Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 2, April 2026, pp. 1429~1441
ISSN: 2302-9285, DOI: 10.11591/eei.v15i2.10540

Comparative analysis of ResNet backbones in single shot detector for visual-based waste detection

Zahra Khalila Salsabila, Nurcahya Pradana Taufik Prakisya, Febri Liantoni
Department of Informatics Education, Faculty of Teacher and Training Education, Universitas Sebelas Maret, Surakarta, Indonesia

Article Info

Article history:

Received Apr 21, 2025
Revised Jan 18, 2026
Accepted Mar 10, 2026

Keywords:

Deep learning
Mean average precision
Residual network
Single shot detector
Waste detection

ABSTRACT

Waste has become a serious environmental issue that requires effective and efficient management systems. This study compares three residual network (ResNet) variants (ResNet-34, ResNet-50, and ResNet-101) within the single shot detector (SSD) framework for visual waste detection. The dataset consists of 800 images in four categories—food, plastic, paper, and wood—with a 70:20:10 split for training, validation, and testing. The backbone architecture, optimizer (stochastic gradient descent (SGD) and Adam), and learning rate are varied to evaluate fifteen experimental configurations. Model performance is assessed using precision, recall, F1-score, and mean average precision (mAP). The results show that SSD–ResNet-34 with SGD and a learning rate of 0.0005 works best, with a mAP of 91.02%, which is better than deeper backbones. Deeper backbone architectures do not consistently improve accuracy; instead, they increase the risk of overfitting on small datasets. These findings highlight that lightweight architecture, when used with the right hyperparameter settings, strikes a better balance between accuracy, computational efficiency, and generalization for small-scale waste detection tasks.

Corresponding Author:

Nurcahya Pardana Taufik Prakisya
Department of Informatics Education, Faculty of Teacher and Training Education
Universitas Sebelas Maret
St. Ir. Sutami 36A, Kentingan, Jebres, Surakarta 57126, Indonesia
Email: nurcahya.ptp@staff.uns.ac.id

1. INTRODUCTION

Waste is a serious environmental problem faced worldwide. Population growth, rapid urbanization [1], and economic development [2] have led to a significant increase in waste volume. The World Bank reported that there were 2.01 billion tons of global waste in 2016, and this figure is expected to increase to 3.4 billion tons by 2050 [3]. In Indonesia alone, total waste production in 2024 reached 32.66 million tons [4]. This amount, if not managed properly, can cause serious environmental impacts, such as groundwater and air pollution, land degradation, increased incidence of cancer, infant mortality, and birth defects [5]–[8].

One solution to overcome the problem of waste accumulation is through proper management. Historically, waste management was carried out manually, where workers collected and disposed of waste at predetermined locations [9]. However, this method is labor-intensive, inefficient, and error-prone, thus requiring an automated and intelligent approach [10]. Along with technological advances, artificial intelligence (AI) is starting to be considered as an alternative solution to improve waste management efficiency. AI has proven effective in waste classification and detection [11]. In the realm of object detection, there are two-stage algorithms, such as Faster region-based convolutional neural network (Faster R-CNN) [12] as well as single-stage algorithms, such as you only look once (YOLO) [13] and single shot detector

(SSD) [14]. Among the three, SSD stands out for its balance between accuracy and computational efficiency [15], making it a promising choice for visual-based waste detection.

Recent studies have used various backbone networks in SSD, such as VGG, MobileNet, and residual network (ResNet), to improve detection performance. For example, Meng *et al.* [16] demonstrated SSD using MobileNet with FPN as feature extraction to classify waste. The study achieved a mean average precision (mAP) of 93.63% and a speed of 102 FPS. Another study conducted by Haldorai *et al.* [17] used an improved SSD model for an intelligent vision-based water waste cleaning robot achieving an accuracy of 94%. A study by Fang [18] presented an improved algorithm based on SSD-based lightweight for recyclable waste detection, which optimized the model's required features and reduced environmental noise in the detection model, achieving an accuracy of 90.18%. Karthikeyan *et al.* [19] used an SSD algorithm with augmented NMS classifier to detect biodegradable and non-biodegradable waste in real time and achieved an mAP of 96.5%. A study conducted by Lee *et al.* [20] used SSD with AlexNet architecture to detect plastic bottles and beverage waste packaged in aluminum foil, resulting in an accuracy of 95%. When applying SSD algorithms to litter detection, selecting the right architecture plays a crucial role in improving the accuracy and performance of the model in recognizing different types of litter. ResNet, introduced by He *et al.* [21], has been shown to overcome the performance degradation of deeper CNNs through shortcut connections. Commonly used ResNet architecture variants include ResNet-34, ResNet-50, and ResNet-101, which offer varying levels of complexity and have been widely used as backbones in object detection and segmentation tasks [15], [22].

Although SSD has been used in several waste detection studies, comparative research specifically evaluating the performance of different backbone architectures within the SSD framework is limited. Most existing research focuses on a single model architecture, resulting in a lack of systematic understanding of how backbone depth affects performance, especially on small-scale datasets.

On the other hand, several recent studies have explored transformer- and YOLO-based approaches for waste detection, which demonstrate high accuracy but also come with higher computational demands. For example, Wang *et al.* [23] used Swin Transformer for multi-category plastic waste sorting and achieved mAP above 91% under complex lighting and background conditions. Huang *et al.* [24] implemented Vision Transformer on a portable device for real-time waste classification, achieving 96.98% accuracy on the TrashNet dataset. Ji *et al.* [25] proposed a multimodal Swin Transformer that combines RGB and spectral features to identify plastic types with better durability. While these transformer-based models provide superior accuracy, they often require high-end GPUs and extensive datasets, limiting their practicality for embedded or edge computing applications. Consequently, CNN-based frameworks such as SSD remain attractive due to their efficiency and lower computational cost, especially when optimized through backbone and parameter tuning.

Therefore, this study aims to fill this research gap by systematically comparing three ResNet backbone variants (ResNet-34, ResNet-50, and ResNet-101) within the SSD framework. All models were trained with the same experimental configuration, with variations in optimizer and learning rate, and evaluated using precision, recall, F1-score, and mAP. This comparative analysis is expected to provide practical insights into the trade-off between model accuracy and complexity, while contributing to the development of efficient and sustainable deep learning-based waste detection systems.

