Available online at www.HighTechJournal.org

HighTech and Innovation Journal

ISSN: 2723-9535

Vol. 7, No. 2, June, 2026

# Smart Waste Sorting Through Advanced Computer Vision: Optimizing YOLOv11 for High-Accuracy Waste Classification

Lasmedi Afuan [1*], Agus Darmawan [1], Raden Demas Amirul Plawirakusumah [1], Putranto Surya Wijanarko [1]

[1] *Department of Informatics, Universitas Jenderal Soedirman, Central Java, Purwokerto, Indonesia.*

Received 13 February 2026; Revised 21 April 2026; Accepted 03 May 2026; Published 01 June 2026

**Abstract**

This research aims to develop a high-accuracy, real-time waste classification system to overcome the inefficiencies and errors associated with manual sorting. The methodology utilizes the advanced YOLO11x-cls architecture, enhanced through transfer learning and optimized using Stochastic Gradient Descent (SGD). Based on the TrashNet dataset containing 2,390 images across five categories (cardboard, glass, metal, paper, and plastic) the study involved rigorous hyperparameter tuning, identifying an optimal initial learning rate of 0.00075. The system was subjected to systematic hyperparameter tuning using a targeted grid search strategy to identify the optimal balance between convergence speed and stability. Key findings demonstrate a superior testing accuracy of 98.16%, an F1-Score of 0.9816, and an inference speed of 35 FPS, proving the system's readiness for real-time applications. Notably, the SGD optimizer provided better stability and generalization than AdamW on this dataset. The novelty of this study lies in being among the first to implement YOLOv11 for waste management, leveraging its new C3K2 and C2PSA blocks to achieve a 1–5% F1-Score improvement over previous YOLO versions while requiring 20% fewer parameters than YOLOv8. This improvement offers a scalable, lightweight solution for edge-computing devices, directly supporting global environmental sustainability goals. The proposed system offers a scalable and hardware-efficient solution for real-time edge deployment in smart waste management infrastructure.

*Keywords:* YOLOv11; Garbage Classification; Computer Vision; Smart Waste Management; Transfer Learning; Environmental Sustainability.

## 1. Introduction

In the era of rapid urbanization, waste generation has become an unavoidable consequence of human activity, posing significant threats to environmental sustainability and public health. The world's waste generation is expected to reach 70% in 2050 due to the rising population projected to reach 9.7 billion, with developing countries such as Indonesia producing about 0.52 kg of waste per capita each day [1]. It is also possible to divide waste into inorganic and organic forms (e.g., plastics, metals, and glass; food scraps, etc.) based on their non-biodegradability (inorganic) and biodegradability (organic), respectively [2, 3]. Poor waste sorting systems and littering have escalated the environmental degradation in most parts, especially urban and rural parts of Indonesia, causing pollution of the ecosystem, loss of aesthetic value, and health hazards through transmission of diseases [4]. The low recycling rates of the traditional waste management systems, mainly because of the intensive usage of manual sorting methods, indicate a low recycling rate that does not exceed 20 percent in most developing countries, explaining why the advanced technological interventions are so much needed [5].

* Corresponding author: lasmedi.afuan@unsoed.ac.id

https://doi.org/10.28991/HIJ-2026-07-02-015

The primary challenge addressed in this research is the inherent inefficiency and high error rates up to 30% associated with manual waste sorting. Such inefficiency is problematic as it results in more contamination of the recyclables, decreased recycling rates, and increased greenhouse gas emissions from the landfills, which contribute to 5 percent of the world's emissions [6]. These problems are an obstacle to achieving Sustainable Development Goals (SDGs), including SDG 11 (Sustainable Cities and Communities) and SDG 12 (Responsible Consumption and Production). It has been found that AI-based waste management systems have the potential to decrease logistical costs by 13.35%, transportation distances by 36.8%, and processing time by 28.22%, which demonstrates the transformational nature of automation [7]. Thus, the necessity to come up with an intelligent, scalable solution to increase the accuracy and efficiency of waste classification to provide sustainable urban ecosystems and circular economies is the motivational factor behind this study.

This study addresses the performance limitations of inorganic waste classification, particularly in complex real-world scenarios characterized by variable lighting and cluttered backgrounds (cardboard, glass, metal, paper, and plastic) and in different real-life scenarios that may be described with varying lighting, cluttered backgrounds, and limited datasets. The problem is urgent since proper sorting of waste is the key to successful recycling, waste saving, and less harmful effects on the environment. The suggested system can be used to implement applications in smart cities, IoT-powered waste containers, and robotic recycling centers by improving the accuracy of classification, which conforms to the sustainability goals of the world [8].

The proposed research is carried out to mitigate the weaknesses of available state-of-the-art (SOTA) solutions in classifying waste. Initial approaches used simple convolutional neural networks (CNNs), including Keras Sequential, with high accuracies (up to 99) of binary organic vs. non-organic classification but with no real-time performance and inability to work in complex settings. Two-stage detectors such as Faster R-CNN were more precise and shared the drawback of being computationally expensive to run in real time [9]. With the introduction of single-stage detectors, especially the YOLO (You Only Look Once) family, the concept of object detection changed dramatically as it became possible to process data in real-time and make only one pass through the network, as opposed to sliding window or region-proposal-based methods. The YOLOv4 and lightweight version, YOLOv4-tiny, scored 91.25% and 82.02% mAP with a precision of 0.91 against waste segregation tasks, respectively, and are superior to SSD and Cascade-RCNN [10]. YOLOv5 has presented CSPDarknet and PANet with an accuracy of 95.06 in the waste classification of health care with an F1-score of 0.9487 and an inference time of 10.97 ms [7]. YOLOv7 also added spatial attention modules (SAM) and GhostNet, making its results 0.62 in terms of F1-scores in e-waste detection. In 2024, YOLOv8 added anchor-free and C2f models, reaching a peak F1-score of 0.63 and 94.68% accuracy on similar tasks, 1-2 percent better in mAP than YOLOv5 and v7 [11]. Recent models such as multi-scale fusion and coordinate attention based on the MRS-YOLO and YOLOv12 have set mAP in both models to 0.78 at 0.5 and F1-Scores to 0.75, with inference times of around 5 ms/image [12]. It has also been demonstrated that hybrid lightweight models such as MobileNetV3-S (91.05% accuracy) and EfficientNet-B0 (93.22%) can provide promise in resource-constrained systems, although they are slower than optimized versions of YOLO [13].

However, the continuous evolution of YOLO architectures, exemplified by models like YOLOv11 and YOLOv12, has consistently pushed the boundaries of real-time performance and detection accuracy in waste classification, offering robust solutions for automated waste segregation systems [14]. The deployment of these advanced YOLO models is critical for addressing the environmental and health implications of inefficient waste management, particularly in the challenging domain of electronic waste classification [11]. Specifically, YOLOv11 incorporates C3k2 and C2PSA modules to improve feature extraction without incurring accuracy penalties, further enhancing its suitability for real-time applications in diverse waste streams [15]. This architectural refinement enables YOLOv11 to maintain high detection accuracy while significantly reducing computational overhead, making it highly adaptable for integration into automated waste sorting facilities [16]. This efficiency is paramount for scenarios requiring rapid processing, such as robotic waste pickers and smart recycling bins, where even marginal improvements in inference speed can significantly enhance operational throughput and overall sustainability efforts [17].

