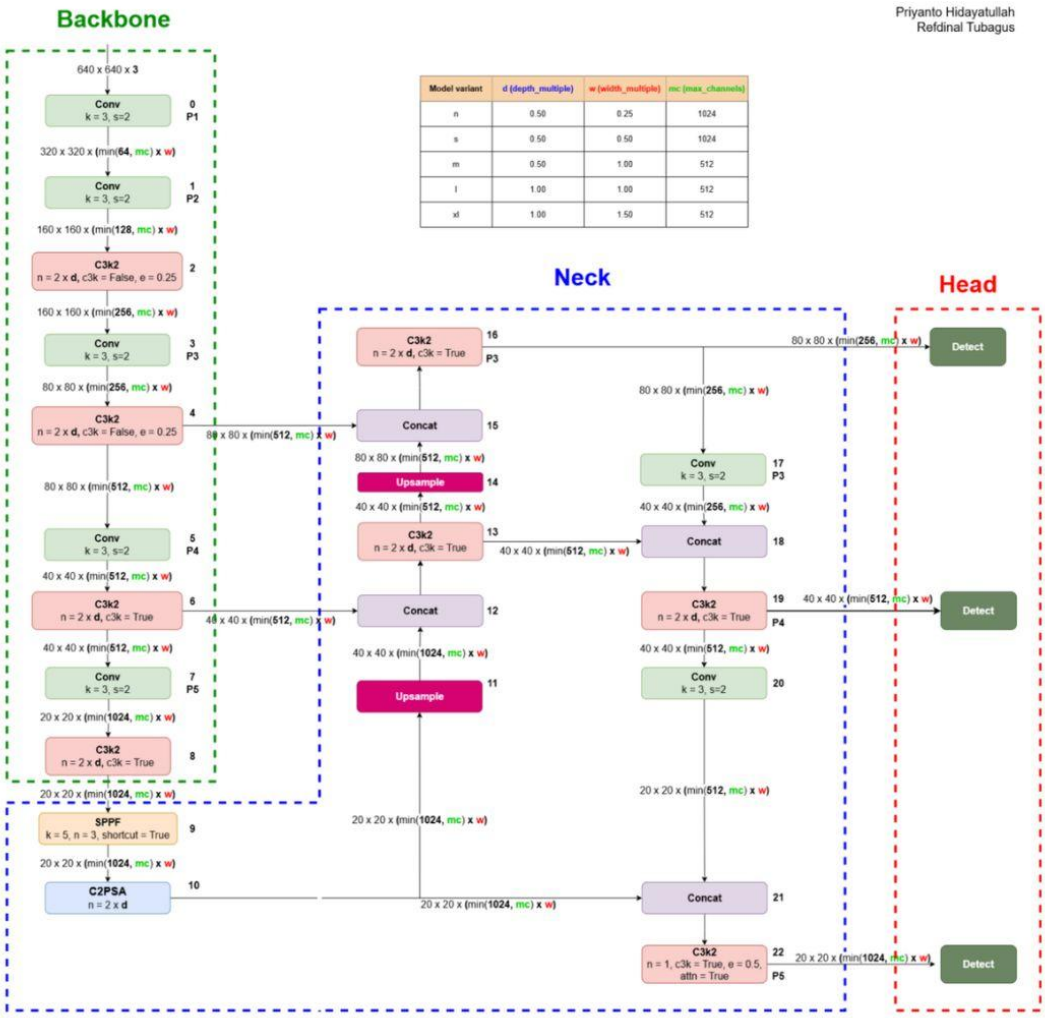

Figure 3. Yolo V26 Architecture

YOLOv26 represents the newest version in the YOLO algorithm family, released in January 2026 as a further development of YOLOv11 and YOLOv12. This model was developed to improve real-time object detection performance with a focus on computational efficiency, inference speed, and better generalization capabilities across various image conditions [21]. YOLOv26 is available in several model size variants: YOLOv26n (nano), YOLOv26s (small), YOLOv26m (medium), YOLOv26l (large), and YOLOv26x (extra large), allowing it to be tailored to device requirements and implementation complexity [21] . The nano and small variants are primarily intended for edge devices or systems with limited resources, while the medium to extra-large variants focus on high accuracy requirements with greater computational capabilities.

The YOLOv26 structure consists of three primary parts: the backbone, neck, and head. In its backbone, YOLOv26 employs a mix of convolutional layers, the C3k2 module, SPPF (Spatial Pyramid Pooling Fast), and C2PSA (Cross Stage Partial Spatial Attention) to enhance feature extraction for objects of varying sizes [21]. The attention module allows the model to focus more on significant feature regions within the image [22] , resulting in more accurate detection, especially for small and complex objects. Furthermore, the YOLOv26 backbone is designed to generate multi-scale feature maps that better preserve the spatial details of objects.