2. METHOD

This study uses a research and development (RnD) approach by applying the SSD algorithm with a ResNet backbone architecture to detect and classify waste objects in digital images. The goal is to identify the most effective ResNet backbone architecture on SSD in the task of waste detection. The research workflow consists of several stages: i) collecting a trash image dataset, ii) data preparation, iii) model training, and iv) model evaluation using performance metrics. The flow in this study can be seen in Figure 1.

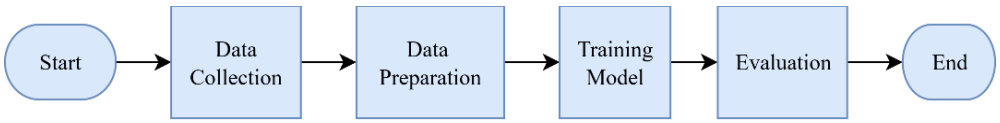

Figure 1. Research flow

2.1. Data collection

The dataset used in this study consists of 800 labeled digital images representing four waste categories: food waste, plastic, paper, and wood, with 200 images in each category. The selection of these four categories is based on data from the Ministry of Environment and Forestry of the Republic of Indonesia

in 2024, which shows that these four types of waste constitute the largest composition of waste production in Indonesia [26].

The dataset was obtained from several open-source sources available on Kaggle [27]–[29] and GitHub [30]. All images are in JPG format with varying resolutions. To maintain consistency, simple pre-processing was performed by resizing the images to suit the model training needs. The annotation process was performed manually using the Roboflow website, then the annotation results were converted to the XML Pascal VOC format to support model training with the PyTorch framework. Figure 2 shows sample data for each class. As can be seen, the dataset contains samples of food, paper, plastic, and wood waste.

Figure 2. Example of the waste dataset

2.2. Data preparation

The waste dataset underwent a series of preprocessing, labeling, and validation steps before being used for model training. At this stage, the collected data was resolution-adjusted to ensure size consistency, labeled according to their respective classes, and verified for missing values. The dataset was then divided into three subsets—training, validation, and testing—in a 70:20:10 ratio.

2.2.1. Data preprocessing

In the data preprocessing stage, each image is resized to standardize the input dimensions. Resizing aims to ensure all images have a uniform size and are compatible with the model architecture, while reducing computational complexity without losing important features. In this study, all images were resized to 256×256 pixels, which provides a balance between visual clarity and processing efficiency.

2.2.2. Object annotation

Object labeling or annotation is the activity of assigning labels, categories, or information to data by creating bounding boxes so that the data can serve as a reference (ground truth) in the model training process. The goal is to train the model to be able to recognize objects to be predicted. The classes defined in this study include four categories: "food," "paper," "plastic," and "wood." Data labeling in this study was carried out manually using the web-based software, Roboflow (https://roboflow.com/). The annotation process is carried out by drawing bounding boxes around relevant waste objects and classifying them into one of four predefined classes.

After all the data was annotated, it was divided into three parts with a proportion of 70% for the training set, 20% for the validation set, and 10% for the test set. Figure 3 shows a graph of the distribution of each trash class in the three subsets. In the training set, the four classes have a relatively balanced amount of data, namely around 130 to 155 images per class. A similar distribution is also seen in the validation set with a range of 35 to 45 images per class, and in the test set with a range of 10 to 25 images per class. This indicates that the data distribution was done proportionally, so there is no imbalance in the amount of data between classes in each subset.

The completed dataset was labeled and divided into sets, then exported in Pascal VOC XML format. Object labeling generated an .xml file containing information about object classes and bounding box coordinates. The object labeling process resulted in one object class per image with 800 objects.

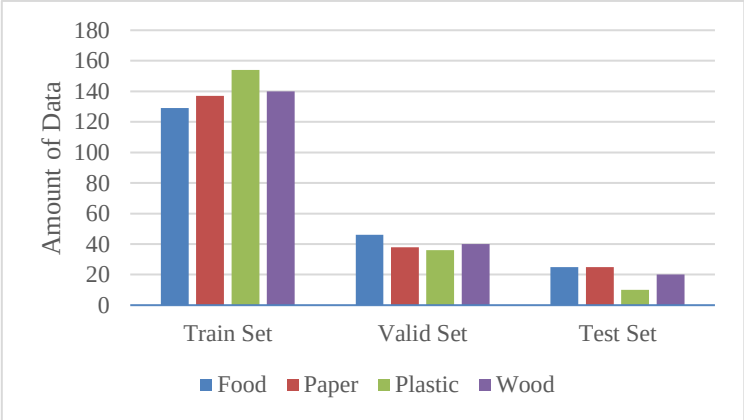

Figure 3. Distribution of number of image data for each type of waste based on dataset division

2.3. Training model

The SSD detector, introduced by Liu *et al.* [14], is a single-stage object detection framework that integrates localization and classification in a single forward pass. The original SSD is built on a VGG-16 network, which consists of multiple convolutional layers followed by a fully connected layer with ReLU activation. To adapt the VGG for detection tasks, the last fully connected layer is replaced with a convolutional layer, and additional convolutional layers are introduced to extract multi-scale feature maps at increasingly smaller resolutions. This design transforms the SSD into a fully convolutional network capable of processing images of various sizes, making it more flexible than conventional fixed-size models [15]. As illustrated in Figure 4, low-level feature maps are effective in detecting small objects, while high-level feature maps capture larger objects, enabling multi-scale detection.