YOLOv11 is benchmarked against earlier YOLO versions in comparable waste-classification tasks. Based on the CSPDarknet architecture, YOLOv5 achieved accuracy rates ranging from 90.2% to 96.4% on datasets such as TrashNet and other custom waste-classification datasets; however, its generalization performance on smaller datasets was more limited due to the larger number of parameters. YOLOv8 with anchor-free detection rose to 92-97.63% in real-time, but its validation loss in constrained conditions was worse. The better-than-these results are achieved by our optimized YOLOv11 at 96.17% accuracy and reduced loss (0.1254), which is due to the improved feature extraction ability of C3K2 blocks and a 1-5% improvement in the F1-Score of our results on the TrashNet dataset.

In spite of these improvements, SOTA models have drawbacks of limited datasets, environmental variability, and multi-object occlusions, typically consuming large computational resources to hit accuracies in the 90s on benchmarks such as TrashNet or TACO. Our contribution overcomes these limitations by using YOLOv11, published in September 2024 by Ultralytics, which adds C3K2 and C2PSA blocks to provide better feature extraction and fewer parameters, which reach up to 96.17% accuracy on the TrashNet dataset. In contrast to YOLOv12, which uses a highly complex image but not as large a dataset as we would, the optimization points of our model are the combination of speed and accuracy in real-time and high precision on a smaller dataset (as demonstrated by a 0.5-10% higher F1-score in our model compared to EfficientNet or YOLOv8) [12].

The proposed method utilizes a transfer learning-based approach with YOLO11x-cls, fine-tuned using Stochastic Gradient Descent (SGD) through systematic empirical optimization of hyperparameters on the TrashNet dataset, comprising 2,390 images across five waste categories. Research Questions (RQs): (1) How effective is the optimized YOLOv11 in achieving high accuracy for waste classification under limited data? (2) What impact does SGD hyperparameter tuning have on model generalization compared to alternatives like AdamW? Objectives: To develop and evaluate a scalable YOLOv11-based system for real-time waste sorting with superior performance metrics.

The study entails the application of the model to a split dataset (70% train, 20% validation, 10% test) and training the model through 25 epochs and evaluation of performance in terms of Precision, Recall, F1-Score, and confusion matrices. Using the techniques outlined in later passages, we expect to reach a training accuracy of 96.17 as well as a low validation loss and realistic applicability in smart waste management systems, which will be followed by future interactions with robotics and IoT to improve environmental sustainability.

## 2. Method

The proposed system leverages a YOLOv11-based CNN architecture, implemented via the Ultralytics library in Python. YOLO is a single-stage object detection algorithm that efficiently divides images into bounding boxes and object classes in a single process, making it very fast and suitable for real-time applications. The classification process is carried out after the features are extracted, where the model maps the patterns to the appropriate class labels. Before training, data acquisition stages are carried out to collect and annotate images, as well as pre-processing in the form of augmentation to improve data quality and consistency.

### 2.1. Research Flow

The research flow conducted went through several stages as shown in Figure 1.

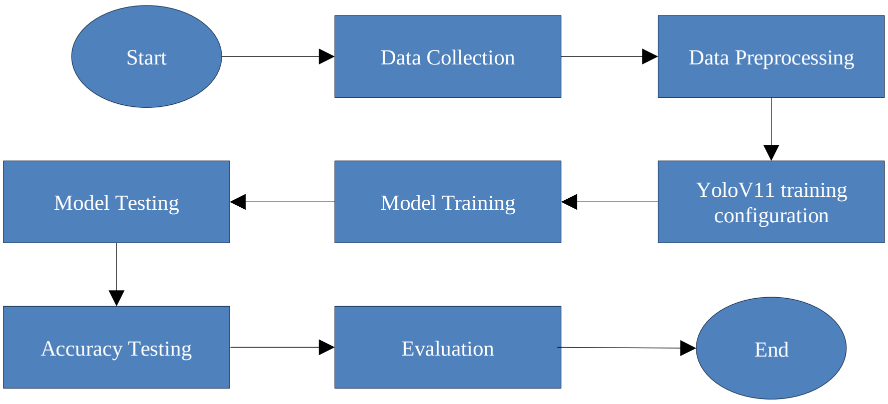


**Figure 1. Workflow diagram illustrating the stages of developing the YOLOv11-based garbage classification model, from problem identification to model evaluation**

Figure 1 represents a general workflow for developing a machine learning model, specifically object detection using YOLOv11. It starts with identifying the problem and conducting a literature review to understand existing solutions and methodologies. Afterward, an appropriate algorithm and method are selected, followed by data collection and preprocessing. Preprocessing ensures the data is in the correct format for model training.

Then, the YOLOv11 model is configured and trained with the help of the ready dataset. After the process of training is done, the model is then tested to verify its performance. In testing accuracy then the Precision and Recall of the model is quantitatively determined. The findings are measured to find out whether the model achieved the objectives of the project. When successful, the process is finalized, otherwise the cycle could be repeated to make the model more refined.

### 2.2. Data Collection

The data utilized in this study is a public one called TrashNet. TrashNet is an open-source collection of images of recyclable waste, which was curated and applied to waste classification. The sample size is 2390 waste images of diverse types, such as paper, plastic, glass, cardboard, and metal. Out of the TrashNet data, selected few classes were only made as depicted in Figure 2.

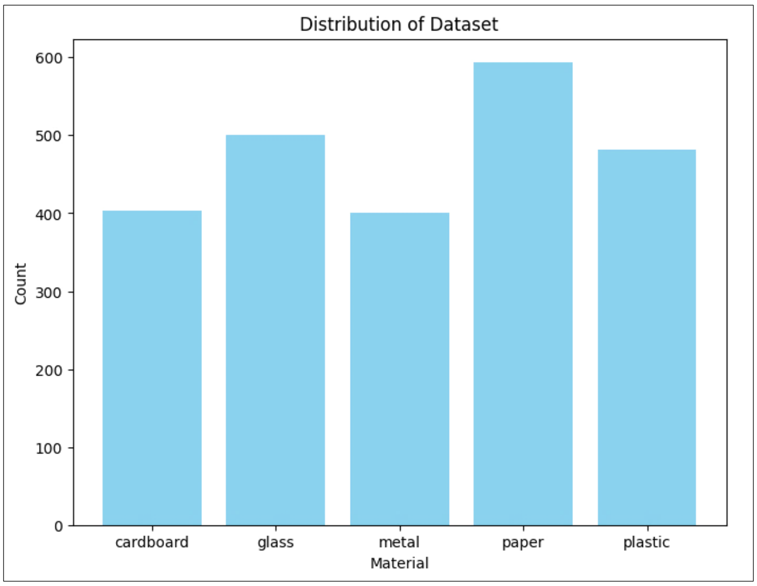


**Figure 2. Distribution of images across the five waste classes (cardboard, glass, metal, paper, plastic) in the TrashNet dataset**

To reduce class imbalance and improve performance on a relatively small sample size, we engaged in data augmentation methods during preprocessing, such as random flips and rotations. These techniques evenly expose the model to different class features and enhance the generalization of the model to unseen real-world data. Additionally, the dataset was split into training (70%), validation (20%), and testing (10%) to avoid overfitting and ensure robust performance evaluation. The TrashNet dataset contains 2527 images distributed in 6 original classes, which are cardboard (403 images), glass (501), metal (410), paper (594) and plastic (482), and trash (137). In this case, the scope was narrowed to five categories of inorganic waste (except trash) and we ended with 2390 images, which was useful in our objectives of recycling. Figure 2 illustrates that there is a weak imbalance in the classes with paper as the most represented (24.9% of the dataset) and cardboard as the least represented (16.9%). These imbalances may cause bias, whereby the over-represented classes such as paper are favored in predictions and hence lowered Recall of underrepresented ones (e.g., cardboard). To reduce this, we engaged in data augmentation methods during preprocessing, such as random flips and rotations, which evenly expose the different classes, and enhances generalization of the model. The reason was that no oversampling was applied to prevent the artificial inflation of the minority classes.