In the neck, YOLOv26 uses a concat and upsample-based a feature fusion approach to integrate feature data from different scales. This process helps the model detect objects of small, medium, and large sizes more

effectively. Furthermore, in the head, YOLOv26 uses a detection layer to generate bounding box predictions, confidence scores, and object classification directly in a single inference process. Compared to previous versions, YOLOv26 optimizes the network structure and inference process, enabling fast detection with more efficient parameter usage. Therefore, YOLOv26 is very suitable for use in the deployment of real-time object recognition systems, including in research on organic and inorganic waste classification based on computer vision.

2.4 Training Model Parameter and Experimental Setup

Model training was carried out on Google Colaboratory using an NVIDIA Tesla T4 GPU as the main computing resource. Annotated organic and inorganic waste datasets were integrated into the Roboflow environment and used as model training data. All input images were resized to 640×640 pixels. This study used three YOLO model variants, namely YOLOv11n, YOLOv12n, and YOLOv26n, using the PyTorch-based Ultralytics YOLO framework with explicitly specified training configurations. The training procedure used the AdamW optimizer with an initial learning rate of 0.01 and a batch size of 16. Training epochs varied at 10, 30, 50, and 100 to determine the effect of iteration numbers on model performance.

All experiments were conducted using the same parameters for each model to ensure objective performance comparisons. The models were initialized using pretrained weights provided by the Ultralytics YOLO framework. A patience value of 20 was applied as an early stopping configuration to reduce overfitting during training. In addition, the confidence threshold was set to 0.25 and the IoU threshold was set to 0.5 during model evaluation. To improve experiment reproducibility, a fixed random seed value of 0 was used and deterministic mode was enabled throughout the training process, ensuring consistent results across repeated training runs with identical configurations. This study did not apply additional augmentation techniques, so all models were trained using the original annotated dataset. Model evaluation was performed using precision, recall, mAP50, mAP50-95, and inference time metrics to assess both detection accuracy and computational efficiency across all YOLO models.

Table 1. Training model parameter and experimental setup

Parameters	Value
Model Variants	Yolo v11n, Yolo v12n, Yolo v26n
Framework	Ultralytics YOLO (PyTorch)
Image Size	640 x 640
Optimizer	AdamW
Confidence Threshold	0.25
IoU Threshold	0.5
Learning Rate	0.01
Batch Size	16
Patience	20
Random Seed	0
Deterministic Training	True
Epoch	10, 30, 50, and 100
Hardware	GPU Tesla T4
Programming Language	Python

3. Methodology

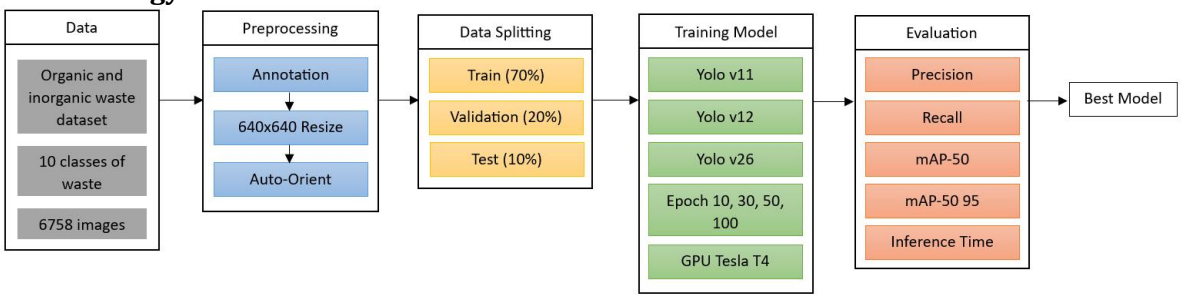

Figure 4. Workflow training model

This study is designed to assess and contrast the effectiveness of different YOLO models in identifying and classifying organic and inorganic waste based on digital images. The research methodology used can be seen in Figure 1 and explained in the following subsections.

3.1 Dataset

The data used in this dataset is organic and inorganic waste from Roboflow https://roboflow.com/ . In this dataset, there are 10 classes used according to the condition of waste in Solo Technopark, namely wet organic, dry organic, glass bottles, plastic bottles, food packaging, cans, plastic bags, cardboard, paper, and tissue with a total of 6,578 images.

Table 2. Dataset class distributions

Class	Count
wet organic	2958
dry organic	1752
glass bottles	842
plastic bottles	1363
food packaging	532
cans	836
plastic bag	282
cardboard	258
paper	762
tissue	267

3.2 Preprocessing

The preprocessing stage is carried out to prepare the dataset before the model training process. The stages involved include:

- Annotation
  At this stage, each trash object in the image is labeled (annotated) using a bounding box format. The labeling process aims to provide information on the object's location and class category in each image so that it can be used as training data for the YOLO model [10].
- 640x640 Resize
  Every image is scaled to 640 × 640 pixels to match the input size needed by the YOLO model. The image is then processed through several initial convolutional layers to reduce the image dimensionality while expanding the number of feature channels. This step is performed to extract and learn important information from the image more effectively and efficiently [23].
- Auto-Orient

This stage is carried out to automatically adjust the image orientation so that the position of the object in the image is consistent for strengthens the ability to generalize to varying object positions [23].

