ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 14 (2024) No. 5, pp. 420-439
DOI:10.33168/JSMS.2024.0526

Efficient E-waste Classification for Sustainable Recycling

Kah Ming Ong [1], Shih Yin Ooi [1, *], Chin Leei Cham [2], Wei Peng Kwan [3], Ying Han Pang [1], Sze Yen Low [1], Jia Er Tneo [1]

[1] Faculty of Information Science & Technology (FIST), Multimedia University

[2] Faculty of Engineering (FOE), Multimedia University

[3] Universiti Putra Malaysia

syooi@mmu.edu.my (Corresponding author)

Abstract. The growing number of manufactured electrical and electronic devices has led to increased electronic waste (e-waste), including damaged, useless, or outdated electronic equipment. Effective e-waste management has become a global issue, and one way to address this problem is through machine learning techniques to classify e-waste and identify its most valuable components. In this paper, we deployed machine learning algorithms like random forest (RF), support vector machine (SVM), decision tree (DT), and k-nearest neighbors (KNN) comparing with deep learning algorithm "You Only Look Once" version 5 (YOLOv5) with the e-waste bin for image classification to identify e-waste and non-e-waste. The YOLOv5 with smart container uses features to extract relevant information from the images to classify each student input sample into e-waste or non-e-waste. Our results show that the YOLOv5 classifier achieved an accuracy of 90.32% in classifying the e-waste samples.

Keywords: E-waste, Object detection, Machine learning, Classification, YOLOv5, RF, SVM, DT, KNN

1. Introduction

Due to the increased manufacturing of electrical and electronic gadgets, the word "e-waste" refers to broken, worthless, or outmoded electronic equipment, and it has become a global concern in today's technology-driven society. The incorrect disposal of electronic trash can have a negative impact on the environment and overall well-being and is frequently the result of inadequate recycling and disposal processes. As the need for ethical e-waste management grows, we must grasp the gaps in the literature and define a clear path for our efforts. Recent research in this sector has highlighted the growing e-waste problem, underlining the vital need for novel solutions.

To address this growing concern, we propose installing an intelligent e-waste container on the Multimedia University Melaka campus to raise students' awareness of e-waste recycling. This smart bin recognizes and categorizes e-waste and non-e-waste objects using deep learning algorithms, with an emphasis on small IT and telecommunication gadgets often used by university students.

By utilizing cutting-edge technology, we hope to fill a gap in practical e-waste management solutions and contribute to sustainable waste disposal practices. In doing so, we seek to not only decrease the environmental impact of e-waste, but also to motivate students and the wider community to take a more informed and responsible attitude to e-waste disposal. This study builds on previous research on the subject, providing a fresh and practical addition to the ongoing endeavor to properly handle e-waste. The development and deployment of an intelligent e-waste classification system is the primary topic of this article. This mechanism is critical to our solution for various reasons.

An effective e-waste classification system is critical for accurately distinguishing between electronic waste and non-waste items. In an age when electronic devices are constantly evolving and becoming more integrated into our daily lives, the ability to precisely identify e-waste is essential. This classification system enables us to specifically target e-waste, ensuring that discarded electronics are correctly separated from regular waste.

In addition, by implementing a sophisticated classification system, we will be able to better quantify and track the volume of e-waste generated within the academic environment. This information is essential for determining the scope of the e-waste problem and developing targeted strategies for responsible disposal and recycling.

Furthermore, our classification system lays the groundwork for effective e-waste recycling and management. It enables the automation of sorting processes, reducing the burden on human labor and minimizing categorization errors. This efficiency not only speeds up the recycling process, but it also encourages more participation in e-waste disposal among the university community.

Finally, the successful implementation of this classification system is the cornerstone of our solution, allowing us to develop a more environmentally sustainable and user-friendly approach to e-waste management on campus. By precisely distinguishing e-waste from non-waste items, we pave the way for a more informed and efficient recycling process, emphasizing the significance of this system in our research.

The structure of the paper is as follows: In Section 2, the existing literature on the subject is discussed. Section 3 covers the model selection and the proposed system, including data collection, image pre-processing, and model training. Section 4 focuses on the key indicators, the results of the training process, and the visualization of the findings. Finally, Section 5 presents the conclusions drawn from the study.

2. Related Works

2.1. Deep Learning and Machine Learning Methods

In year 2016, researchers (Yang & Thung, 2016) proposed a trash classification system based on the recyclability status to solve the problem of sorting trash manually. The system divided the 2400 picture dataset into the six categories of paper, glass, plastic, metal, cardboard, and trash using Support Vector Machine (SVM) machine learning algorithm compared to Convolutional Neural Network (CNN). Scale-Invariant Feature Transform (SIFT) was employed for the SVM, where each pixel in an image was compared to its neighboring pixels at different scales. Due to the relatively small class size in the dataset, data augmentation techniques that randomize picture rotation, brightness control, translation, scaling, and shearing were applied to each image.