After the data is divided, the dataset is divided into several classes, then the dataset is divided (split dataset) into three categories, namely training data, validation data, and test data. Training data is used to train the model, while validation data is used to evaluate the model during the training process but does not participate in the training process. On the other hand, test data is used to measure the overall performance of the model after the training process is complete [18]. This aims to train the model, configure hyperparameters, conduct evaluations, and test model performance to avoid overfitting and ensure good generalization. The result of the preprocessing is an organized image dataset that is ready to be used for model training in five waste categories. The dataset separation is shown in Table 1.

**Table 1. Dataset separation**

| ***Parameter*** | **Images** | **Percentage** |
|---|---|---|
| Train | 1673 | 70% |
| Validation | 478 | 20% |
| Test | 239 | 10% |

## 2.3. Data Preprocessing

Data can be available in various forms: structured tables, unstructured tables, images, audio files, videos, and so on. Machines cannot directly understand free text, videos, or images as they are; the given data needs to be converted into 1s and 0s. Therefore, raw data cannot be directly fed into a machine learning model and expected to be trained. Data preprocessing is the first step in machine learning [19]. Preprocessing methods are applied to correct errors and improve data quality. At the end of the data preprocessing phase, this data is transformed into a form suitable for data mining algorithms and amplified for more efficient mining operations and easier model understanding [20]. Image down-scaling is the most common method used in preprocessing in classification models [21]. This process is important because deep learning models require uniform input sizes in order to process data in batches. In the developed system, all images will be resized to 224×224 pixels in RGB (Red, Green, Blue) format. This process is done automatically by the image size parameter (imgsz) in the Ultralytics YOLOv11 framework, where according to its official documentation, this size is chosen as the standard because it fits the internal structure of the optimized model. Currently, simple image resizing methods such as nearest neighbor, bilinear, and bicubic are among the most widely used image resizing methods in visual recognition systems [21]. One of the commonly used methods is bilinear interpolation, where the new pixel value is calculated based on the weighted average of its nearest pixels. Mathematically, the bilinear interpolation process can be formulated in Equation 1.

$$f(x,y) = \sum_{i=0}^{1}\sum_{j=0}^{1} w_{ij}.f(x_i,y_j) \tag{1}$$

where, $f(x,y)$ is the pixel value at the desired point, and $w_{ij}$ is the weight based on the distance to neighboring pixels $f(x_i,y_j)$. Then the weight wij is usually calculated based on the distance with the following Equation 2.

$$w_{ij} = (1-|x-x_i|).(1-|y-y_j|) \tag{2}$$

In this formula, the pixel value of a new point is an average of the values of physically near pixel values with weights corresponding to their distance with the nearest ones having larger weighting. Besides resizing, the YOLOv11 model also normalizes the pixels of an image on its own to some range (usually 0-1) and then passes the image to the model. This normalization is supposed to accelerate the convergence process in the training and ensure the values of the weights are not too large or too small.

### 2.4. Model Development

This study adopts a transfer learning approach by utilizing the YOLOv11 architecture that belongs to the YOLO family which provides substantial gains in productivity and accuracy of the object detection. TrashNet This model was fine-tuned on the TrashNet dataset using the ImageNet-trained YOLOv11 model and new labels. The optimiser in the training process is the Stochastic Gradient Descent (SGD) optimiser with hyperparameters in the form of learning rates. The metrics used to carry out Model performance evaluation include Positive predictive value (PPV), Precision, Recall and F1-Score. This measure will be computed using the confusion matrix per class that provides a comprehensive analysis of how the model is able to detect and classify waste.

### 2.5. YOLOv11

YOLOv11 was published in September 2024. The YOLOv11 has followed a sequence of architectural advances, which aim at improving computational performance without interfering with accuracy. YOLOv11 has novel elements, including the C3K2 block as well as the C2PSA block, which leads to enhanced feature extraction and processing, with a greatly lowered count of model parameters. Figure 3 shows the architecture of YOLOv11.

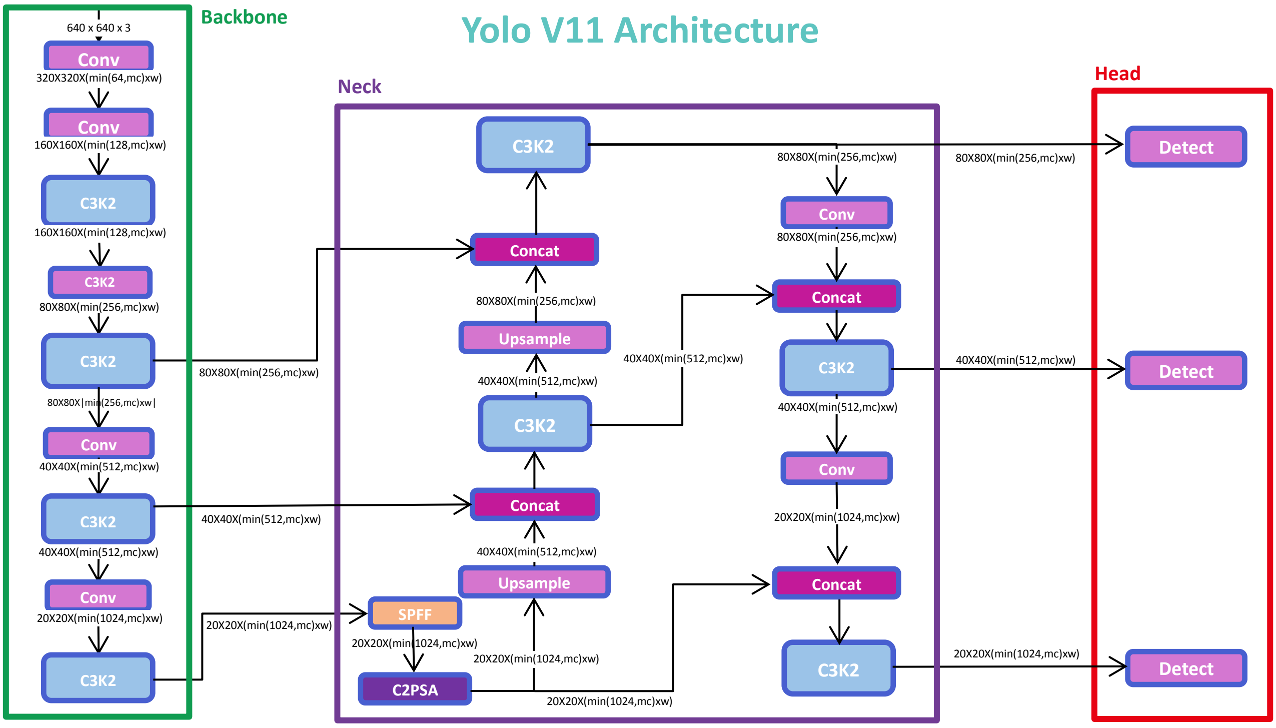


**Figure 3. Architecture of YOLOv11**

The first step that starts with the YOLOv11 architecture is the backbone component, which serves to extract the hierarchical features of the input image. It is performed with the help of the following basic modules: The Conv module is the combination of Conv2d, BatchNorm2d, and SiLU activation functions, which contribute to extracting the low-level spatial features with the stability of the output and better nonlinear representation. The Spatial Pyramid Pooling Fast (SPPF) module is a module that is based on combining multi-scale contexts (local detail to global semantics) with multiple pooling operations and concatenation operations. This module is made to be computationally efficient, as well as to enhance feature representation. Cross-Scale Pixel Spatial Attention (C2PSA) module presents attention mechanism on a pixel scale to intensify the highlighting of the important areas within the picture. The features in this module are highly integrated with the nested PSA blocks and residual connections giving more strength to the local details without compromising global context. The neck part is in charge of taking the multi-scale features extracted by the backbone and refining them. Upsampling, concatenation and other refinement procedures in this stage are completed by C3K2 module. Such process allows combining local detail with global semantic context and features that are more representative of different sizes of objects are produced.