In this study, data augmentation was not used, allowing the model to be trained using raw data that had undergone annotation, auto-orientation, and image size normalization to 640x640 pixels. This decision was made because the dataset used already contained a wide variety of objects, shooting angles, positions, and lighting conditions, making it capable of representing the real-world conditions of the study environment. Furthermore, this study focused on comparing the performance of several YOLO architectures, namely YOLOv11, YOLOv12, and YOLOv26. Therefore, the use of raw data aimed to maintain consistency in model evaluation without the additional influence of data augmentation techniques. Excessive use of augmentation also has the potential to produce unnatural changes in the shape of objects and can affect the original characteristics of the dataset. Therefore, the model was trained using raw images to evaluate its object detection capabilities directly under real-world conditions.

3.3 Data Splitting

The dataset was later divided into three parts:

- 70% for training data
- 20% for validation data
- 10% for testing data

this dataset division was used to train the model, validate its performance during training, and evaluate its final performance on unseen data [24].

3.4 Training Model

This study used several variants of the YOLO algorithm, namely YOLOv11, YOLOv12, and YOLOv26, to detect and classify organic and inorganic waste. The model training process was carried out using epoch variations of 10, 30, 50, and 100 epochs [25], utilizing a Tesla T4 GPU as the computing device.

3.5 Evaluation

- Precision

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

- Recall

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

- mAP

$$\frac{1}{N}\sum_{i=1}^{N} AP_i \quad (3)$$

- Inference Time
  Inference time is utilized to assess the rate of the model in carrying out the object detection process on each image [26].

4. Result and Discussions

4.1 Dataset

At the dataset stage, the data consists of a collection of organic and inorganic waste images with 10 classes used according to the condition of waste in Solo Technopark, namely wet organic, dry organic, glass bottles, plastic bottles, food packaging, cans, plastic bags, cardboard, paper, and tissue with a total of 6,578 images. The research dataset is illustrated in Figure 5.

Figure 5. Organic and inorganic waste dataset

4.2 Preprocessing

In the preprocessing stage, several stages are carried out, starting with annotation, resizing images, and auto-orienting all the garbage images in the dataset. The following are the stages.

- **Annotation**
  In the annotation stage, each waste object in the image is labeled using bounding box annotation. This process aims to provide object location information and class categories in each image so that the dataset can be used in the YOLO model training process. The classes used consist of 10 categories of organic and inorganic waste. The annotation process is performed using dataset labeling tools that support the YOLO format. Each waste object in the image is labeled according to its class type, then the system automatically produces bounding box parameters in YOLO format, comprising the class ID, x-center, y-center, width, and height.
- **640x640 Resize**
  At this point, every image in the dataset is adjusted to 640 × 640 pixels per image to conform to the input dimensions required by the YOLO model.
- **Auto-Orient**
  In the auto-orient stage, the system automatically adjusts the image orientation based on the image metadata to maintain consistent image position. This process is performed to prevent upside-down or tilted images when used in model training.

4.3 Data Splitting

In the data splitting stage, the image dataset is split into three subsets, 6,578 images were divided into 4,605 training images (70%), 1,316 validation images (20%), and 657 testing images (10%).

4.4 Training Model

In the first stage, after preprocessing and data splitting, the Generate Version Dataset process is performed, which saves the results of preprocessing and data splitting on the organic and inorganic waste datasets. The generated version can then be used as an API to retrieve the generated version for training on Google Colab.