In the same year, (Sakr et al., 2016) developed a trash bin that automatically sorts rubbish on small and big scales using AlexNet and SVM classifiers. The overall result is a Raspberry Pi-controlled trash bin that uses artificial intelligence to automatically identify waste type and transfer it to the appropriate container after taking a photo. A Pi camera is connected to Raspberry Pi, which implements a simple Python script that captures images; then, the images are sorted as 256 x 256 lossless PNG-colored images. The training set was artificially increased to 6000 images using label-preserving transformations to minimize overfitting.

Except for the SVM classifier, an AI-automated sorting trash bin (Fieno Silva et al., 2018) was developed in the year 2018 by adopting the CNNs and ML algorithms and comparing against VGG-16, AlexNet, SVM, K-nearest Neighbors (KNN) and Random Forest (RF). The image database uses the same as Yang et al. (2016), which consists of around 2400 images across six classes with about 400-500 images each. The author also performed a balanced one-way Analysis of Variance (ANOVA) to compare the classifier's methods; the results show that the difference between the methods is significant at the 0.05 level.

In a recent study by (Nasik Sami et al., 2020; Poo et al., 2023)), the efficiency of ML and DL classifiers was compared. The classifiers used were SVM, Decision Tree (DT), RF, and CNN. To collect the data, the researchers utilized an image database similar to the one used by (Yang et al., 2016). Google Colab was the chosen platform for coding purposes and to optimize GPU runtime. The original waste images were resized to 512x384 pixels and organized in specific directories. The images were converted to grayscale with dimensions of 50x50 pixels to expedite computation. Besides accuracy, the researchers also evaluated the algorithms' performance using precision, recall, and F1-score.

A trash classification system based on images was proposed in a study by (Satvilkar, 2018) using the same dataset created by (Yang et al., 2016). The study employed ML and DL algorithms, including SVM, RF, eXtreme Gradient Boosting (XGBoost), KNN, and CNN. The images were divided into five categories: cardboard, glass, metal, paper, and plastic. To enhance the dataset, augmented techniques were applied. The researcher reduced the photos' size to 10% of their original size, reducing the computation time required for the analysis. The data was partitioned for model training, with 75% of the images allocated to the training set and the remaining 25% used to evaluate the performance of the developed models.

Furthermore, the study conducted by (Adedeji and Wang, 2019) focused on analyzing a dataset of trash images initially developed by Yang et al. (2016) by utilizing ResNet-50 CNN and SVM. During the pre-training process, the ResNet-50 CNN model exhibited the ability to classify images into 1000 distinct classes. The images were resized to 512x384 pixels and categorized into four main groups: glass, paper, plastic, and metal. Data augmentation techniques were applied during the pre-processing phase to enhance the dataset. These techniques included random scaling, image shearing, and translation, which helped optimize the dataset size.

In a study conducted by (Kushal Samir Mehta et al., 2017), a module was developed for SVM and CNN classifiers into OpenCV for recognizing uncollected rubbish in metropolitan areas. To collect the data, researchers provided a smartphone application that allowed individuals to report uncollected waste. 21,000 images were collected from these complaints, divided into two categories: images indicating the presence of garbage and images indicating the absence of garbage. Before analysis, all images were converted to grayscale, and critical points were extracted from these images. Researchers opted for the

Speeded-Up Robust Features (SURF) detector to extract key points, scale, and rotation invariant and store the key points as vectors and descriptors as Mat objects. OpenCV played a significant role by simplifying programming through trainAuto, which dynamically adjusts the model's parameters during training.

In 2021, a team of researchers led by (Dassi et al., 2021) proposed an innovative system for identifying and categorizing e-waste using the "You Only Look Once" Version 5 (YOLOv5) DL algorithm. They also developed a user-friendly web portal that simplifies the connection between e-waste producers and collectors, streamlining the collection process. The classification task involves the waste producer submitting an image of e-waste to the YOLOv5 model. The model identifies the types of e-waste present in the image and classifies them into specific categories. To train their system, the researchers created an extensive dataset of e-waste, consisting of over 2.4k images. Each image was manually annotated using a Python-based graphical tool called LabelImg. The training process was remarkably efficient, with the model completing 500 epochs in just 2.5 hours. The evaluation of the object detection models, including R-CNN and YOLO, utilized the mean average precision (mAP) metric. The mAP measures the model’s accuracy by comparing the detected bounding boxes with the ground-truth bounding boxes. Furthermore, precision scores were calculated to evaluate the model's performance for each e-waste class.

In the same year, a group of researchers led by (G. Yang et a., 2021) conducted a study using the YOLOv5 algorithm to enhance garbage identification and detection. Researchers found that YOLO outperformed models like Deformable Parts Model (DPM) and R-CNN. To conduct their experiments, the researchers utilized the TACO dataset, which provided a diverse collection of garbage images captured in different environmental settings. To train and evaluate the target detection algorithm, the images were labeled and classified manually in a hierarchical manner. The experiment focused on 8 waste classes and utilized a dataset of 1195 images. YOLOv5 was employed to identify the distinctive features of waste, ensuring the effectiveness of garbage classification. 90% of the images were used for training, while the remaining 10% were used for testing. They designed a batch size of 24 and trained the model for 50 epochs. Overall, the researchers successfully demonstrated the effectiveness of YOLOv5 in accurately detecting and identifying different types of garbage, significantly improving the results compared to their initial training attempt.