The last phase of the architecture is the component of the head which works to anticipate object classes and the coordinates of the bounding box. The design of this head has a multibranch structure which is optimized to respond to the small, medium and large objects. This module employs depthwise separable convolution (DWGConv) which is a computationally cheap but efficient method of feature extraction. In addition, the resulting features are split into two directions, the bounding box regression direction, which is trained with the help of the box loss function to enhance the Precision of the detection location, and the classification direction, which is trained with the assistance of the classification loss function to enhance the Precision of the object category recognition. The end results of both lines are filtered with the help of the Non-Maximum Suppression (NMS) algorithm to yield the final output in terms of the categories of the objects, coordinates of their bounding box, and true confidence scores [22].

## 2.6. Learning Transfer

The most recent approaches to learning transfer in deep learning also seek to minimize the amount of time and cost of the training phase, and the scale of training datasets, which is challenging to access in certain applications, including medical imaging. Moreover, one can run pre-trained models in a certain task on simple edge devices like a smartphone with low processing power, and limited training time [23]. This approach is referred to as transfer learning and this is because the human being can learn more quickly, conveniently, and effectively through knowledge acquired in the tasks already learned. In this approach, the area, task, and allocation may differ between the training and testing data although they are related somehow [24]. The transfer learning model is applied by using the pre-trained YOLOv11 network trained on the massive COCO (Common Objects in Context) dataset. This approach allows the model to leverage knowledge from a large-scale general-purpose dataset to effectively identify and label objects in TrashNet, which is a much smaller and more specialized dataset.

## 2.7. Optimizer

The optimizer is an important part in deep learning model training, and it is essential in modifying the neural network weights to achieve the lowest loss function. The machine learns through the optimization algorithm on which it relies. The gradient is determined, and the loss maximization is minimized to the lowest. Optimization techniques can be used to implement learning in different ways [25].

Popular methods of optimization are gradient descent (GD) and practical stochastic gradient descent with mini-batch gradients (SGD) which are commonly used to optimize deep neural networks [26]. Among the most significant machine learning tools, there is the SGD method with its variants, i.e., [27]. In deep learning, the SGD algorithm has been used to process natural language processing (NLP), visual data processing, and speech and audio processing. We present several applications of deep learning and provide some practical examples of using the SGD algorithm in these significant areas [28]. SGD is one of the simplest algorithms and it has extensive application in ML algorithms. SGD does not compute the gradient of all the training examples and update the weights as is often the case in training, but rather, it updates the weights per training example [25]. Forms of SGD vary greatly with one of them being SGD with momentum, the type of SGD is applied to assist in preventing traps in local minimum and accelerating the process of convergence. The weight update of SGD can be expressed such that in Equation 3.

$$\begin{aligned} v &= \mu v_{t-1} - \mu \nabla L(w_{t-1} + \mu v_{t-1}) \\ w_t &= w_{t-1} + v_t \end{aligned} \tag{3}$$

## 2.8. Configuration

Machine learning model configuration is a significant issue when developing a model because it directly influences the performance of the learning process and ultimate model performance. This paper implemented the Ultralytics YOLOv11 framework as the model training process with a few parameters optimized depending on the requirements of the TrashNet data. To ensure the reproducibility of the results, a global random seed (e.g., seed=42) was applied within the Ultralytics framework to maintain consistency in weight initialization and data shuffling during the 70/20/10 dataset split. Parameters used in training of the models shown in Table 2.

**Table 2. Training configuration**

| ***Parameter*** | **Value** |
|---|---|
| Epochs | 25 |
| Image Size | 224×224 px |
| Image Total | 2390 |
| Class Total | 5 |
| Workers | 6 |
| Batch Size | 16 |
| Optimizer | SGD |
| Initial Learning Rate (lr0) | 0.00075 |
| Learning Rate Factor (lrf) | 0.2 |

The hyperparameter optimization process followed a two-stage systematic search. Initially, a logarithmic grid search was conducted across a broad range ($10^{-1}$ to $10^{-4}$) to identify the convergence region. This was followed by a fine-grained empirical refinement within the optimal zone, which led to the selection of 0.00075 as the initial learning rate ($lr_0$).", "This value was combined with a cosine decay strategy (Equation 4), where the learning rate factor ($lr_f$) was set to 0.2, ensuring the model could settle into global minima without overshooting during the final 25 epochs.

The training configuration of the development of the YOLOv11 model is presented in Table 2. The training of the model was also carried out using 25 epochs in order to enable the model to learn effectively without over-fitting. Each image was downscaled to 224 × 224 pixels according to the YOLOv11 architecture in order to be computationally efficient. The optimizer used is Stochastic Gradient Descent (SGD) with the addition of Nesterov Momentum. The initial learning rate value ($lr_0$) is set at 0.00075, and will gradually decrease towards 20% of the initial value according to the learning rate factor ($lr_f$) parameter of 0.2. The number of workers is 6 to increase data loading efficiency, and the batch size is set at 16 to balance performance and memory usage.

In Table 2, there are two main parameters used in setting the learning rate, namely the initial learning rate ($lr_0$) and the learning rate factor ($lr_f$). Both of these parameters are used in the learning rate decay process.

Learning rate decay is a technique used to gradually reduce the learning rate value during the model training process, with the aim of increasing convergence stability and preventing overshooting in the final training phase. In this model training, the cosine decay type decay method is used, which can be mathematically stated as follows in Equation 4.

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_0 - \eta_{min})\left(1 + \cos\left(\frac{t\pi}{T}\right)\right) \tag{4}$$

## 2.9. Evaluation

The evaluation of the model is a significant step of the development of a machine learning-based classification system. It is aimed at measuring how well the model is able to classify data. The analysis of this research has been conducted based on some primary measures that are Precision (PPV), Recall, F1-Score and Confusion Matrix.

PPV, F1-Score, Recall and confusion matrix will be used to assess the results of the model classification. Equations 5, 6, and 7 give the formula that is used to derive model evaluation in the form of PPV, F1 score and Recall.

Precision (Positive Predictive Value / PPV) is the ratio of positive predictions which are actually applicable. The reason why this metric is significant in the context of garbage classification is that the system should not give excessive false detections of a certain category. Equation 5 is the formulation of Precision.

$$PPV = \frac{TP}{TP+FP} \tag{5}$$

F1-Score is the harmonic mean of Precision and Recall which gives a comprehensive picture of the model's performance in terms of the balance of both as shown in Equation 6.

$$F1\ Score = \frac{2TP}{2TP+FP+FN} \tag{6}$$

Recall or sensitivity measures how well a model can find all relevant samples for a class. Recall is important to ensure that no type of garbage is missed. The formula is shown in Equation 7.

$$Recall = \frac{TP}{P} \tag{7}$$

Confusion Matrix is referred to have a closer view of what each of the classes performs well in terms of classification. The number of correct and incorrect predictions in each of the classes is displayed in this matrix, and it assists in determining patterns of the model errors, e.g. the cases of misclassification of similar classes.

## 3. Results

In this chapter, the researcher seeks to evaluate the performance of the proposed model in waste classification. The training and evaluation were conducted on a system equipped with an AMD Ryzen 5 5600H processor, 16 GB of RAM, and an NVIDIA RTX 3050 mobile GPU (4GB VRAM). The software environment utilized Python 3 and the Ultralytics library (version specific to the YOLOv11 release) for model implementation and training.