Then, the model is trained using several YOLO algorithm variants: YOLOv11, YOLOv12, and YOLOv26. Each model is trained using several epoch variations: 10, 30, 50, and 100 epochs to determine the effect of the number of training iterations on model performance. The training process is performed using a Tesla T4 GPU for optimal and efficient computation.

```
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 580.82.07              Driver Version: 580.82.07      CUDA Version: 13.0     |
+-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla T4                       Off |   00000000:00:04.0 Off |                    0 |
| N/A   41C    P8             11W /   70W |       0MiB /  15360MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

Figure 6. GPU Tesla T4

The final stage is Model Evaluation, which assesses the performance of each model is assessed using evaluation metrics like precision, recall, mAP50, mAP50-95, and inference time. The evaluation results are utilized to compare each model's evaluate performance and determine the top model based on precision and detection rate. In addition, the best model was also tested using new images outside the training dataset to determine the model's capability to detect waste items under different circumstances. The process for training the Yolo algorithm model is illustrated in Figure 7.

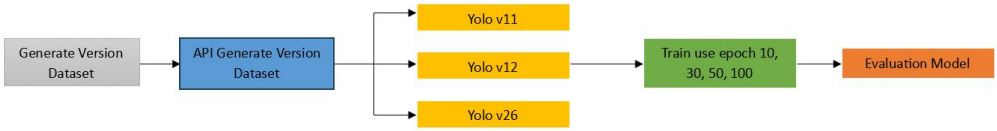

Figure 7. Workflow Training Model Algorithm Yolo in Google Colab

4.5 Evaluation

The results presented in Table 3 were obtained from a single training run for each model using identical datasets, hyperparameters, and training configurations.

Table 3. Comparison Performance Metrics

Model	Epoch	Precision	Recall	mAP-50	mAP-50 95	Inference Time
Yolo v11	10	0.91	0.766	0.878	0.663	5.3ms
	30	0.891	0.834	0.894	0.729	4.9ms
	50	0.912	0.835	0.897	0.747	4.9ms
	100	0.913	0.845	0.898	0.746	4.9ms
Yolo v12	10	0.649	0.376	0.427	0.274	4.3ms
	30	0.761	0.554	0.676	0.478	4.4ms
	50	0.913	0.524	0.722	0.523	4.4ms
	100	0.739	0.709	0.785	0.571	4.4ms
Yolo v26	10	0.832	0.798	0.863	0.69	2.7ms
	30	0.908	0.835	0.886	0.738	2.7ms
	50	0.914	0.84	0.891	0.737	2.7ms
	100	0.92	0.847	0.892	0.741	3.0ms

- **Precision**
  Based on the evaluation results in Table 3, YOLOv26 achieved the best overall precision performance among the three evaluated models. YOLOv26 obtained the highest precision value of 0.920 at epoch 100, followed by YOLOv11 with a precision value of 0.891, while YOLOv12 achieved a lower

precision value of 0.739. Precision reflects the model's ability to correctly classify detected objects and minimize false-positive predictions. A higher precision value indicates that the detected waste objects are more likely to belong to the correct class. The superior precision performance of YOLOv26 can be attributed to the implementation of the C2PSA (Cross Stage Partial with Spatial Attention) module. This module enhances feature extraction by focusing on important spatial regions while maintaining efficient information flow through cross-stage connections. By selectively emphasizing relevant spatial features while reducing redundant feature propagation, C2PSA enables the model to learn more discriminative representations of waste objects. Consequently, YOLOv26 can better distinguish between visually similar waste categories and reduce false-positive classifications, resulting in higher precision performance.

In contrast, YOLOv12 demonstrated unstable training behavior throughout the experiment. The precision value increased significantly from 0.649 at epoch 10 to 0.913 at epoch 50 but subsequently decreased to 0.739 at epoch 100. This pattern indicates that the model reached its optimal learning stage around epoch 50 and then began to experience overfitting. During the early training stages, the model successfully learned discriminative object features, leading to a substantial improvement in precision. However, as training continued, the model increasingly adapted to specific patterns and noise present in the training data rather than learning generalizable representations. As a result, its ability to classify unseen validation samples deteriorated, causing an increase in false-positive predictions and a reduction in precision performance. The observed decline in precision after epoch 50 suggests a reduction in generalization capability and may indicate the onset of generalization collapse. In this condition, additional training no longer improves model performance but instead causes the learned feature representations to become overly specialized toward the training dataset. Consequently, the model becomes less robust when confronted with unseen samples, leading to unstable validation performance despite continued optimization of the training objective.