A summary of various ML and DL approaches for e-waste classification is presented in Table 1.

Table 1. Related Works of Deep Learning and Machine Learning Methods

Author	No of class (No of samples)	Feature Extraction	Approaches	Accuracy (%)
(M. Yang & Thung, 2016)	6(2400)	SIFT	SVM CNN	63% 22%
(Sakr et al., 2016)	3(6000)	-	SVM AlexNet(CNN)	94.8% 83%
(Fieno Silva et al., 2018)	6(2400)	-	VGG-16 AlexNet RF KNN SVM	93% 91% 88% 85% 80%
(Nasik Sami et al., 2020)	6(2400)	-	CNN SVM DT RF	90% 85% 65% 55%
(Adedeji & Wang, 2019)	4(2000)	-	ResNet-50 SVM	94.5% 87%

(Satvilkar, 2018)	5(2400)	-	CNN XGBoost SVM RF KNN	89.8% 70.1% 65.7% 62.6% 52.5%
(Kushal Samir Mehta et al., 2017)	2(21000)	SURF	CNN+OpenCV SVM+OpenCV	85% 72%
(Dassi et al., 2021)	7(2400)	-	YOLOv5	88.0%
(G. Yang et al., 2021)	8(1195)	-	YOLOv5	94.5%

In comparing ML algorithms, various models were considered, including RF, SVM, KNN, and DT. These algorithms have been studied extensively, and their performance levels have been documented. SVM consistently achieved the highest accuracy, ranging from 63% to 94.8% in different experiments. RF, KNN, and DT also performed well, with 65% to 88% accuracy. These ML algorithms will be utilized in this research based on these findings. Regarding deep learning, the YOLOv5 algorithm was a suitable choice due to its exceptional performance in live detection tasks.

2.2. Reviews of Recent E-Waste Classification Systems

In recent years, several works have utilized YOLO extension in classifying e-waste. Effective waste classification is a critical aspect of urban management influencing societal sustainability. Previous research has concentrated on single-category waste recognition, which falls short in real-world waste classification scenarios. A YOLO-WASTE multi-label waste classification model based on transfer learning was developed by Zhang et al. (2022) to achieve rapid recognition and classification of multiple waste types. A multi-label waste image dataset with images from various waste categories is created to improve learning efficiency. The experimental results show that the YOLO-WASTE model has a remarkable mAP value of 92.23%, with an average image detection time of 0.424 seconds, outperforming other classification algorithms. The proposed YOLO-WASTE model provides new insights into complex waste identification, potentially advancing waste management efficiency and supporting sustainable urban development.

Integrating artificial intelligence into waste management in the context of "Zero Waste" and Industry 4.0 initiatives has resulted in a proliferation of image data, accompanied by advances in analytical techniques. Convolutional neural networks (CNNs), in particular, have emerged as critical tools for uncovering hidden patterns within visual features. While CNNs have gained popularity in intelligent waste identification and recycling (IWIR), their application in environmental research is relatively new, resulting in a lack of standardized benchmarks and datasets. The review conducted by Wu et al. (2023) begins with an introduction to basic CNN concepts, followed by an overview of open-source datasets and advanced CNN models used in IWIR, focusing on three primary tasks: classification, object detection, and segmentation. It then combines CNN applications in three key areas of IWIR: recyclable material identification, waste pollution detection, and solid waste classification. Finally, the review discusses the current challenges and limitations of CNN applications and the future potential of CNNs in waste management and environmental research.

However, Qiao et al. (2023) postulated that the existing waste classification models face difficulties in accurately classifying waste in low-illumination settings due to reduced object discriminability. To address this issue, the Dark-Waste waste classification model is introduced. The model employs an efficient Illumination Conversion method to generate low-light images, which mitigates the scarcity of training data. In addition, it combines an improved ConvNeXt network with YOLOv5 for precise and efficient waste classification. Dark-Waste demonstrates superior detection performance in low-illumination environments after validation on a custom real-world dataset. Dark-Waste is a novel approach to waste management in complex environments that contributes significantly to the long-term development of urban ecological environments.

Majchrowska et al. (2023) summarize previous research on e-waste, especially with the lack of standard or benchmark datasets. They also introduced two new benchmark datasets, "detect-waste" and "classify-waste," which combine data from open-source datasets with unified annotations spanning various waste categories. A two-stage detection approach is also described, employing EfficientDet-D2 for litter localization and EfficientNet-B2 for waste classification even without fully labeled data. On the test dataset, this method achieves an average precision of up to 70% in waste detection and an approximate classification accuracy of 75%. Importantly, the study makes the code and annotations used in these studies available to the public.

The reviews show that the current research is hampered by a lack of standardized metrics and benchmarks, making effective e-waste detection difficult. Several studies address these issues by critically evaluating existing datasets and proposing novel Deep Learning-based solutions. Thus, in this research, we have collected a set of in-house datasets, and YOLO is chosen as the main deep-learning algorithm.