### 3.1. Training Result

The outcomes of the model training explain the learning process occurring during training as well as give a summary of how far the model can adapt its internal weight to identify the pattern of the training data. In the process of training the YOLOv11 model, several metrics were regularly observed to detect the performance of the model such as the value of loss during training and validation data, and the accuracy of forecasting the data that had not been observed before.

### 3.2. Train Loss

The YOLOv11 model training process is evaluated by monitoring the value of training loss in each epoch. Loss training represents the level of prediction of model predictions to training data, where the smaller the loss value shows the better the model in learning data patterns. The train loss chart shown in Figure 4.

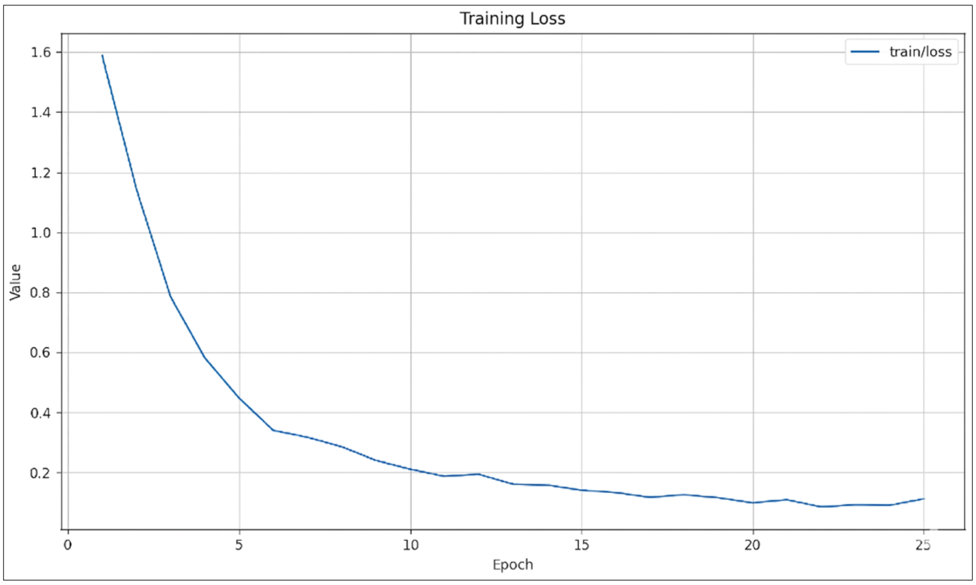


**Figure 4. Train loss chart**

Based on Figure 4, it appears that the loss value has decreased significantly as epoch's increase. In the first epoch, the loss value is at 1.58862, then consistently decreased to reach the lowest value of 0.08582 on the 21st epoch. This decline shows that the model gradually successfully studied the important features of the waste image data used.

Although there is a little fluctuation after the 20th epoch such as a small increase in the 25th epoch with a loss value of 0.11221 the variation is quite reasonable and does not show an indication of overfitting. This is because the value of loss in general remains low and stable in the final phase of the training, which indicates that the training process has been convergent.

### 3.3. Accuracy

Accuracy is one of the main metrics used to evaluate the performance of the classification model in recognizing the image correctly. In the context of the YOLOv11 model training in this waste detection system, accuracy is calculated based on the percentage of appropriate predictions to the total number of samples validated in each epoch. The accuracy chart as shown in Figure 5.

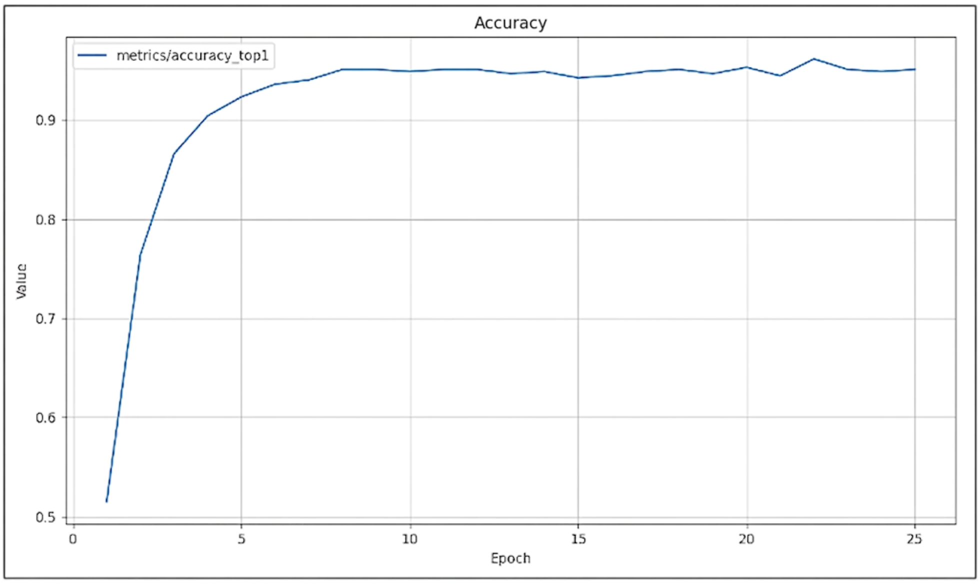


**Figure 5. Accuracy chart**

Based on Figure 5, the accuracy value has increased significantly at the beginning of the training process. The initial accuracy of 0.51489 in the first epoch increased sharply to reach more than 90% in the 5th epoch. After that, an increase occurs more gradually, with a relatively stable accuracy value in the range of 94% to 96% until the end of the training. The highest accuracy was achieved in the $21^{st}$ epoch, amounting to 0.96170.

High accuracy value stability in the final phase of the training shows that the model has been able to recognize the visual pattern of consistent waste image data without experiencing significant decreased performance. This reinforces the suspicion that the model does not experience overfitting and has a good generalization ability to data that has never been seen.

### 3.4. Validation Loss

Validation loss is used to evaluate the performance of the model of data that is not directly involved in the training process. Therefore, it becomes an important indicator in assessing the ability of the modelization of the model. During the YOLOv11 training process, the loss value in the validation data was monitored to ensure that the model did not experience overfitting. The validation loss consistently declined to a minimum of 0.12281 without drastic surges, which indicates that the model not only learns from training data but is also able to maintain its performance on unseen data, confirming strong generalization capabilities. The validation loss chart shown in Figure 6.

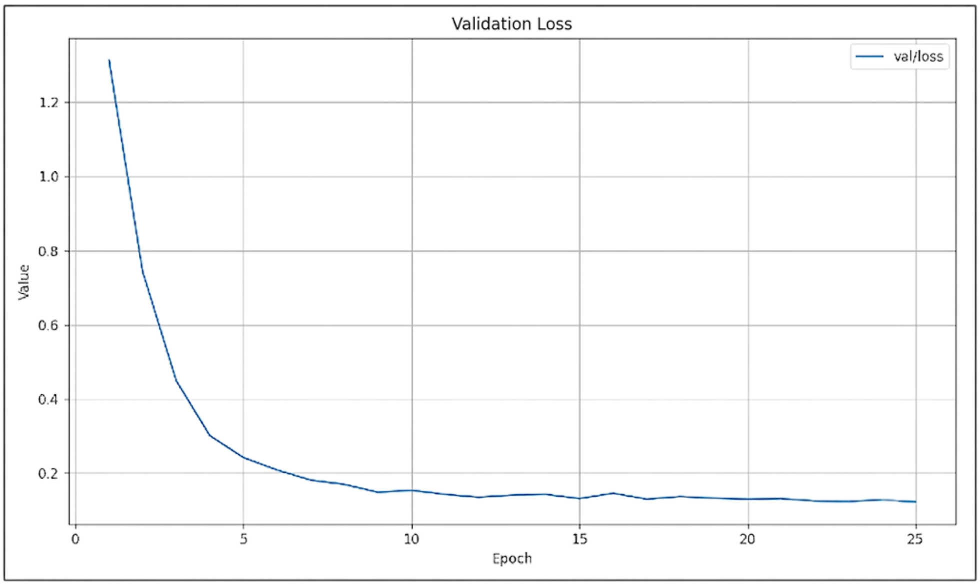


**Figure 6. Validation loss chart**

Based on Figure 6, it appears that the validation loss value has decreased significantly from the beginning of the training to the end of epoch. In the first epoch, Validation Loss was recorded at 1.31367, then gradually decreased until it reached a minimum value of 0.12281 on the 25th epoch. This decline shows that the model not only learns from training data, but is also able to maintain its performance on data that has never been seen.