From an architectural perspective, the instability of YOLOv12 may be associated with the combined use of the R-ELAN backbone and FlashAttention mechanism. R-ELAN expands feature aggregation pathways to capture richer and more diverse feature representations, while FlashAttention efficiently models long-range dependencies among image regions through attention-based computations. Although these components increase representational capacity, they also introduce a more complex optimization landscape compared to conventional convolution-based architectures. On medium-sized datasets such as the waste classification dataset used in this study, the high-capacity attention-based architecture may learn dataset-specific characteristics more aggressively than simpler architectures. As a result, the model becomes more sensitive to class distribution variations, intra-class appearance diversity, and limited sample availability, leading to slower convergence and larger performance fluctuations across training epochs.

Compared with YOLOv12, the C2PSA mechanism employed by YOLOv26 provides a more balanced approach between feature enhancement and optimization stability. The cross-stage partial structure preserves gradient flow and reduces feature redundancy, while the spatial attention mechanism selectively focuses on informative image regions without introducing excessive architectural complexity. This design allows YOLOv26 to achieve more stable feature learning and smoother convergence behavior throughout training. Consequently, YOLOv26 maintains consistent precision improvements as the number of epochs increases and avoids the severe performance degradation observed in YOLOv12 after prolonged training.

Compared with previous studies, the precision values obtained in this research are generally higher. A precision value of 0.85 was reported using YOLOv8 for organic and inorganic waste detection [9], while a precision value of 0.83 was achieved using YOLOv9 [10]. Furthermore, a transfer learning-based YOLOv11 model obtained a precision value of 0.823 [11]. The highest precision achieved by YOLOv26 in this study (0.920) indicates that recent YOLO architectures equipped with advanced attention mechanisms and optimized feature extraction modules can provide more discriminative feature representations and superior classification performance, even when evaluated on a more complex dataset consisting of ten waste categories.

- **Recall**
  The test results in Table 3 show that the best recall value was obtained by YOLOv26 at epoch 100 with a value of 0.847, followed closely by YOLOv11 with a recall value of 0.845 and Yolov12 recall value of 0.709. Recall measures the model's ability to detect all objects present in an image. A higher recall value indicates fewer false-negative detections and better object coverage. The high recall achieved by YOLOv26 demonstrates its effectiveness in identifying waste objects across multiple categories. The spatial attention mechanism implemented in the C2PSA module enables the model to preserve important object information during feature extraction, thereby improving the probability of detecting waste objects that vary in size, shape, and appearance.

  Meanwhile, YOLOv12 produced a relatively low recall value, particularly during the early training stage where the recall was only 0.376 at epoch 10. Although the performance improved in subsequent epochs, the fluctuations observed throughout training indicate that the model struggled to learn stable feature representations. This behavior is consistent with the precision analysis and further suggests that YOLOv12 was more sensitive to dataset variations than YOLOv11 and YOLOv26. Overall, the recall values obtained by YOLOv26 and YOLOv11 indicate that both models are capable of minimizing missed detections. This capability is particularly important in waste detection systems because undetected waste objects may reduce the effectiveness of automatic waste classification and sorting processes.
- **mAP-50**
  Based on the evaluation results, YOLOv11 achieved the highest mAP50 value of 0.898 at epoch 100, while YOLOv26 achieved a slightly lower mAP50 value of 0.892. The difference between the two models is only 0.006, indicating that their detection performances are highly comparable. Since the experiments were conducted using a single training run for each model, this small difference should be interpreted with caution because statistical variability across multiple runs was not evaluated. Therefore, it cannot be conclusively stated that YOLOv11 significantly outperforms YOLOv26 in terms of mAP50.

  The mAP50 metric evaluates the model's detection accuracy at an Intersection over Union (IoU) threshold of 0.50. Higher mAP50 values indicate better agreement between predicted and ground-truth bounding boxes. In addition, YOLOv11 achieved an mAP50-95 value of 0.746, while YOLOv26 achieved 0.741, further confirming that both models possess strong localization and classification capabilities. Although YOLOv11 obtained the highest mAP50 score, YOLOv26 achieved the highest precision (0.920), the highest recall (0.847), and the fastest inference time (3.0 ms). These results indicate that YOLOv26 provides a better overall balance between detection accuracy and computational efficiency. For real-time waste detection applications, inference speed and detection consistency are critical factors alongside mAP performance. Therefore, despite having a slightly lower mAP50 value, YOLOv26 was selected as the most suitable model because it offers competitive detection accuracy while significantly reducing computational latency.