3. Proposed Solution

3.1. E-waste Classification Model Selection

The YOLOv5 model was chosen as a classification detection model in this research. Its numerous advantages and superior performance set it apart from other options. YOLOv5 combines the speed of the YOLO series with improved accuracy, making it particularly well-suited for real-time object detection applications. Compared to traditional machine learning methods like SVM, DT, KNN, and RF, YOLOv5 leverages deep learning techniques to achieve state-of-the-art results.
To validate the selection of YOLOv5, a comprehensive comparison of different machine learning (ML) and deep learning (DL) algorithms was conducted, as presented in Table 2.

Table 2. Comparison of Different ML and DL Algorithms

Classifier	No of class (No of samples)	Approaches	Accuracy (%)
YOLOv5	2(4600)	DL	90.32%
RF	2(4600)	ML	68%
SVM	2(4600)	ML	62%
KNN	2(4600)	ML	56%

The results demonstrated that YOLOv5 outperformed all the other classifiers. Its efficient architecture, anchor-free object detection approach, and advanced data augmentation strategies contributed to its exceptional performance. YOLOv5 strikes an excellent balance between accuracy and speed, making it the preferred choice for the classification detection tasks in this e-waste classification system.

3.2. Proposed E-waste Classification System

In this proposed system, the YOLOv5 (Ng et al., 2023) model is used for training and predicting the correct categories of the user's input. Before training the YOLOv5 model, several stages of data pre-processing are necessary. These stages involve resizing the images, labeling the images, and augmenting the image data to enhance the training process. Once the pre-processing phase is completed, the data is used to train the YOLOv5 model. After training, the pre-trained YOLOv5 model is integrated into the system to effectively classify data into e-waste and non-e-waste categories.

Moreover, the YOLOv5 model is integrated into Flask to develop a user interface that enables users to provide e-waste data as input to utilize the e-waste classification system. There are two ways to input data: using the camera's live detection function or directly uploading images into the system. The user's input, whether e-waste or non-e-waste, is then processed by the pre-trained YOLOv5 model, which accurately categorizes and analyses the input. Once the user's input is received, the system initiates the

YOLOv5 classification process. This involves running the pre-trained YOLOv5 model on the input data and performing computations and analysis to determine whether the input corresponds to e-waste or non-e-waste.

If the classification result indicates that the input is e-waste, the system saves the e-waste images in the database. This step is crucial as it improves system accuracy and expands the training data for future use. By storing the e-waste images, the system builds a growing dataset that can be utilized to enhance the training data in subsequent iterations. This expanded dataset provides a broader range of examples and variations for the YOLOv5 model to learn from. As a result, the model becomes better equipped to handle more complex scenarios and continuously improves its overall performance.

The block diagram of the proposed e-waste classification system is displayed in Fig. 1.

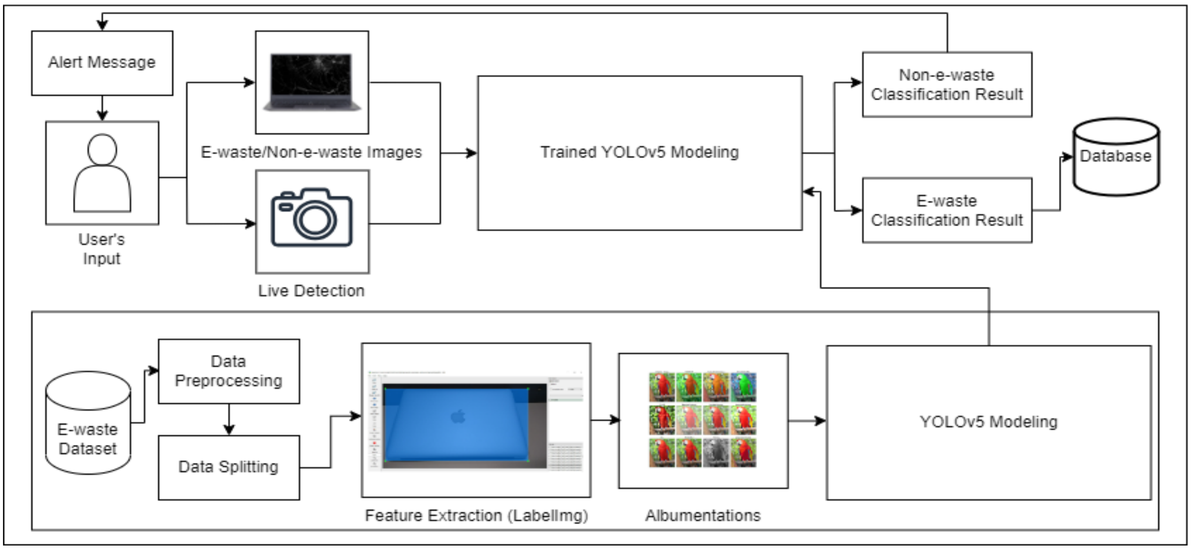

Fig. 1. Block diagram of the proposed e-waste classification system

3.3. Data Collection

Due to the shortage and unclassified public datasets, we have collected a set of in-house images comprising of commonly encountered types of e-waste, such as keyboards, laptops, mice, and phones conducted in this research, as well as some images of non-e-waste materials like glass, metal, paper, and plastic, which are frequently disposing of by the university students. Therefore, the size of the e-waste and non-e-waste we mainly focus on is small IT and telecommunication devices and non-e-waste that university students frequently use.