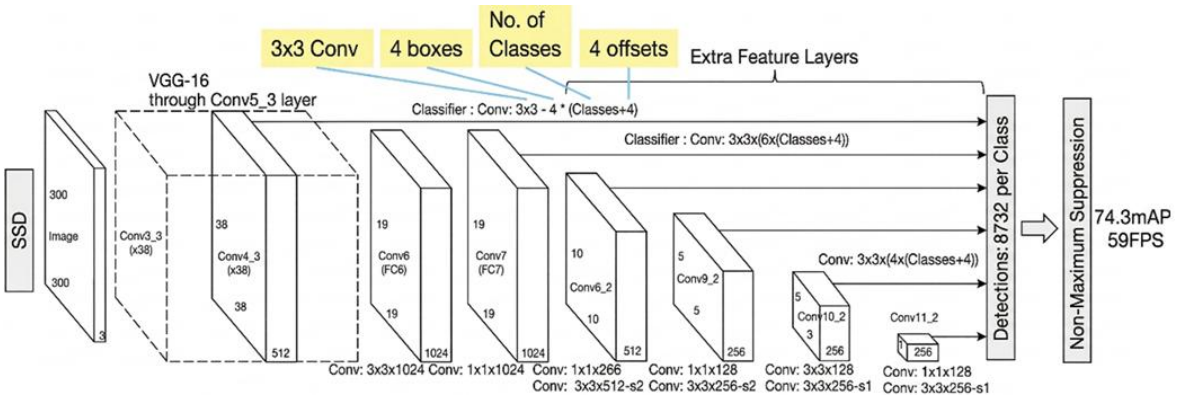

Figure 4. SSD architecture

To achieve robust detection, SSD uses default anchor boxes with varying aspect ratios at each feature map location and matches them to ground truth objects using the highest Jaccard overlap. During training, the network optimizes a weighted combination of localization and confidence losses, while strategies such as hard negative mining and extensive data augmentation further improve accuracy [31]. Despite offering superior speed compared to Faster R-CNN and higher accuracy than YOLO, SSD initially struggled to detect small objects due to the limited representation power of its VGG-based backbone. This limitation highlights a common problem in deep convolutional neural networks: as depth increases, traditional architectures can experience degradation, where adding layers no longer improves and can even reduce accuracy. Empirical evidence shows that the test risk forms a U-shaped curve with increasing depth in CNNs [32] and that excessive depth can degrade accuracy even in ResNets for remote sensing tasks [33].

To address these challenges, more sophisticated backbones such as ResNet were later integrated into SSD. ResNet residual connections mitigate the degradation problem by allowing deeper networks to converge effectively, thereby improving detection performance, particularly for small and complex

objects [34]. This evolution from VGG to ResNet marks a significant step in improving SSD's balance between detection speed, accuracy, and scalability.

2.3.1. Residual network architecture as backbone

In SSD architectures, backbone selection plays a crucial role in determining the quality of the resulting feature representations. Traditionally, SSDs use VGG-16 as their backbone, but the limitations of this architecture in network depth and computational efficiency have prompted the development of alternative architectures [14]. ResNet has become one of the most widely adopted backbones due to its ability to address the vanishing gradient problem, enabling deeper network training with more stable performance [21]. The use of ResNet as an SSD backbone improves the quality of the resulting feature maps, supporting the detection of objects with a greater variety of sizes and complexities compared to conventional backbones [35]. ResNet also improves deep network training with shortcuts that facilitate gradient flow, resulting in faster training and significant improvements in depth accuracy [36], [37].

There are several variations of the ResNet architecture, such as ResNet-34, ResNet-50, and ResNet-101, which are distinguished by the level of depth and complexity used. Aydemir *et al.* [38] stated that ResNet variations are named based on the number of layers, for example, ResNet-34, ResNet-50, and ResNet-101 have 34, 50, and 101 layers, respectively. These layers include convolutional layers, pooling layers, fully connected layers, and skip connections that form the network. Figure 5 shows the architecture of ResNet-34, ResNet-50, and ResNet-101.

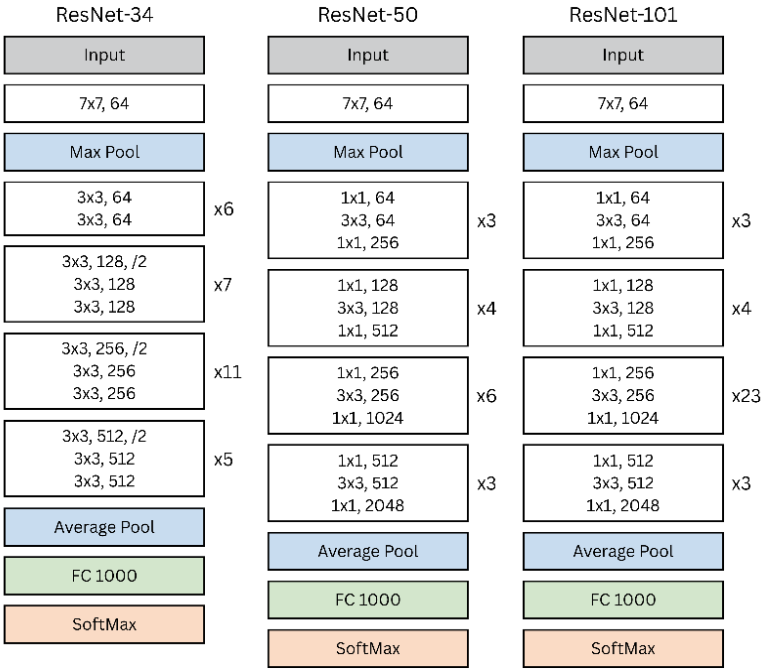

Figure 5. ResNet-34, ResNet-50, and ResNet-101 architecture

The application of the ResNet architecture to digital images has been used in various studies. For example, a study conducted by Gao *et al.* [39] performed node detection using transfer learning with ResNet 34, showing an overall accuracy of 98.69%. Another study conducted by Rajpal *et al.* [40] detected COVID-19 from chest X-ray images using features selected with ResNet-50, resulting in an overall classification accuracy of 0.974±0.02 and sensitivities of 0.987±0.05, 0.963±0.05, and 0.973±0.04 at 95% confidence intervals for the COVID-19, normal, and pneumonia classes, respectively. A study conducted by Ma'rifah *et al.* [41] compared Faster R-CNN with ResNet-50 and ResNet-101 for detecting recyclable waste, resulting in average F1-scores of 63% and 77%.

2.3.2. Training configuration

All experiments were conducted using Google Colab with GPU acceleration and implemented in PyTorch. The SSD model was trained for 75 epochs with a batch size of 16. To assess the impact of optimization strategies, two optimizers—stochastic gradient descent (SGD) with momentum and Adam—were tested with learning rates of 0.01, 0.001, 0.0005, and 0.0001. The MultiStepLR scheduler was used to

dynamically adjust the learning rate during training, thus facilitating stable convergence. The training objective combined the Softmax cross entropy loss for multi-class classification and the Smooth L1 loss for bounding box regression. In total, 15 model configurations were evaluated by integrating three ResNet backbones (ResNet-34, ResNet-50, and ResNet-101) with different optimizer–learning rate pairs.

To improve model generalization across various visual conditions, extensive data augmentation was applied using the TensorV2 library, including horizontal flip, blur, motion blur, median blur, brightness/contrast adjustment, color jitter, and random gamma transform. These augmentations were applied to all 800 images in the dataset, which comprise four waste categories: food, plastic, paper, and wood. In addition to the artificial augmentations, the dataset also exhibits inherent diversity in lighting conditions (indoor, outdoor, and low-light environments) and background settings (plain, textured, and cluttered). Table 1 illustrates the architectural integration used in this study, combining the SSD framework with a ResNet backbone. Meanwhile, Table 2 summarizes the complete training setup, including optimizer type, learning rate, loss function, scheduler settings, and augmentation strategy.