  Compared with previous studies, the mAP50 values obtained in this research remain higher than those previously reported for YOLOv8-based waste detection (0.69) [9], YOLOv9-based waste detection (0.80) [10], and the transfer learning-based YOLOv11 model (0.84) [11]. These improvements suggest that recent YOLO architectures provide more effective feature extraction and object localization capabilities for multi-class waste detection tasks.
- **Inference Time**
  Based on the test results, YOLOv26 achieved the fastest inference time among all evaluated models. At epoch 100, YOLOv26 required only 3.0 ms to process an image, whereas YOLOv11 required 4.9 ms and YOLOv12 required 4.4 ms. Inference time represents the speed at which a model performs object detection and is a critical factor for real-time applications. The superior inference speed of YOLOv26 can be explained by its NMS-free detection head architecture. Conventional YOLO models, including YOLOv11 and YOLOv12, rely on a Non-Maximum Suppression (NMS) stage to eliminate overlapping bounding box predictions after detection. Although effective, this post-processing step introduces additional computational overhead and increases overall inference latency.

In contrast, YOLOv26 eliminates the NMS process by performing prediction assignment directly within the detection head. As shown in Table 3, this architectural improvement contributes directly to the lower inference times achieved by YOLOv26 (2.7–3.0 ms) compared with YOLOv11 (4.9–5.3 ms) and YOLOv12 (4.3–4.4 ms). Therefore, the NMS-free design not only simplifies the detection pipeline but also enhances computational efficiency for real-time deployment. The combination of competitive detection accuracy and the fastest inference speed makes YOLOv26 the most suitable model for real-time waste detection systems. These findings suggest that YOLOv26 provides the best balance between accuracy and computational efficiency among the evaluated models.

4.6 Best Model

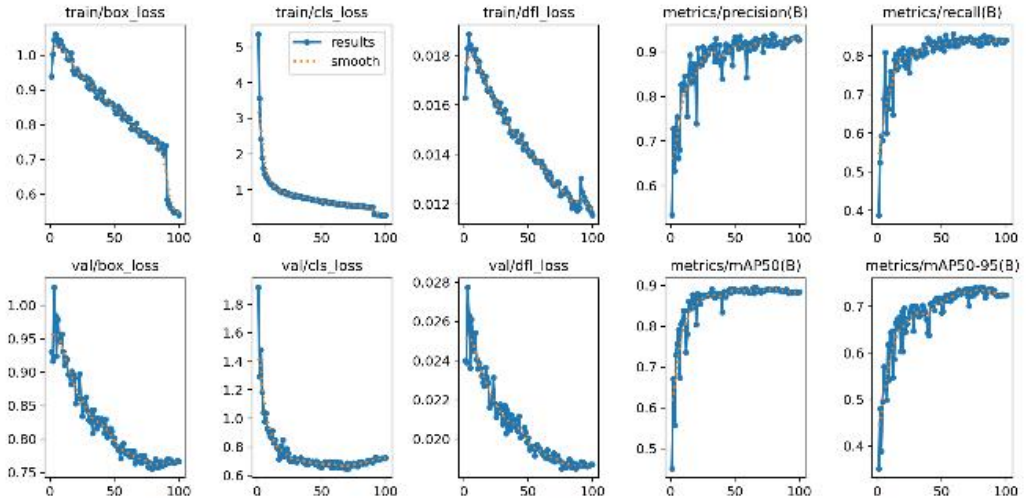

Figure 8. Training result graph

Based on the training results shown in Figure 8, the YOLOv26 model at epoch 100 demonstrates a gradual decrease in both training and validation loss values as the number of training epochs increases. The reductions in train/box_loss, train/cls_loss, and train/dfl_loss indicate that the model progressively improved its ability to learn object localization and classification patterns from the waste dataset. Furthermore, the validation loss curves exhibit a similar downward trend throughout the training process, suggesting that the model maintained relatively stable performance on the validation dataset and did not exhibit clear signs of severe overfitting during training. Based on the overall evaluation results, YOLOv26 at epoch 100 was selected as the best model because it achieved the highest precision (0.920) and recall (0.847) while maintaining competitive mAP values and the fastest inference time among all evaluated models. Although YOLOv11 achieved a slightly higher mAP50 value (0.898) than YOLOv26 (0.892), the difference was only 0.006 and should be interpreted with caution because the experiments were conducted using a single training run. Therefore, the selection of YOLOv26 was based on its overall balance between detection accuracy and computational efficiency rather than on a single evaluation metric. These results indicate that the model is capable of accurately identifying and classifying organic and inorganic waste across multiple categories while maintaining high processing speed for real-time applications.

Several waste classes achieved excellent detection performance, including can with an mAP50 value of 0.992, dry organic with 0.984, and tissue with 0.995. In addition, food packaging, glass bottles, plastic bottles, cardboard, and paper also achieved mAP50 values above 0.90. These results indicate that YOLOv26 successfully learned discriminative features for most waste categories and was able to distinguish between classes with high accuracy. However, several classes still exhibited lower detection performance, particularly the wet organic class, which achieved a recall value of 0.411 and an mAP50 value of 0.553. This lower performance may be attributed to several factors, including high visual similarity with other organic waste categories, inconsistent object shapes, overlapping waste objects, and substantial variations in texture and lighting conditions. Although the wet organic class contained the largest number of images in the dataset