The dataset used to train and validate the classification system consists of 460 images of e-waste and non-e-waste. These images were augmented using image augmentation techniques to enhance the dataset's diversity and quality. The images are consolidated from various platforms, including Kaggle, specifically the datasets "(*Starter: E-Waste Dataset 93b07fb8-a | Kaggle*, n.d.)," "(*Electronics Object Image Dataset | Computer Parts | Kaggle*, n.d.)," and "(*Microcontroller Detection | Kaggle*, n.d.)".

In addition to the collected data from external sources, we gathered live images directly from the laptop's camera. The background complexity observed in the collected data can be characterized as controlled and relatively uncomplicated. Our experimental setup simulates a scenario where a student holds e-waste or non-e-waste items while standing before the camera for classification. As a result, the main emphasis in these images is placed on the objects the student holds. This approach ensures that the model can detect and distinguish between e-waste and non-e-waste categories. By incorporating this comprehensive dataset, the YOLOv5 model gains a diverse understanding of different object types, making highly accurate and reliable predictions when applied to waste classification and management tasks. The laptop's camera, with a resolution of 1280 pixels width x 720 pixels height, will also be utilized for real-time detection purposes, further enhancing the system's functionality. The summary of data collection is represented in Table 3.

Table 3. Summary of Data Collection

Number of Classes	2 (E-waste and Non-e-waste)
Number of Data Collected from Laptop Camera	E-waste: 30 Non-e-waste: 30
Number of Data Collected from Kaggle & Images.CV	E-waste: 200 Non-e-waste: 200
Image Resolution	1280 pixels width X 720 pixels height

3.4. Image Pre-processing

In the YOLOv5 model, the initial stage of deep learning involves gathering images of e-waste and incorporating them into a dataset. This dataset comprises images of both non-e-waste and e-waste and before proceeding, all the data must undergo pre-processing.

3.4.1. Image Resizing and Renaming

In YOLOv5 deep learning, resizing the images to a consistent size is often necessary. This research adjusts all data to a resolution of 1280 pixels in height x 720 pixels in width. This crucial step guarantees that all input images share the exact dimensions of the laptop's camera resolution, a necessity for most deep-learning models. Resizing alleviates computational demands and ensures compatibility with the selected model architecture. For the image rename, this can ensure unique and organized filenames or align them with specific naming conventions. Renaming the images can facilitate better organization and ease of access during the training and validation stages.

3.4.2. Data Splitting

Once the images have been resized and renamed, the next step is to divide the e-waste and non-e-waste datasets into training and validation sets. Based on the research conducted by Muraina (n.d.), since there are only two classes in the dataset (e-waste and non-e-waste) and the dataset size is considered medium, it is common practice to split the data in a 60:40 ratio. This means 60% of the data is allocated for training the model, while the remaining 40% is reserved for validating the trained model. This division serves two purposes: evaluating the model's performance on unseen data and preventing overfitting.

The training set is used to train the deep learning model by providing it with input data and corresponding target labels. Through iterative optimization, the model learns from this data, adjusting its parameters to minimize the training loss and enhance its ability to make accurate predictions. It is important for the training set to be sufficiently large, as this allows the model to capture a diverse range of patterns and examples from the target domain, enabling it to learn meaningful representations.

On the other hand, the validation set, also known as the development set or holdout set, plays a crucial role in assessing the model's performance during training. It is an unbiased evaluation metric that monitors the model's progress and aids in making decisions related to hyperparameter tuning or early stopping. By evaluating the model on unseen data from the validation set, it becomes possible to estimate its performance on new, unseen examples that fall within the categories of e-waste and non-e-waste.

The dataset is split into two parts: one for training the model and the other for validating the trained model. This ensures the model performs well on new, unseen data and avoids overfitting. The training set guides the model by providing examples and target labels, while the validation set is used to assess the model's performance and adjust if needed. Table 4 represents the number of images after the dataset splitting.

Table 4. Number of Images after Splitting

Type of Classes	Number of Images
E-waste	276
Non-e-waste	184

3.4.3. Image Labelling

Once the data pre-processing and splitting steps are completed, the subsequent task involves image labeling on the dataset. In object detection, labeling refers to assigning categorical or numerical labels to data instances or specific attributes within the data, explicitly identifying whether they belong to the e-waste or non-e-waste categories. This step holds immense significance in YOLOv5 deep learning, where each data instance is associated with a corresponding label or target value. Labelling provides crucial ground truth information for training and evaluating YOLOv5 models.

The annotation process typically entails labeling the objects of interest in the images and specifying their respective bounding boxes and class labels. These annotated images serve as the training data. During training, the YOLOv5 model learns to extract relevant features from the images and associate them with the specified object classes, namely e-waste, and non-e-waste. Accurate annotations, facilitated by tools such as LabelImg, enable the YOLOv5 model to discern the distinctive characteristics and patterns of the objects, which are vital for precise detection.