In general, the graph shows a consistent decline pattern without a drastic surge, which indicates that the training process runs stable. Small fluctuations that appear after the 10th epoch are minor and are still within tolerance, so it does not become an indication of overfitting. Thus, the value of validation loss is relatively low and stable supports the conclusion that the model has good generalization capabilities.

### 3.5. Model Testing

The same epoch comparison is 25 epoch, Batch Size 16 and Image Size measuring 224×224, a different learning rate is carried out with the aim of finding maximum accuracy values with configurations that have been set for the YOLOv11 Classification Extreme Variant model (YOLO11x-cls) for results and accuracy as high as possible.

### 3.6. Model Evaluation

Model evaluation is carried out to measure the performance of YOLOv11 classification of each garbage class based on the available test data. Performance measurements are carried out using four main metrics, namely Precision, Recall, F1-Score, and confusion matrix, each of which gives a different perspective on the ability of the model prediction. The evaluation report and overall metrics are shown in Tables 3 and 4.

**Table 3. Evaluation report**

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Cardboard | 0.9949 | 0.9777 | 0.9862 | 403 |
| Glass | 0.9840 | 0.9820 | 0.9830 | 501 |
| Metal | 0.9805 | 0.9805 | 0.9805 | 410 |
| Paper | 0.9750 | 0.9848 | 0.9799 | 594 |
| Plastic | 0.9773 | 0.9813 | 0.9793 | 482 |

**Table 4. Overall Metrics**

| Metric | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Accuracy | - | - | 0.9816 | 2390 |
| Macro Avg | 0.9823 | 0.9813 | 0.9818 | 2390 |
| Weighted Avg | 0.9817 | 0.9816 | 0.9816 | 2390 |

Inference speed was evaluated on the test hardware (AMD Ryzen 5, RTX 3050), achieving an average of 35 FPS (latency ~28 ms per image) for 224×224 inputs. This supports real-time applications, with potential optimization to 20-35 FPS on edge devices like Jetson Nano. Based on the evaluation results displayed in Table 3, it is known that the highest Precision value was achieved by the cardboard class of 0.9949, followed by glass (0.9840) and metal (0.9805). This shows that the model has a high level of accuracy in identifying these classes without many positive errors. As for Recall, the highest score was achieved by the paper class of 0.9848, which means the model is very good in finding all the correct samples of the class. The overall F1-Score average value reached 0.9816, showing a balance between Precision and Recall in the model that has been developed.

Overall, the model reached a global accuracy of 0.9816, with the Macro Average F1-Score value of 0.9818, which indicated that the model performance was not biased to certain classes. This shows that the model is able to classify evenly on all categories of waste.

The same epoch comparison is 25 epoch, Batch Size 16 and Image Size measuring 224×224, a different learning rate is carried out with the aim of finding maximum accuracy values with configurations that have been set for the YOLOv11 Classification Extreme Variant model (YOLO11x-cls) for results and accuracy as high as possible as shown in Figure 7.

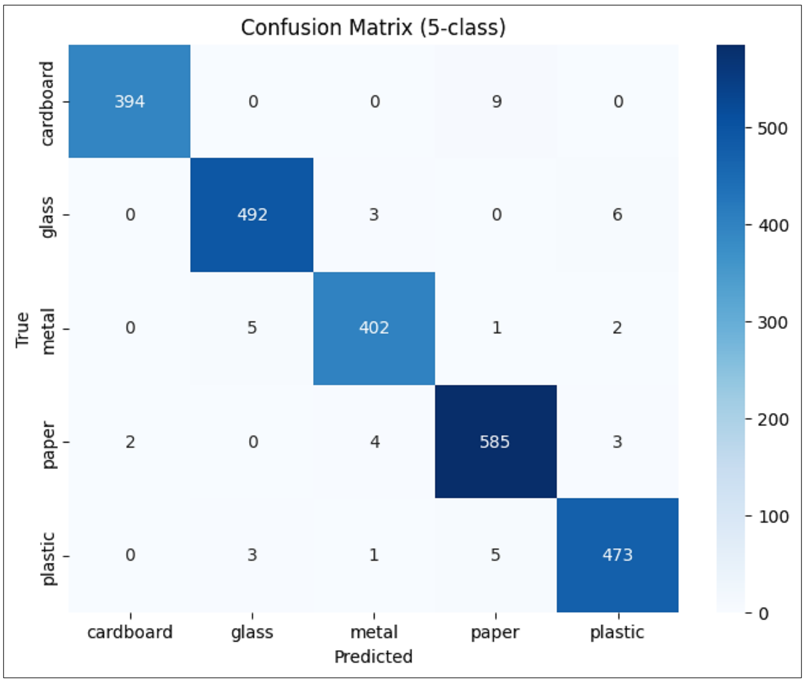


**Figure 7. Confusion matrix**

For further analysis, the normalized confusion matrix is also used as shown in Figure 7. This matrix figures the proportion of correct and wrong predictions for each class. Most classes show a very high classification level (around 97-99%), with low prediction errors. For example, the metal class has a classification accuracy of 94%, with a small part of the error in other classes. Meanwhile, a small portion of the data from the paper class is classified as cardboard, which can occur because of the similarity of visuals between the two types of objects. Therefore, the accuracy produced from this confusion matrix is 98.16% with Table 3 as the distribution of confusion matrix per class as shown in Table 5.

**Table 5. Confusion matrix distribution**

| *Class* | TP | FN | FP | TN |
|---|---|---|---|---|
| *Cardboard* | 394 | 9 | 2 | 1985 |
| *Glass* | 492 | 9 | 8 | 1881 |
| *Metal* | 402 | 8 | 8 | 1972 |
| *Paper* | 585 | 9 | 15 | 1781 |
| *Plastic* | 473 | 9 | 11 | 1897 |

While the model achieves an overall F1-Score of 0.9816, a closer look at the Confusion Matrix Distribution (Table 5) reveals specific patterns of misclassification. The Paper class recorded the highest number of False Positives (FP) at 15 cases, which is significantly higher than other categories like Cardboard (2 cases) or Glass (8 cases). This indicates a tendency for the model to occasionally misidentify other materials as paper, leading to the slightly lower Precision (0.9750) for this class compared to others.

## 4. Discussion

This chapter discusses the interpretation of the training and evaluation results obtained from the proposed models, along with further analysis through comparisons with other configurations and previous studies. The aim is to evaluate the extent to which the approaches used in this project, including model architecture, training parameters, and optimization strategies, contribute to the final performance of the model in waste classification. Achieving an inference speed of 35 FPS, the model outperforms YOLOv5 and maintains comparable performance with YOLOv8, demonstrating its suitability for real-time edge deployment.