(2,958 images), the class also exhibited high intra-class variability. As a result, the model faced difficulties in learning consistent feature representations, leading to a relatively high number of false-negative detections.

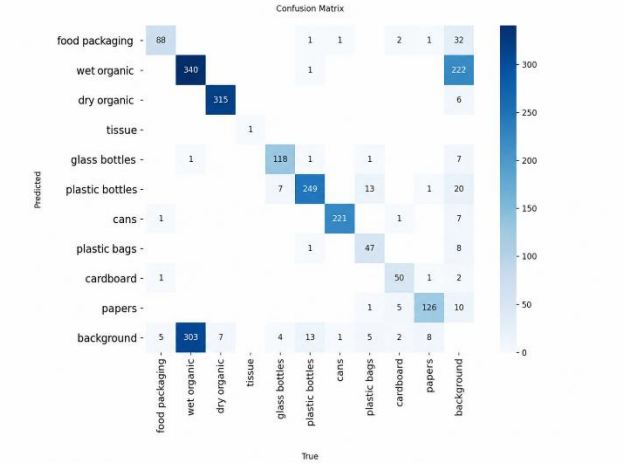

Figure 9. Confusion Matrix

Based on the confusion matrix, most waste categories were classified correctly, as indicated by the high concentration of values along the diagonal. Classes such as dry organic (315), wet organic (340), plastic bottles (249), cans (221), and paper (126) achieved a high number of correct predictions, demonstrating that the model successfully learned the distinguishing characteristics of these categories. Nevertheless, several misclassifications were still observed. The most significant confusion occurred in the wet organic class, where 303 instances were predicted as background. Similarly, the food packaging class recorded 88 correct detections but also showed 32 instances classified as background. These findings indicate that the model frequently failed to detect certain objects, resulting in false-negative predictions. This observation is consistent with the relatively low recall value obtained by the wet organic class.

In addition, the dataset used in this study was imbalanced, with substantial differences in the number of images among classes. For example, the wet organic class contained 2,958 images, whereas the cardboard and tissue classes contained only 258 and 267 images, respectively. Such imbalance may influence the learning process by causing the model to focus more heavily on dominant classes. The relatively low recall value of the plastic bag class (0.664) may be partially attributed to this imbalance, as the class contained only 282 images and exhibited considerable shape variability. Consequently, both dataset imbalance and visual complexity contributed to variations in class-wise detection performance. Future studies may consider applying class-balancing techniques, such as oversampling, class weighting, or data augmentation, to improve the representation of minority classes and enhance overall detection performance. In addition, evaluating the model on a more diverse and larger dataset is necessary to avoid benchmark pitfalls and to ensure reliable generalization under real-world, out-of-distribution conditions [27].

Table 4. Best model yolo v26

Class	Precision	Recall	mAP-50	mAP-50 95
All	0.92	0.847	0.892	0.741
food packaging	0.946	0.926	0.938	0.782
wet organic	0.744	0.411	0.553	0.394
dry organic	0.99	0.972	0.984	0.969
tissue	0.818	1	0.995	0.995
glass bottles	0.966	0.899	0.927	0.638
plastic bottles	0.923	0.91	0.951	0.633
cans	0.982	0.958	0.992	0.731

Class	Precision	Recall	mAP-50	mAP-50 95
plastic bags	0.937	0.664	0.749	0.587
cardboard	0.961	0.828	0.915	0.817
papers	0.93	0.905	0.919	0.864

5. Conclusions and Future Works

According to the results of this study, the YOLO algorithm can be effectively used to detect and classify organic and inorganic waste. Among the evaluated models, YOLOv26 at epoch 100 achieved the best overall performance, with a precision of 0.920, recall of 0.847, mAP50 of 0.892, mAP50-95 of 0.741, and an inference time of 3.0 ms. These results indicate that YOLOv26 provides a favorable balance between detection accuracy and computational efficiency, making it suitable for real-time waste detection applications. Although YOLOv11 achieved a slightly higher mAP50 value (0.898), the difference compared to YOLOv26 (0.892) was only 0.006. Therefore, the selection of YOLOv26 as the best model was based on its overall performance across multiple evaluation metrics rather than a single metric. Despite achieving promising results, this study has several limitations. Some waste classes, such as wet organics and plastic bags, still exhibited relatively lower detection performance due to variations in object shape, overlapping objects, and lighting conditions. In addition, data augmentation techniques were not applied. Furthermore, each model was evaluated using a single training run, meaning that statistical variability across repeated experiments could not be measured. Therefore, future research is recommended to utilize larger and more diverse datasets, apply data augmentation techniques, conduct multiple training repetitions, and evaluate the models in real-world environments to improve the reliability and generalization of the results.

6. References

[1] I. Zaenudin and A. B. Riyan, "Perkembangan Kecerdasan Buatan (AI) Dan Dampaknya Pada Dunia Teknologi," *Jurnal Informatika Utama*, vol. 2, no. 2, pp. 128–153, Nov. 2024, doi: 10.55903/jitu.v2i2.240.

[2] N. N. Kartika Yanti, "Analisis Kebijakan Publik dalam Pengelolaan Sampah Perkotaan di Indonesia: Studi Literatur tentang Efektivitas Implementasi dan Tantangan," *Locus*, vol. 18, no. 1, pp. 100–119, Feb. 2026, doi: 10.37637/locus.v18i1.2683.

[3] Y. S. Nur'aini, N. Al Zahra, M. F. Ilham, I. T. Kusnadi, S. Bahri, and T. S. N. Koeswara, "Sistem Klasifikasi Sampah Berbasis YOLOv8 dengan Pemicu Ultrasonik untuk Efisiensi Daya," *Jurnal Ilmiah Sistem Informasi*, vol. 5, no. 2, pp. 141–153, Jan. 2026, doi: 10.51903/pg864m25.