In simpler terms, after preparing the data and dividing it into training and validation sets, the next step is to label the images. Labeling means identifying and marking the objects in the images as e-waste or non-e-waste. This labeling process is important for teaching the YOLOv5 model what objects look like and how to recognize them accurately. The labeled images are then used to train the model, allowing it to learn the unique features and characteristics of e-waste and non-e-waste objects. The labeling is done precisely using a tool called LabelImg, which ensures that the model can detect and classify the objects correctly.

3.4.4. Image Augmentation

Furthermore, as mentioned earlier, the YOLOv5 model requires a substantial dataset for higher accuracy. Hence, we have incorporated the image augmentation technique to enhance the dataset. Image augmentation involves applying various transformations or modifications to the original images, aiming to increase the diversity and quantity of training data. In this research, we have implemented several augmentation techniques, such as random cropping with 640 pixels width x 640 pixels height, random rotation with a probability of 50%, random adjustments to brightness and contrast with a probability of 20%, random gamma transformation with a probability of 20%, RGB shifts with a probability of 20%, and vertical flipping with a probability of 50%. These augmentations mimic real-world variations that object in the images may exhibit. The image augmentations process is performed up to 10 times, and the augmented bounding box coordinates and class labels are then saved in the YOLO format.

By applying these augmentations, the training data becomes more robust and representative of the potential variations the model may encounter during its inference phase. This augmentation process enhances the model's ability to generalize and handle real-world scenarios, including changes in object position, scale, or lighting conditions. Moreover, image augmentation is vital in preventing overfitting, a phenomenon where the model becomes too specialized in recognizing specific training examples but struggles with new, unseen data. Exposing the model to a broader range of variations through augmentation reduces the risk of overfitting and enables the model to learn more generalizable features and patterns. Table 5 presents the count of images in the dataset after the augmentation process.

Table 5. Number of Images after Augmentation

Type of Classes	Number of Images
E-waste	2760
Non-e-waste	1840

3.5. YOLOv5 Model in E-waste Classification System

3.5.1. YOLOv5 Model

With the advancements in deep learning, the YOLO family of models offers a scalable solution for classifying electronic waste (e-waste) into distinct categories. Due to its high accuracy and efficiency in object detection, we have selected the YOLOv5 model as the preferred choice. YOLOv5 represents a series of object detection models developed and trained using the COCO dataset. As the latest offering from Ultralytics, YOLOv5 presents a valuable contribution to object detection research by providing accessible open-source solutions for future advancements in this domain. Utilizing an open-source dataset like COCO has proven highly effective in object segmentation, object detection, and classification tasks. This model encompasses three essential architectural components: Backbone, Neck, and Head, as visually depicted in Fig. 2.

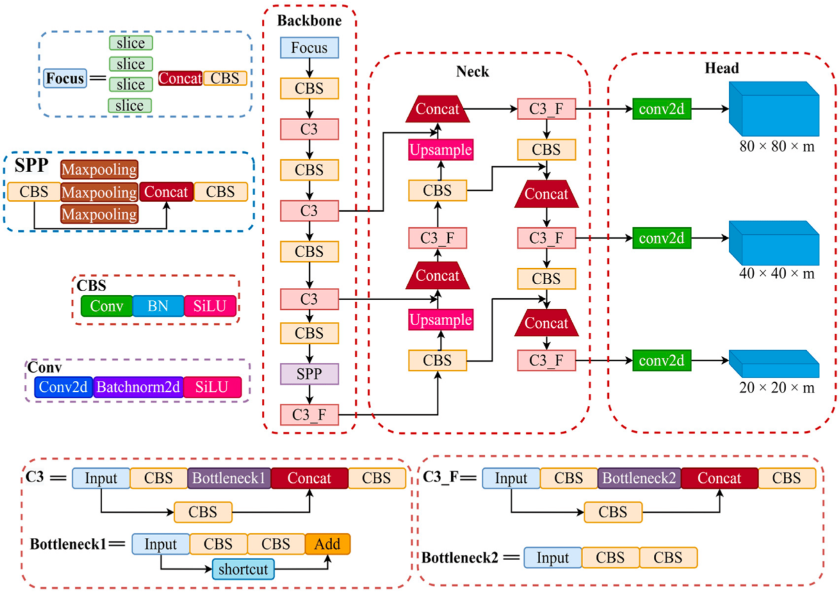

Fig. 2. YOLOv5 Architecture

For YOLOv5 Backbone, it integrates CSPDarknet as the underlying framework for extracting features from images, employing cross-stage partial networks. Including the focus, the module enables rapid downsampling of dataset images while preserving essential image information within the channel. This ensures the complete extraction of image information features without loss. The backbone layer incorporates C3, C3_F, and Spatial Pyramid Pooling (SPP) modules. The C3 and C3_F modules enhance feature extraction capability from images, simplifying the YOLOv5 model and optimizing detection speed. Furthermore, the SPP module enhances the dataset image's scale invariance, expands the receptive range of backbone features, facilitates network convergence, and improves overall accuracy.