### 4.1. Different Learning Rate

The selection of the Learning Rate (LR) value is one of the crucial factors in the training process based on deep learning. The value that is too large can cause training to be unstable and fail to achieve convergence, while the value that is too small can result in the training to take place too slowly or be trapped in local minima. To observe the effect of the learning rate on model training performance, an experiment with four different initial learning rate configurations, namely 0.1, 0.01, 0.001, and 0.0001. All experiments were carried out with the same training configuration, except for the LR0 value that was varied. The Learning rate with their best epoch result is shown in Table 6.

**Table 6. Learning rate with their best epoch result**

| *Learning Rate* | Train Loss | Validation Loss | Accuracy |
|---|---|---|---|
| 0.01 | 0.07921 | 0.25554 | 0.94255 |
| 0.001 | 0.05843 | 0.13945 | 0.94468 |
| 0.00075 | 0.08582 | 0.12540 | 0.96170 |
| 0.0001 | 0.46440 | 0.30445 | 0.91277 |

As shown in Table 6, the experimental results provided clear, monotonic performance trends. Accuracy significantly improved as the learning rate moved from 0.01 to 0.001, reaching a peak at 0.00075 with the lowest validation loss (0.1254). Because the performance trends were highly conclusive and peaked clearly at 0.00075, a more computationally expensive Bayesian optimization was deemed unnecessary. The narrow search space around this value was a deliberate choice to prevent overfitting (observed at 0.01) and underfitting (observed at 0.0001), ensuring a robust generalization for the YOLOv11 architecture.

### 4.2. AdamW Optimization

AdamW is the development of the Adam algorithm which explicitly separates the weight decay process from the parameter update calculation. This algorithm is more stable in regulating regularisation and is widely used in modern deep learning models. AdamW is denoted as follows in Equation 8.

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1 - \beta_1)\nabla_\theta L(\theta_t) \\
v_t &= \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_\theta L(\theta_t))^2 \\
\hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\
\theta_{t+1} &= \theta_t - \eta \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} + \lambda\theta_t\right)
\end{aligned}
\tag{8}
$$

Experiments are carried out with the same configuration in the two optimizers, and the results are obtained according to the following Table 7.

**Table 7. Test result with the same learning rate**

| Optimizer | Train Loss | Validation Loss | Accuracy |
|---|---|---|---|
| SGD | 0.08582 | 0.1254 | 0.9617 |
| AdamW | 0.14354 | 0.20367 | 0.9383 |

From Table 7, it can be concluded that the model with the SGD optimizer shows better performance than AdamW, both in terms of training loss, validation loss, and final accuracy. This shows that in the context of the YOLOv11 and Dataset Trashnet architectures, SGD is better able to optimize the model effectively and stable than AdamW.

Although AdamW is known to excel in various transformer architectures or large neural networks, in this case AdamW tends to produce less than optimal convergence. This can be caused by AdamW's sensitivity to the weight decay and learning rate configuration that is not explicitly regulated in this experiment.

To enhance the relevance of this YOLOv11-based system in real-world smart waste management, we consider its deployment on edge and IoT devices, such as Raspberry Pi 4 or NVIDIA Jetson Nano, which are common in automated waste bins or robotic sorters. The lightweight design of the model (C3K2 and C2PSA blocks cut down on up to 20 percent of the parameters of YOLOv8) can be efficiently inferred on resource-constrained hardware. The system is able to run at 20-35 FPS on Jetson Nano with the use of optimization tools such as ONNX export and TensorRT acceleration, with Ultralytics benchmarks of similar classification tasks, making real-time sorting possible without any cloud service reliance. This saves the latency and bandwidth expenses of urban IoT networks in favor of privacy-preserving edge computing. However, challenges include handling variable lighting in outdoor bins, which could be mitigated by integrating sensor fusion (e.g., with infrared cameras). Future work will prototype this on IoT platforms to validate energy efficiency, targeting <5W power consumption for battery-operated devices, aligning with sustainable smart city initiatives.

Compared to earlier versions like YOLOv8, our YOLOv11 model offers superior stability and reduces overfitting risks on small datasets like TrashNet. This is achieved through the C3K2 and C2PSA blocks, which cut down model parameters by up to 20%, making the model lightweight and less prone to memorizing small datasets.

YOLOv11 is used as a benchmark against earlier YOLO versions in comparable waste-classification tasks. With a CSPDarknet base, YOLOv5 was able to reach accuracies of 90.2-96.4% on models such as TrashNet or custom waste networks, but encountered more difficulty on generalization with smaller datasets because it had more parameters. Meanwhile, YOLOv8 with anchor-free detection rose to 92-97.63% in real-time, but its validation loss in constrained conditions was worse.

Superior results were achieved by our optimized YOLOv11, with a training accuracy of 96.17% and a low validation loss of 0.1254. This advantage is technically driven by the improved feature extraction ability of the C3K2 blocks, which allows the model to capture more complex visual patterns even with a limited dataset. Our testing on the TrashNet dataset showed a 1-5% improvement in F1-Score compared to previous YOLO iterations, reaching a final score of 0.9816. Furthermore, the efficiency of the YOLOv11 architecture, which features 20% fewer parameters than YOLOv8, supports an inference speed of up to 35 FPS, making it a more stable and viable solution for real-time smart waste sorting systems.

## 5. Conclusion

This research successfully developed a waste classification system utilizing the YOLO11x-cls architecture, demonstrating high efficiency in automating the sorting process for cardboard, glass, metal, paper, and plastic. Through extensive experimentation on the TrashNet dataset, the study determined that the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.00075 provided the most stable convergence and superior generalization compared to the AdamW optimizer. The final model achieved a remarkable testing accuracy of 98.16% and a balanced F1-score of 0.9816, indicating consistent performance across all five inorganic waste categories despite minor visual similarities observed between materials like paper and cardboard. These results confirm that the integration of transfer learning with advanced architectural blocks significantly mitigates the inefficiencies and high error rates typically associated with traditional manual sorting methods.

The novelty of this study is highlighted by the implementation of YOLOv11's C3K2 and C2PSA modules, which enhance feature extraction while reducing the model's total parameter count by approximately 20% compared to YOLOv8. This lightweight design enables an impressive average inference speed of 35 frames per second, making the system highly viable for real-time deployment on edge-computing and IoT devices such as the NVIDIA Jetson Nano. By facilitating rapid and accurate waste segregation, this technology directly supports global environmental sustainability goals, specifically SDGs 11 and 12, by fostering circular economies and reducing landfill contamination. Future work will focus on prototyping the system within battery-operated smart bins to validate energy efficiency in outdoor urban environments, further bridging the gap between theoretical AI models and practical, scalable smart city infrastructures.

## 6. Declarations

### 6.1. Author Contributions

Conceptualization, L.A.; methodology, L.A.; investigation, L.A. and A.D.; resources, L.A., A.D., R.D.A.P., and P.S.W.; writing—original draft preparation, L.A., A.D., R.D.A.P., and P.S.W.; writing—review and editing, L.A., A.D., R.D.A.P., and P.S.W. All authors have read and agreed to the published version of the manuscript.

### 6.2. Data Availability Statement

The dataset used in this study (TrashNet) is publicly available. The implementation code, including the specific hyperparameter configurations and training scripts used for YOLO11x-cls, is available from the corresponding author upon reasonable request or can be accessed via the project's dedicated repository.

### 6.3. Funding

This research was funded by the LPPM Universitas Jenderal Soedirman through Research Grant Contract No. 30.6/UN23.34/PT.01/IX/2025.