Table 1. Experimental configuration of model, optimizer, and learning rate combinations

Architecture SSD+ResNet	Optimizer	Learning rate
SSD+ResNet-34	SGD	0.01
		0.001
		0.0001
		0.0005
	Adam	0.001
SSD+ResNet-50	SGD	0.01
		0.001
		0.0001
		0.0005
	Adam	0.001
SSD+ResNet-101	SGD	0.01
		0.001
		0.0001
		0.0005
	Adam	0.001

Table 2. Training parameter details

Description	Detail
Epoch	75
Batch size	16
Optimizer	SGD with momentum, Adam
Initial learning rate	0.01, 0.001, 0.0001, and 0.0005
Scheduler	MultiStepLR
Loss	Softmax cross entropy
Augmentation	Horizontal flip, blur, motion blur, median blur, to glar, random brightness contrast, color jitter, random gamma, and TensorV2
# of input images	800

2.4. Evaluation

To evaluate the performance of the object detection model, this study uses several widely known metrics: mAP, precision, recall, and F1-score. mAP is a standard metric in object detection, which reflects the average precision across all classes and captures the model's precision-recall trade-off [42]. Meanwhile, precision, recall, and F1-score are obtained from the confusion matrix, which is constructed based on the model's predictions across four variables: true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

In classification, TP refers to a positive instance that is correctly predicted as positive, while TN refers to a negative instance that is correctly predicted as negative. Conversely, FP occurs when a negative instance is incorrectly classified as positive, and FN occurs when a positive instance is incorrectly classified as negative [43].

2.4.1. Mean average precision

mAP is one of the most commonly used metrics to assess the performance of object detectors [44]. mAP is obtained by taking the average value of the average precision APi across all n evaluated samples [43]. mAP can be calculated using (1):

$$mAP = \frac{1}{N}\sum_{k=1}^{k=n} AP_k \times 100\% \quad (1)$$

2.4.2. Precision

Precision is useful for calculating the performance analysis of a method by validating the method's accurate positive predictions [45]. As shown in (2), precision is the ratio of correctly predicted positive cases divided by the total positive class.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \times 100\% \quad (2)$$

2.4.3. Recall

Recall, another name for sensitivity, measures how well a model can detect positive cases. It is calculated by dividing the total number of actual positive cases by the number of accurately predicted positive cases [46]. Recall can be calculated using (3):

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \times 100\% \quad (3)$$

2.4.4. F1-score

The F1-score is a performance metric that combines precision and recall by calculating their harmonic mean, thus requiring consistently high precision and recall values to achieve a high F1-score [47]. The F1-score can be calculated using (4):

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \quad (4)$$

3. RESULTS AND DISCUSSION

3.1. Results

3.1.1. Training model

The SSD model was trained 15 times with various backbone architectures, optimizer types, and learning rates. Each model was trained for 75 epochs, and the training results were used for the testing and evaluation phase. During the training process, the training loss was monitored as a key indicator of the model's quality in predicting the training data. The total loss in SSD is a combination of the classification loss and the bounding box regression loss (localization loss). The development of the training loss value during the training process is shown in Figure 6, which illustrates a comparison of model performance based on the architecture configuration and parameters used.

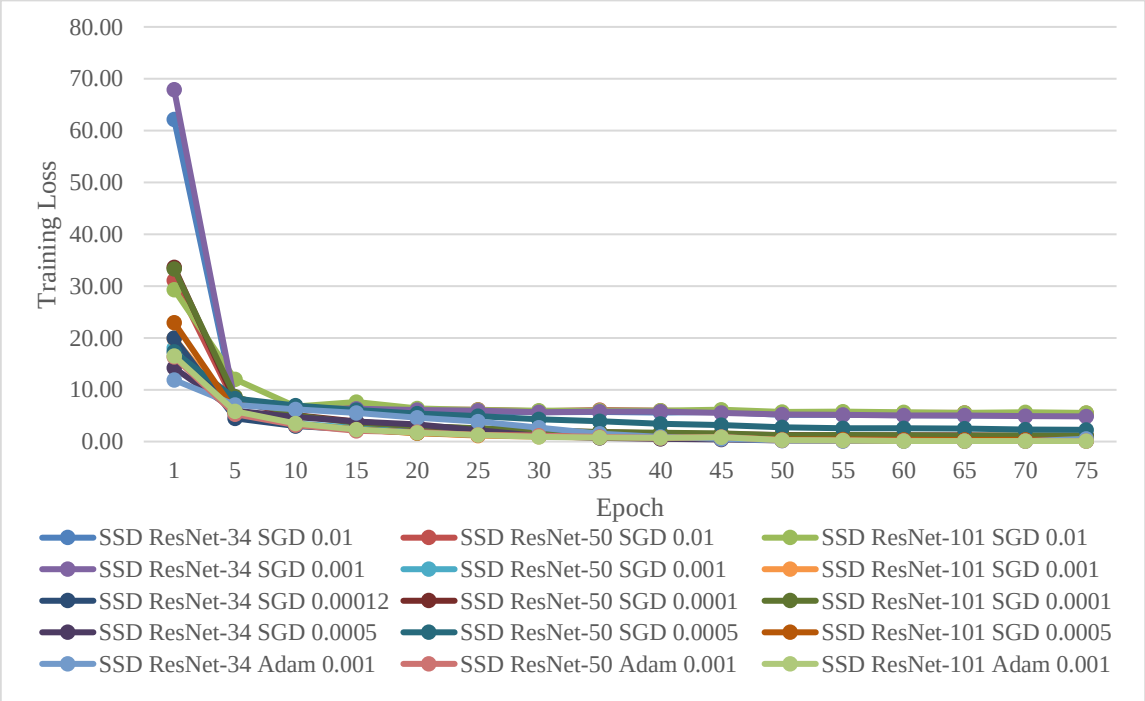

Figure 6. Train loss graph SSD training

Based on Figure 6, all models show a decreasing trend in training loss as the number of epochs increases, indicating a successful learning process. However, the stability of the loss reduction varies depending on the combination of optimizer, learning rate, and backbone architecture used. Using SGD with a high learning rate (0.01) results in significant fluctuations, while lower learning rates (0.001–0.0001) show a more stable loss reduction pattern. The SSD+ResNet-34 model shows relatively fast and stable convergence, while SSD+ResNet-50 experiences larger fluctuations, and SSD+ResNet-101 takes longer to reach stability. From the optimizer's perspective, using SGD with a learning rate of 0.0005 results in the most stable loss reduction trend compared to other combinations.