Moreover, the YOLOv5 Neck employs PANet to establish a feature pyramid network that conducts feature aggregation and passes it to the Head module for prediction. The bottleneck layer of YOLOv5 integrates both the feature pyramid network (FPN) and path aggregation network (PAN) structures. The deep feature images contain rich semantic information but relatively weak location information, while shallow feature images possess more robust location information but weaker semantic information. Leveraging FPN allows semantic information from deep feature images to be transferred to shallow feature images. Conversely, PAN facilitates the transfer of location information from the shallow and deep feature layers. This combination of FPN and PAN enables the aggregation of parameters from various detection layers across different trunk layers, greatly enhancing the network's capacity for feature fusion.

Next, YOLOv5 Head consists of layers that generate predictions from the object detection anchor boxes. The head includes loss function and non-maximum suppression (NMS). In YOLOv5, the binary cross entropy loss function calculates classification loss and confidence loss. In contrast, the complete IoU (CIoU) loss function is applied to calculate location loss (bounding box regression loss). All the losses add up to the total loss. The CIoU loss function fully considers three key geometric parameters: the overlap area, the distance from the center point, and the aspect ratio, thus improving the speed and accuracy of the regression of the prediction box. The NMS mainly removes redundant detection boxes and reserves the candidate box with the highest prediction probability as the final prediction box.

3.5.2. YOLOv5 Model Selection

YOLOv5 offers different types of models, each with its unique configuration and parameter sizes. These models, namely YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extra-large), have their strengths and weaknesses, mainly related to their complexity, performance, and overall accuracy. According to the findings shared in the Kaggle article "YOLO V5 Object Detection Model," these models range in size from 4MB to 166MB. The mean average precision (mAP) values achieved on the full COCO dataset range from 28.4 to 50.7. Additionally, the inference time on the Nvidia V100 GPU varies from 6.3 to 12.1 milliseconds. These results indicate that the models are well-suited for real-time applications, mainly when utilized on specialized AI hardware. Fig. 3 provides an overview of the different models offered by YOLOv5.

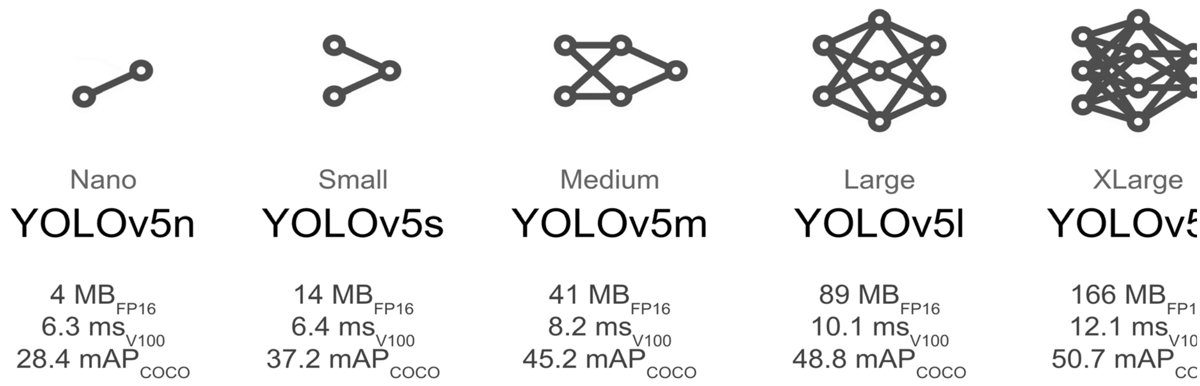

Fig. 3. Different YOLOv5 Model Selection

As we consider the different models offered by YOLOv5, namely YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, we observe a progression in size and complexity, which leads to improved accuracy. However, this improvement in accuracy comes at the cost of larger model sizes and longer inference times. In our analysis of the trade-offs, we find that YOLOv5n has the smallest model size and the fastest inference time, but it achieves the lowest mean average precision (mAP). On the other hand, YOLOv5x achieves the highest mAP but has the most significant model size and longest inference time.

For this research, we need to strike a balance between accuracy and speed. While accuracy is crucial for precise classification, the ability to process e-waste images in real-time is essential for seamlessly handling a continuous stream of data. Furthermore, selecting an appropriate model considers the specific characteristics of the e-waste dataset, such as the complexity of objects, variations in size, and instances of occlusion. We need to accurately detect and classify e-waste objects, considering that our dataset consists of only two classes: e-waste and non-e-waste. In this research, the efficiency and performance of various YOLOv5 models were examined using a single epoch and a batch size of 16 for the e-waste dataset. This approach provided insights into the training progress, allowed for evaluation of the model's performance, and offered a preliminary understanding of the results generated, as shown in Table 6.