[4] M. A. Alghiffary, Y. Saputra, S. N. Ali, S. Sumanto, G. Taufig, and J. T. Kumalasari, "Analisis Kinerja Model Yolov8 Berbasis Roboflow pada Deteksi Sampah Plastik Non-Plastik Otomatis," *Semnas Ristek (Seminar Nasional Riset dan Inovasi Teknologi)*, vol. 10, no. 1, pp. 454–459, Jan. 2026, doi: 10.30998/semnasristek.v10i1.8896.

[5] E. Arkin, N. Yadikar, X. Xu, A. Aysa, and K. Ubul, "A survey: object detection methods from CNN to transformer," *Multimed. Tools Appl.*, vol. 82, no. 14, pp. 21353–21383, Jun. 2023, doi: 10.1007/s11042-022-13801-3.

[6] D. Siahaan, "Penerapan Model YOLO untuk Deteksi Kerusakan Jalan Berdasarkan Citra Visual," *Jurnal Ilmiah Matematika*, vol. 13, no. 2, pp. 198–199, 2025.

[7] I. Beqqali Hassani, S. Benhida, N. Lamii, K. Oqaidi, A. Ouiddad, and S. Ghiadi, "From YOLO V1 to YOLO V11: comparative analysis of YOLO algorithm (review)," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 16, no. 1, p. 450, Feb. 2026, doi: 10.11591/ijece.v16i1.pp450-462.

[8] R. Sapkota, R. H. Cheppally, A. Sharda, and M. Karkee, "YOLO26: Key Architectural Enhancements and Performance Benchmarking for Real-Time Object Detection," 2026.

[9] M. R. Ridha, S. Syafrijon, Y. Hendriyani, and A. Hadi, "Implementasi Model Yolov8 untuk Deteksi Jenis Sampah Organik dan Anorganik Berbasis Android," *Abdimas Indonesian Journal*, vol. 5, no. 1, pp. 419–426, Apr. 2025, doi: 10.59525/aij.v5i1.655.

[10] D. Adriansyah and M. P. K. Putra, "Sistem Deteksi Objek Visual Sampah Organik Dan Anorganik Berbasis Algoritma YOL0v9," *Journal of Information System Research (JOSH)*, vol. 6, no. 2, Jan. 2025, doi: 10.47065/josh.v6i2.6454.

[11] V. T. Inkiriwang and M. H. Tinambunan, "Implementasi Model YOLO11N Berbasis Transfer Learning untuk Deteksi Sampah Daur Ulang," *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, vol. 11, no. 1, pp. 1298–1313, Jan. 2026, doi: 10.36341/rabit.v11i1.7323.

[12] M. Ramadhoni, "Pengembangan Model Deteksi Sampah Berbasis YOLOV8 Dan Evaluasi Performanya Dalam Sistem Monitoring Lingkungan Sungai," *Jurnal Profesi Insinyur Universitas Lampung*, vol. 6, no. 2, 2025.

[13] D. Wang *et al.*, "SDS-YOLO: An improved vibratory position detection algorithm based on YOLOv11," *Measurement*, vol. 244, p. 116518, Feb. 2025, doi: 10.1016/j.measurement.2024.116518.

[14] R. Sapkota, Z. Meng, and M. Karkee, "Synthetic meets authentic: Leveraging LLM generated datasets for YOLO11 and YOLOv10-based apple detection through machine vision sensors," *Smart Agricultural Technology*, vol. 9, p. 100614, Dec. 2024, doi: 10.1016/j.atech.2024.100614.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[16] Z. F. Elsharkawy, "Enhanced YOLOv11 framework for high precision defect detection in printed circuit boards," *Sci. Rep.*, vol. 15, no. 1, p. 42550, Nov. 2025, doi: 10.1038/s41598-025-27415-w.

[17] S. E. Prasetyo, G. Wijaya, and A. Kwan, "Klasifikasi Kematangan Buah Pisang Menggunakan YOLOv12 Berbasis Deep Learning," *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 5, no. 1, pp. 30–39, Feb. 2026, doi: 10.55123/storage.v5i1.7557.

[18] A. Raza, F. Hanif, and H. A. Mohammed, "Analyzing the enhancement of CNN-YOLO and transformer based architectures for real-time animal detection in complex ecological environments," *Sci. Rep.*, vol. 15, no. 1, p. 39142, Nov. 2025, doi: 10.1038/s41598-025-26645-2.

[19] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[21] Ultralytics, "Ultralytics YOLO26," *Ultralytics Docs*, 2025. [Online]. Available: https://docs.ultralytics.com/models/yolo26/. Accessed: Jun. 25, 2026.

[22] F. Li, H. Yan, and L. Shi, "Multi-scale coupled attention for visual object detection," *Sci. Rep.*, vol. 14, no. 1, p. 11191, May 2024, doi: 10.1038/s41598-024-60897-8.

[23] M. D. Z. Baig, M. Kamal, and Z. Ullah, "Small Dents, Big Impact: A Dataset and Deep Learning Approach for Vehicle Dent Detection," *arXiv preprint arXiv:2508.15431*, 2025.

[24] B. İnan, İ. A. Özkan, and N. Doğan, “Deep Learning-Based Skin Disease Detection: Comparative Performance Analysis of YOLOv8 and YOLOv11 Models,” *International Journal of Applied Methods in Electronics and Computers*, Dec. 2025, doi: 10.58190/ijamec.2025.155.

[25] M. P. Biswas, “Drowsiness Detection Using ML,” *Interantional Journal of Scientific Research in Engineering and Management*, vol. 08, no. 05, pp. 1–5, May 2024, doi: 10.55041/IJSREM32980.

[26] I. D. G. A. W. Natih, M. W. A. Kesiman, and I. M. G. Sunarya, “Analisis Perbandingan Arsitektur dan Optimizer YOLOv11 untuk Estimasi Buah Kelapa,” *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 4, no. 4, pp. 12–19, Nov. 2025, doi: 10.31004/riggs.v4i4.3329.

[27] C. Wu, W. He, C.-H. Cheng, X. Huang, and S. Bensalem, “Revisiting Out-of-Distribution Detection in Real-Time Object Detection: From Benchmark Pitfalls to a New Mitigation Paradigm,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 48, no. 5, pp. 5243–5260, May 2026, doi: 10.1109/TPAMI.2025.3650695.