3.1.2. Testing model

Each model, with varying backbone architecture, optimizer type, and learning rate, was then tested using test data. The test data consisted of 80 datasets used for testing. Testing was conducted to assess the model's performance in recognizing digital waste images. To provide a more comprehensive qualitative comparison, the detection output of each model under its best-performing configuration is presented in Figure 7. Figure 7(a) shows the detection results of SSD–ResNet-34, Figure 7(b) shows the detection results of SSD–ResNet-50, and Figure 7(c) shows the detection results of SSD–ResNet-101. The SSD–ResNet-34 model shows consistent and accurate detection, even under varying lighting conditions. In contrast, SSD–ResNet-50 sometimes misclassifies transparent plastic as paper due to color and texture similarities, while SSD–ResNet-101 tends to produce redundant bounding boxes for similar objects due to feature overrepresentation.

Some detection results still fail to correctly identify objects or confuse one class of waste with another, such as misclassifying paper as plastic. These misclassifications are mainly caused by visual similarities between materials, including color hues and surface reflectance, which reduce the discriminatory power between classes. Similar findings were also reported by Kang *et al.* [48], who emphasized that confusion tends to occur when interclass features are not sufficiently discriminatory, and Chhabra *et al.* [49], who highlighted that low interclass variability is a major cause of misclassification in a litter image dataset.

3.1.3. Performance

This study evaluates the performance of three SSD-based object detection models—each integrated with a different ResNet architecture (ResNet-34, ResNet-50, and ResNet-101)—using various optimizer settings and learning rates. Evaluation metrics include precision, recall, F1-score, and mAP, as summarized in Table 3.

Table 3. Training results of ResNet architecture on SSD algorithm

Parameter	Model	Precision (%)	Recall (%)	F1-score (%)	mAP (%)
SGD 0.01	SSD+ResNet-34	42.58	48.25	40.81	28.85
	SSD+ResNet-50	42.69	34.75	32.59	20.19
	SSD+ResNet-101	28.07	28.25	21.61	30.71
SGD 0.001	SSD+ResNet-34	53.87	56.75	40.81	29.58
	SSD+ResNet-50	60	59	55	41.75
	SSD+ResNet-101	63.4	60.5	61.3	59.7
SGD 0.0001	SSD+ResNet-34	61.9	64.7	60.9	43.71
	SSD+ResNet-50	63.83	64.5	60.15	46.96
	SSD+ResNet-101	71.21	69.25	65.83	33.54
SGD 0.0005	**SSD+ResNet-34**	**79.52**	**82.5**	**79.37**	**91.02**
	SSD+ResNet-50	77.37	80.25	75.45	66.61
	SSD+ResNet-101	67.52	66.5	63.45	46.13
Adam 0.001	SSD+ResNet-34	64.26	67.33	64.94	58.26
	SSD+ResNet-50	52.94	54.13	49.35	42.10
	SSD+ResNet-101	58.39	54.76	53.3	47.89

3.2. Discussion

Compared with previous studies, the proposed SSD with ResNet backbone shows competitive performance. Meng *et al.* [16] reported an mAP of 93.6% using SSD-MobileNet, while Haldorai *et al.* [17] achieved an accuracy of 94% on an intelligent vision-based water waste cleaning robot. Although our results are slightly lower, these studies focused on multiclass problems with more diverse categories, which are inherently more challenging. Furthermore, comparisons across different ResNet backbones show that deeper architectures (ResNet-50 and ResNet-101) do not always produce better results on small-scale datasets, as they are prone to overfitting and increased computational burden.

The qualitative results in Figure 7 further confirm this pattern, showing that SSD–ResNet-34 produces more reliable and precise detections compared to deeper backbones. Misclassifications primarily

occur between the plastic and paper classes, as both have similar reflective properties and textures. This finding is consistent with Kang *et al.* [48] and Chhabra *et al.* [49], who noted that limited inter-class distinction in waste datasets often leads to confusion during feature extraction. Despite these challenges, the SSD–ResNet-34 configuration demonstrates stable detection performance across a wide range of lighting conditions and object scales, demonstrating strong feature generalization in a limited dataset environment.