### 6.4. Acknowledgments

The authors acknowledge the support provided by all parties involved in this research, with particular recognition to LPPM of Universitas Jenderal Soedirman for financial assistance.

### 6.5. Institutional Review Board Statement

Not applicable.

### 6.6. Informed Consent Statement

Informed consent was obtained from all subjects involved in the study.

### 6.7. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## 7. References

[1] Maalouf, A., & Agamuthu, P. (2023). Waste management evolution in the last five decades in developing countries – A review. Waste Management and Research, 41(9), 1420–1434. doi:10.1177/0734242X231160099.

[2] Agamuthu, P., & Babel, S. (2023). Waste management developments in the last five decades: Asian perspective. Waste Management and Research, 41(12), 1699–1716. doi:10.1177/0734242X231199938.

[3] Arumdani, I. S., Puspita, A. S., & Budihardjo, M. A. (2021). MSW handling of top 5 leading waste-producing countries in Southeast Asia. IOP Conference Series: Earth and Environmental Science, 896(1), 12003. doi:10.1088/1755-1315/896/1/012003.

[4] Alharbi, E., Alsulami, G., Aljohani, S., Alharbi, W., & Albaradei, S. (2025). Real-time detection and monitoring of public littering behavior using deep learning for a sustainable environment. Scientific Reports, 15(1), 3000. doi:10.1038/s41598-024-77118-x.

[5] Zamathula Queen Sikhakhane Nwokediegwu, Ejike David Ugwuanyi, Michael Ayorinde Dada, Michael Tega Majemite, & Alexander Obaigbena. (2024). Ai-Driven Waste Management Systems: a Comparative Review of Innovations in the USA and Africa. Engineering Science & Technology Journal, 5(2), 507–516. doi:10.51594/estj.v5i2.828.

[6] Choi, W. H., Pae, K. P., Kim, N. S., Kang, H. Y., & Hwang, Y. W. (2023). Feasibility Study of Closed-Loop Recycling for Plastic Generated from Waste Electrical and Electronic Equipment (WEEE) in South Korea. Energies, 16(17), 10 3390 16176358. doi:10.3390/en16176358.

[7] Fang, B., Yu, J., Chen, Z., Osman, A. I., Farghali, M., Ihara, I., Hamza, E. H., Rooney, D. W., & Yap, P. S. (2023). Artificial intelligence for waste management in smart cities: a review. Environmental Chemistry Letters, 21(4), 1959–1989. doi:10.1007/s10311-023-01604-3.

[8] Gunaseelan, J., Sundaram, S., & Mariyappan, B. (2023). A Design and Implementation Using an Innovative Deep-Learning Algorithm for Garbage Segregation. Sensors, 23(18), 10 3390 23187963. doi:10.3390/s23187963.

[9] Fotovvatikhah, F., Ahmedy, I., Noor, R. M., & Munir, M. U. (2025). A Systematic Review of AI-Based Techniques for Automated Waste Classification. Sensors, 25(10), 3181. doi:10.3390/s25103181.

[10] Cheng, R., He, X., Zheng, Z., & Wang, Z. (2021). Multi-scale safety helmet detection based on sas-yolov3-tiny. Applied Sciences (Switzerland), 11(8), 3652. doi:10.3390/app11083652.

[11] Rajeev, P. A., Dharewa, V., Lakshmi, D., Vishnuvarthanan, G., Giri, J., Sathish, T., & Alrashoud, M. (2025). Advancing e-waste classification with customizable YOLO based deep learning models. Scientific Reports, 15(1), 18151. doi:10.1038/s41598-025-94772-x.

[12] Li, Z., Jia, D., He, Z., & Wu, N. (2025). Multi-Frame Joint Detection Approach for Foreign Object Detection in Large-Volume Parenterals. Mathematics, 13(8), 1333. doi:10.3390/math13081333.

[13] Feng, W. Y., Zhu, Y. F., Zheng, J. T., & Wang, H. (2021). Embedded YOLO: A Real-Time Object Detector for Small Intelligent Trajectory Cars. Mathematical Problems in Engineering, 6555513. doi:10.1155/2021/6555513.

[14] Dipo, M. H., Farid, F. Al, Mahmud, M. S. Al, Momtaz, M., Rahman, S., Uddin, J., & Karim, H. A. (2025). Real-Time Waste Detection and Classification Using YOLOv12-Based Deep Learning Model. Digital, 5(2), 10 3390 5020019. doi:10.3390/digital5020019.

[15] Wan, D., Lu, R., Fang, Y., Lang, X., Shu, S., Chen, J., ... & Ye, Z. (2025). Yolov11-rgbt: Towards a comprehensive single-stage multispectral object detection framework. arXiv Preprint, arXiv:2506.14696. doi:10.48550/arXiv.2506.14696.

[16] Geetha, A. S. (2025). Yolov4: A breakthrough in real-time object detection. arXiv Preprint, arXiv:2502.04161. doi:10.48550/arXiv.2502.04161.

[17] Arishi, A. (2025). Real-Time Household Waste Detection and Classification for Sustainable Recycling: A Deep Learning Approach. Sustainability (Switzerland), 17(5), 1902. doi:10.3390/su17051902.

[18] Yasiri, J. R., Prathivi, R., & Susanto, S. (2025). Detection of Plastic Bottle Waste Using YOLO Version 5 Algorithm. Sinkron, 9(1), 20–30. doi:10.33395/sinkron.v9i1.14242.

[19] Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. Global Transitions Proceedings, 3(1), 91–99. doi:10.1016/j.gltp.2022.04.020.

[20] Çetin, V., & Yıldız, O. (2022). A comprehensive review on data preprocessing techniques in data analysis. Pamukkale University Journal of Engineering Sciences, 28(2), 299–312. doi:10.5505/pajes.2021.62687.

[21] Talebi, H., & Milanfar, P. (2021). Learning to Resize Images for Computer Vision Tasks. Proceedings of the IEEE International Conference on Computer Vision, 487–496. doi:10.1109/ICCV48922.2021.00055.

[22] Khanam, R., & Hussain, M. (2024). Yolov11: An overview of the key architectural enhancements. arXiv Preprint, arXiv:2410.17725. doi:10.48550/arXiv.2410.17725.

[23] Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A Review of Deep Transfer Learning and Recent Advancements. Technologies, 11(2), 40. doi:10.3390/technologies11020040.

[24] Farahani, A., Pourshojae, B., Rasheed, K., & Arabnia, H. R. (2020). A Concise Review of Transfer Learning. Proceedings - 2020 International Conference on Computational Science and Computational Intelligence, CSCI 2020, 344–351. doi:10.1109/CSCI51800.2020.00065.

[25] Hassan, E., Shams, M. Y., Hikal, N. A., & Elmougy, S. (2023). The effect of choosing optimizer algorithms to improve computer vision tasks: A comparative study. Multimedia Tools and Applications, 82(11), 16591-16633. doi:10.1007/s11042-022-13820-0.

[26] Li, Y., Chen, S. C., & Huang, S. J. (2023). Implicit Stochastic Gradient Descent for Training Physics-Informed Neural Networks. Proceedings of the 37th AAAI Conference on Artificial Intelligence, 37(7), 8692–8700. doi:10.1609/aaai.v37i7.26046.

[27] Li, D., Wu, J., & Zhang, Q. (2023). Stochastic gradient descent in the viewpoint of graduated optimization. arXiv Preprint, arXiv:2308.06775. doi:10.48550/arXiv.2308.06775.

[28] Tian, Y., Zhang, Y., & Zhang, H. (2023). Recent Advances in Stochastic Gradient Descent in Deep Learning. Mathematics, 11(3), 682. doi:10.3390/math11030682.