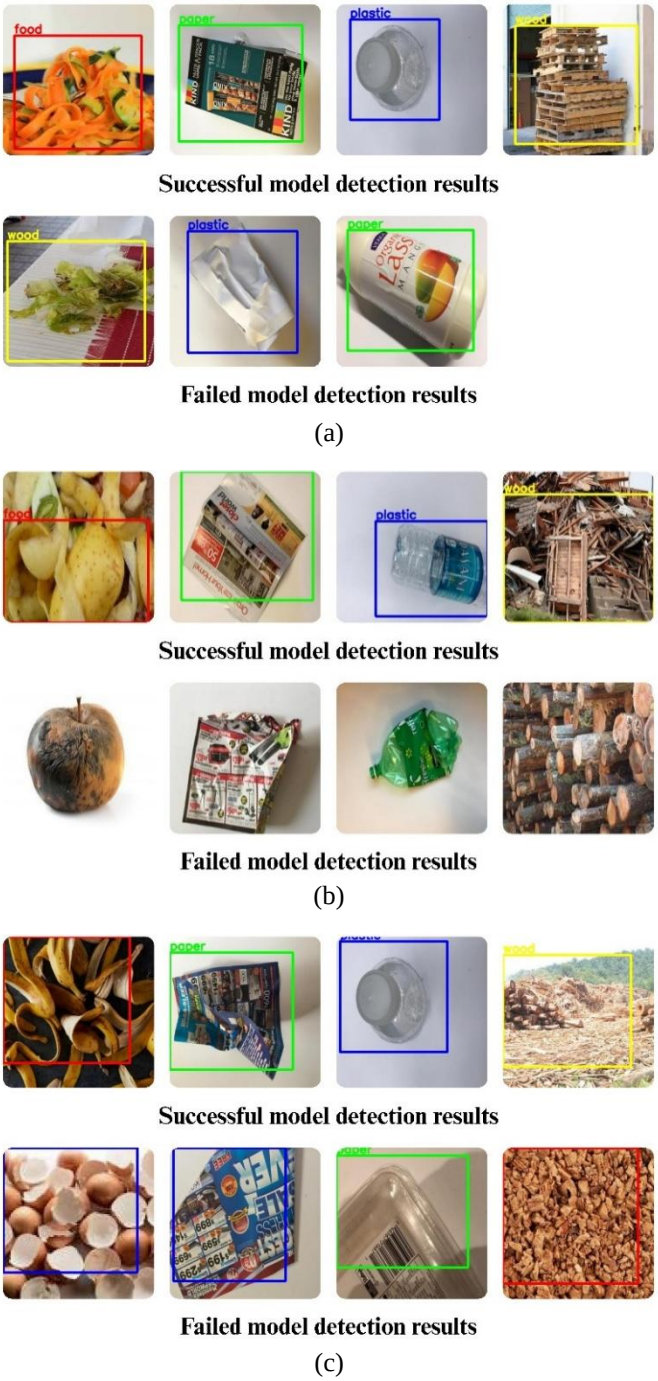

Figure 7. Visual comparison of detection outputs using; (a) SSD–ResNet-34, (b) SSD–ResNet-50, and (c) SSD–ResNet-101 models under their best-performing configurations

Furthermore, based on the test results in Table 3, it can be seen that the combination of the SSD+ResNet-34 algorithm with the SGD optimizer and a learning rate of 0.0005 produces the best performance, with a precision of 79.52%, recall of 82.5%, F1-score of 79.37%, and mAP50 of 91.02%. The

model with the ResNet-50 backbone only recorded a maximum mAP50 of 66.61%, while the ResNet-101 was even lower at 46.13%. In addition, the Adam optimizer consistently produced lower mAP compared to SGD across all backbones, indicating that SGD is more suitable in the context of the limited dataset used.

These findings confirm that the lower-complexity backbone architecture (ResNet-34) is superior to the deeper architecture, as it minimizes the risk of overfitting and is more efficient on small-scale datasets. This is in line with previous findings [48], which showed that lightweight models are able to achieve optimal performance under balanced parameter settings. In contrast, ResNet-101, although deeper, is more prone to overfitting and requires a larger dataset and a more complex training strategy to achieve good generalization [49].

Furthermore, these results indicate that optimizer selection significantly impacts model performance. Adam does tend to converge faster, but test results show weaker generalization compared to SGD. This is in line with previous studies reporting that SGD is superior by providing better generalization and more stable convergence, whereas Adam, despite its faster convergence, often leads to weaker performance in object detection tasks [50], [51]. Similarly, inappropriate hyperparameter selection, such as too high a learning rate, can cause the model to oscillate during training and fail to reach the optimal convergence point, thus degrading model performance [52].

Overall, this study makes a novel contribution by comparing different ResNet variants as the backbone of SSD in the context of waste detection. The results show that a lighter model like ResNet-34, with appropriate hyperparameter settings, can provide the best balance between accuracy and efficiency. These findings are relevant for real-world applications in developing countries, where limited computing resources are a key consideration in building sustainable deep learning-based waste detection systems.

4. CONCLUSION

This study compares three ResNet backbone architectures (ResNet-34, ResNet-50, and ResNet-101) within the SSD framework for visual-based waste detection. Experimental results show that SSD–ResNet-34 achieves the best balance between accuracy and efficiency on limited datasets, outperforming deeper variants. This finding suggests that sufficiently deep architectures can provide optimal generalization on small-scale datasets while maintaining real-time detection capabilities. Furthermore, the use of SGD provides better generalization than Adam, confirming its effectiveness for training object detection models under limited data conditions.

The novelty of this study lies in the systematic evaluation of the interaction between backbone–optimizer–learning rate, which offers empirical insights into how network depth affects detection accuracy and computational efficiency. Unlike previous SSD-based studies that focused on a single backbone or dataset, this study emphasizes robustness and implementation feasibility testing, providing both theoretical and practical contributions to deep learning-based waste detection.

From an engineering perspective, the proposed SSD–ResNet-34 model is lightweight and efficient, making it suitable for deployment on edge devices such as the Jetson Nano or Raspberry Pi. When integrated into IoT-based or cloud-based waste management systems, this model can automate the detection and classification processes to improve the efficiency of recycling, collection, and monitoring in smart cities, especially in resource-constrained environments.

However, this study is limited by the size and diversity of the dataset, which may limit its generalizability to more complex real-world conditions. Future research should extend this study by incorporating a hybrid CNN-Transformer architecture, expanding the dataset to encompass more environmental variability, and conducting real-world field tests. These steps will further enhance the robustness, scalability, and interpretability of next-generation smart waste management systems.

FUNDING INFORMATION

This research was supported by Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Sebelas Maret under the 2026 Penguatan Kapasitas Grup Riset (PKGR-UNS) scheme.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Zahra Khalila Salsabila	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Nurcahya Pradana	✓	✓		✓						✓		✓	✓	
Taufik Prakisya														
Febri Liantoni		✓		✓						✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

All trained models, confusion matrix, detection results for 15 types of SSD models with ResNet, and all test images can be accessed through this link: https://github.com/zahrakhalila/SSD-ResNet-Waste-Detection.

REFERENCES

[1] A. Maalouf and A. Mavropoulos, "Re-assessing global municipal solid waste generation," *Waste Management and Research*, vol. 41, no. 4, pp. 936–947, 2023, doi: 10.1177/0734242X221074116.

[2] W. Czekała, D. Janczak, P. Pochwatka, M. Nowak, and J. Dach, "Gases Emissions during Composting Process of Agri-Food Industry Waste," *Applied Sciences (Switzerland)*, vol. 12, no. 18, p. 9245, 2022, doi: 10.3390/app12189245.

[3] S. Kaza, L. Yao, P. Bhada-Tata, and F. Van Woerden, *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050*. Washington DC: World Bank Publications, 2018, doi: 10.1596/978-1-4648-1329-0.

[4] Ministry of Environment and Forestry of the Republic of Indonesia, "2024 Waste Source Data," *SIPSN (National Waste Management Information System)*, 2024.

[5] T. W. Wu, H. Zhang, W. Peng, F. Lü, and P. J. He, "Applications of convolutional neural networks for intelligent waste identification and recycling: A review," *Resources, Conservation and Recycling*, vol. 190, p. 106813, 2023, doi: 10.1016/j.resconrec.2022.106813.

[6] B. Fang *et al.*, "Artificial Intelligence for Waste Management in Smart Cities: A Review," *Environmental Chemistry Letters*, vol. 21, no. 4, pp. 1959–1989, 2023, doi: 10.1007/s10311-023-01604-3.

[7] Robin, P. Kaur, J. Kaur, and S. A. Bhat, "Solid waste management: challenges and health hazards," *Recent Trends in Solid Waste Management*, no. 3, pp. 171–195, 2023, doi: 10.1016/B978-0-443-15206-1.00002-5.

[8] M. O. Adelodun and E. C. Anyanwu, "Public Health Risks Associated with Environmental Radiation from Improper Medical Waste Disposal," *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 6, no. 2, pp. 21–32, 2025, doi: 10.54660/.IJMRGE.2025.6.2.21-32.

[9] P. Brancoli, K. Bolton, and M. Eriksson, "Environmental impacts of waste management and valorisation pathways for surplus bread in Sweden," *Waste Management*, vol. 117, pp. 136–145, 2020, doi: 10.1016/j.wasman.2020.07.043.

[10] S. M. Raza, S. M. G. Hassan, S. A. Hassan, and S. Y. Shin, "Real-Time Trash Detection for Modern Societies using CCTV to Identifying Trash by utilizing Deep Convolutional Neural Network," *arXiv preprint*, 2021, doi: 10.48550/arXiv.2109.09611.

[11] M. Abdallah, M. A. Talib, S. Feroz, Q. Nasir, H. Abdalla, and B. Mahfood, "Artificial intelligence applications in solid waste management: A systematic research review," *Waste Management*, vol. 109, pp. 231–246, 2020, doi: 10.1016/j.wasman.2020.04.057.

[12] Y. Ren, C. Zhu, and S. Xiao, "Object Detection Based on Fast/Faster RCNN Employing Fully Convolutional Architectures," *Mathematical Problems in Engineering*, vol. 2018, no. 1, p. 3598316, 2018, doi: 10.1155/2018/3598316.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[14] W. Liu *et al.*, "SSD: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[15] X. Gao, J. Xu, C. Luo, J. Zhou, P. Huang, and J. Deng, "Detection of Lower Body for AGV Based on SSD Algorithm with ResNet," *Sensors*, vol. 22, no. 5, p. 2008, 2022, doi: 10.3390/s22052008.

[16] J. Meng, P. Jiang, J. Wang, and K. Wang, "A MobileNet-SSD Model with FPN for Waste Detection," *Journal of Electrical Engineering and Technology*, vol. 17, pp. 1425–1431, 2022, doi: 10.1007/s42835-021-00960-w.

[17] A. Haldorai, B. Lincy R, S. M, and M. Balakrishnan, "An improved single short detection method for smart vision-based water garbage cleaning robot," *Cognitive Robotics*, vol. 4, pp. 19–29, 2024, doi: 10.1016/j.cogr.2023.11.002.

[18] J. Fang, "SSD-based Lightweight Recyclable Garbage Target Detection Algorithm," *Innovation in Science and Technology*, vol. 1, no. 1, pp. 40–45, 2022, doi: 10.56397/ist.2022.08.05.

[19] M. Karthikeyan, T. S. Subashini, and R. Jebakumar, "SSD based waste separation in smart garbage using augmented clustering NMS," *Automated Software Engineering*, vol. 28, no. 2, pp. 1–17, 2021, doi: 10.1007/s10515-021-00296-9.

[20] S. H. Lee, T. W. Hou, C. H. Yeh, and C. S. Yang, "A Lightweight Neural Network Based on AlexNet-SSD Model for Garbage Detection," in *Proceedings of the 2019 3rd High Performance Computing and Cluster Technologies Conference*, 2019, pp. 274–278, doi: 10.1145/3341069.3341087.

[21] 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), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90..
[22] I. C. Duta, L. Liu, F. Zhu, and L. Shao, "Improved residual networks for image and video recognition," in *Proceedings - International Conference on Pattern Recognition*, 2020, pp. 9415–9422, doi: 10.1109/ICPR48806.2021.9412193.
[23] Z. Wang, L. Ye, F. Chen, T. Zhou, and Y. Zhao, "Multi-category sorting of plastic waste using Swin Transformer: A vision-based approach," *Journal of Environmental Management*, vol. 370, p. 122742, 2024, doi: 10.1016/j.jenvman.2024.122742.
[24] K. Huang, H. Lei, Z. Jiao, and Z. Zhong, "Recycling waste classification using vision transformer on portable device," *Sustainability (Switzerland)*, vol. 13, no. 21, pp. 1–14, 2021, doi: 10.3390/su132111572.
[25] T. Ji, H. Fang, R. Zhang, J. Yang, Z. Wang, and X. Wang, "Plastic waste identification based on multimodal feature selection and cross-modal Swin Transformer," *Waste Management*, vol. 192, pp. 58–68, 2025, doi: 10.1016/j.wasman.2024.11.027.
[26] Ministry of Environment and Forestry of the Republic of Indonesia, "Waste Composition," *SIPSN (National Waste Management Information System)*, pp. 6–7, 2024.
[27] M. Mohamed, "Garbage Classification (12 Classes)," Kaggle, 2020. https://www.kaggle.com/datasets/mostafaabla/garbage-classification. (Accessed Jun. 24, 2024).
[28] A. Dutt and A. Dutt, "Waste Segregation Image Dataset," Kaggle, 2022. https://www.kaggle.com/datasets/aashidutt3/waste-segregation-image-dataset. (Accessed Jun. 24, 2024).
[29] T. Alda, "Dataset Waste," *Kaggle*, 2024. https://www.kaggle.com/datasets/talithaalda/datasetwaste1 (accessed Jun. 24, 2024).
[30] F. D. Cahya, F. Savitri, M. Anantha, N. Hanan, and R. Annafii, "Garbage Dataset," *GitHub*, 2019.
[31] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," *Digital Signal Processing: A Review Journal*, vol. 132, 2022, doi: 10.1016/j.dsp.2022.103812.
[32] E. Nichani, A. Radhakrishnan, and C. Uhler, "Increasing Depth Leads to U-Shaped Test Risk in Over-parameterized Convolutional Networks," *arXiv*, pp. 1–27, 2020, doi: 10.48550/arXiv.2010.09610.
[33] F. Chen and J. Y. Tsou, "Assessing the effects of convolutional neural network architectural factors on model performance for remote sensing image classification: An in-depth investigation," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, p. 102865, 2022, doi: 10.1016/j.jag.2022.102865.
[34] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing: A Review Journal*, vol. 126, 2022, doi: 10.1016/j.dsp.2022.103514.
[35] L. Cheng, Y. Ji, C. Li, X. Liu, and G. Fang, "Improved SSD network for fast concealed object detection and recognition in passive terahertz security images," *Scientific Reports*, vol. 12, no. 1, pp. 1–16, 2022, doi: 10.1038/s41598-022-16208-0.
[36] L. H. Shehab, O. M. Fahmy, S. M. Gasser, and M. S. El-Mahallawy, "An efficient brain tumor image segmentation based on deep residual networks (ResNets)," *Journal of King Saud University - Engineering Sciences*, vol. 33, no. 6, pp. 404–412, 2021, doi: 10.1016/j.jksues.2020.06.001.
[37] N. P. T. Prakisya, A. Supriyadi, and A. Wirawan, "Optimizing GPU Performance in Machine Learning with Resizable BAR : An Analytical Study," *International Journal on Advanced Science, Engineering & Information Technology*, vol. 15, no. 4, pp. 1021–1028, 2025, doi: 10.18517/ijaseit.15.4.20540.
[38] E. Aydemir *et al.*, "Hybrid Deep Feature Generation for Appropriate Face Mask Use Detection," *International Journal of Environmental Research and Public Health*, vol. 19, no. 4, 2022, doi: 10.3390/ijerph19041939.
[39] M. Gao, J. Chen, H. Mu, and D. Qi, "A transfer residual neural network based on resnet-34 for detection of wood knot defects," *Forests*, vol. 12, no. 2, pp. 1–16, 2021, doi: 10.3390/f12020212.
[40] S. Rajpal, N. Lakhyani, A. K. Singh, R. Kohli, and N. Kumar, "Using Handpicked Features in Conjunction with ResNet-50 for Improved Detection of COVID-19 from Chest X-ray Images," *Chaos, Solitons and Fractals*, vol. 145, no. 110749, 2021, doi: 10.1016/j.chaos.2021.110749.
[41] P. N. Ma'rifah, M. Sarosa, and E. Rohadi, "Comparison of Faster R-CNN ResNet-50 and ResNet-101 Methods for Recycling Waste Detection," *International Journal of Computer Applications Technology and Research*, vol. 12, no. 12, pp. 26–32, 2023, doi: 10.7753/IJCATR1212.1006.
[42] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. Da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics (Switzerland)*, vol. 10, no. 3, pp. 1–28, 2021, doi: 10.3390/electronics10030279.
[43] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Trends in vision-based machine learning techniques for plant disease identification: A systematic review," *Expert Systems with Applications*, vol. 208, 2022, doi: 10.1016/j.eswa.2022.118117.
[44] R. Padilla, S. L. Netto, and E. A. B. Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Niteroi, Brazil, pp. 237–242, 2020, doi: 10.1109/IWSSIP48289.2020.9145130.
[45] M. M. Talha, H. U. Khan, S. Iqbal, M. Alghobiri, T. Iqbal, and M. Fayyaz, "Deep learning in news recommender systems: A comprehensive survey, challenges and future trends," *Neurocomputing*, vol. 562, 2023, doi: 10.1016/j.neucom.2023.126881.
[46] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Computers and Operations Research*, vol. 152, 2023, doi: 10.1016/j.cor.2022.106131.
[47] C. Miller, T. Portlock, D. M. Nyaga, and J. M. O'Sullivan, "A review of model evaluation metrics for machine learning in genetics and genomics," *Frontiers in Bioinformatics*, vol. 4, pp. 1–13, 2024, doi: 10.3389/fbinf.2024.1457619.
[48] Z. Kang, J. Yang, G. Li, and Z. Zhang, "An Automatic Garbage Classification System Based on Deep Learning," *IEEE Access*, vol. 8, pp. 140019–140029, 2020, doi: 10.1109/ACCESS.2020.3010496.
[49] M. Chhabra, B. Sharan, M. Elbarachi, and M. Kumar, "Intelligent Waste Classification Approach Based on Improved Multi-Layered Convolutional Neural Network," *Multimedia Tools and Applications*, 2024, doi: 10.1007/s11042-024-18939-w.
[50] H. Naganuma *et al.*, "Empirical Study on Optimizer Selection for Out-of-Distribution Generalization," *Transactions on Machine Learning Research*, vol. 2023, 2023, doi: 10.48550/arXiv.2211.08583.
[51] J. Yang, M. Bagavathiannan, Y. Wang, Y. Chen, and J. Yu, "A comparative evaluation of convolutional neural networks, training image sizes, and deep learning optimizers for weed detection in alfalfa," *Weed Technology*, vol. 36, no. 4, pp. 512–522, 2022, doi: 10.1017/wet.2022.46.
[52] C. Peng, "Comprehensive Analysis of the Impact of Learning Rate and Dropout Rate on the Performance of Convolutional Neural Networks on the CIFAR-10 Dataset," *Applied and Computational Engineering*, vol. 102, no. 1, pp. 183–192, 2024, doi: 10.54254/2755-2721/102/20241161.

.

BIOGRAPHIES OF AUTHORS

Zahra Khalila Salsabila is an undergraduate student at Universitas Sebelas Maret Surakarta, Indonesia, majoring in Computer and Informatics Engineering Education. Her research focuses on artificial intelligence, particularly in the areas of deep learning, computer vision, and object detection. She has experience in data mining techniques, including Naïve Bayes classification. Her current research explores the implementation of deep learning models for object detection tasks. She can be contacted at email: zkhalilas1524@student.uns.ac.id.

Nurcahya Pradana Taufik Prakisya was born in Surakarta, Indonesia in 1991. He received his bachelor's degree in informatics from Universitas Sebelas Maret, in 2013 and master's degree in computer science from Universitas Gadjah Mada, in 2017. He is currently a lecturer in software engineering and artificial intelligence at Bachelor of Informatics Education, Faculty of Teacher Training and Education, Universitas Sebelas Maret. His research interests focus on computer vision, particularly in medical imaging. He can be contacted at email: nurcahya.ptp@staff.uns.ac.id.

Febri Liantoni received his bachelor's degree in informatics engineering from Politeknik Elektronika Negeri Surabaya, in 2010 and a master's degree in information technology from Institut Teknologi Sepuluh Nopember, in 2015. He is currently a lecturer in data mining, and artificial intelligence at the Bachelor of Informatics Education, Faculty of Teacher Training and Education, Universitas Sebelas Maret. His research interests focus on data mining. He can be contacted at email: febri.liantoni@staff.uns.ac.id.