Table 6. Results of Different YOLOv5 Models for E-waste Dataset

Model	Run Time	Precision	Recall	mAP@.5	Model
YOLOv5n	0.421 hours	0.286	0.458	0.3072	YOLOv5n
YOLOv5s	0.713 hours	0.453	0.485	0.4404	YOLOv5s
YOLOv5m	1.946 hours	0.513	0.585	0.5451	YOLOv5m
YOLOv5l	3.816 hours	0.578	0.672	0.6154	YOLOv5l

After conducting a thorough trade-off analysis, considering our specific task requirements and dataset characteristics, we have determined that YOLOv5s is the most appropriate choice for our proposed e-waste classification framework. This model strikes a favorable balance between accuracy and speed, addressing our primary concerns. While YOLOv5m and YOLOv5l exhibit higher precision and recall rates in Table 6, they come with larger model sizes, longer training times, and increased inference durations. Referring to Fig. 3, we observe that YOLOv5s, with a reasonable model size of 14MB and an inference time of 6.4ms, achieves a good mean average precision (mAP) of 37.2 on the COCO dataset. Consequently, YOLOv5s is a practical solution for real-time e-waste classification, providing precise outcomes without compromising processing speed. Hence, we will implement YOLOv5s as the chosen model for this research.

3.5.3. YOLOv5 Model Training

To train YOLOv5 effectively, it is necessary to follow several essential steps. The first step involves installing and importing the required dependencies, such as Torch, TorchVision, and TorchAudio, to ensure all the necessary packages and libraries are available and properly configured. These dependencies are crucial for implementing deep learning tasks and the YOLO algorithm, providing the tools and capabilities for successful YOLOv5 training.

The next step is to clone the YOLOv5 repository from GitHub, which grants access to the YOLOv5 architecture and pre-trained weights. After cloning the repository, navigate to the repository directory and install the required dependencies listed in the requirements text file.

Creating a custom dataset is another critical step in the process. To accomplish this, utilize the 'yaml' format designed explicitly for the e-waste classification system. The 'dataset.yaml' file defines important details like the paths for training and validation data and the class labels. This systematic approach aids in identifying and handling the objects of interest, such as e-waste and non-e-waste.

Once the dataset is prepared, train the YOLOv5 model within the YOLOv5 directory. Hyperparameter tuning is crucial to optimize the training settings and improve the model's accuracy. Parameters like learning rate, batch size, and weight decay can be fine-tuned to align the model with the specific dataset and task.

It is important to note that, in this study, the dataset is cropped to an image size of 640x640 pixels during the hyperparameter tuning process. This cropping is done to align with the employed augmentation technique and optimize the training process. Setting the image size hyperparameter to 640 pixels allows the YOLOv5 model to handle the dimensions and learn from the dataset effectively. Training at the native resolution of 640 ensures compatibility with the YOLOv5 architecture and helps optimize performance during the training process.

By following these steps and considering the specific requirements and characteristics of the dataset, the YOLOv5 model can be trained effectively for accurate object detection and classification in the e-waste classification study.

4. Experimental Results and Discussions

4.1. Experimental Key Indicators

In evaluating the performance of the YOLOv5 model of this e-waste classification study, several metrics are considered, including Precision, Recall, and mAP (mean average precision). These metrics collectively provide valuable insights into the model's detection capabilities and ability to balance precision and recall.

Precision is calculated using the following equation:

$$Precision = \frac{TP(True\ Positive)}{TP(True\ Positive) + FP(Fales\ Positive)} = \frac{TP(True\ Positive)}{All\ Detections} \tag{1}$$

The recall is calculated using the following equation:

$$Recall = \frac{TP(True\ Positive)}{TP(True\ Positive) + FN(Fales\ Nagative)} = \frac{TP(True\ Positive)}{All\ Ground\ Truths} \tag{2}$$

Precision is a metric that quantifies the accuracy of the model's detection results. It represents the percentage of correctly detected objects among all the detected results. On the other hand, recall measures the model's ability to identify positive instances correctly when they are present in the input. In simpler terms, it indicates how well the model captures the relevant objects. Furthermore, True Positive (TP) refers to the number of objects correctly detected by the model. False Positive (FP) occurs when the model incorrectly detects an object from a different class. In essence, it represents a false detection. False Negative (FN) signifies an object that should have been detected but was missed by the model. Lastly, True Negative (TN) denotes the absence of an object that should not be detected.

Mean Average Precision (mAP) is calculated using the following equation:

$$AP = \frac{1}{|QR|} \Sigma \sum_{q=QR} AP(q) \tag{3}$$

When evaluating the YOLOv5 model training, the mean Average Precision (mAP) is a significant metric to consider. The mAP is widely used to assess the performance of object detection models. It measures the precision-recall trade-off across different confidence thresholds during object detection.

By adjusting the confidence threshold, a precision-recall curve is generated. The area under this curve gives each class's Average Precision (AP) value. Taking the mean of these AP values across all classes provides the mAP score, which gives an overall evaluation of the model's detection performance. The mAP considers precision and recall, ensuring a balanced assessment of the model's capabilities. Higher mAP scores indicate a better object detection proficiency of the YOLOv5 model.

This study uses two specific metrics, namely "mAP@.5" and "mAP@.5:.95," to evaluate the YOLOv5 system. The "mAP@.5" calculates the mean Average Precision at a single confidence threshold of 0.5, assessing the model's ability to accurately detect objects with a moderate confidence level while considering the precision and recall trade-off at this threshold. On the other hand, the "mAP@.5:.95" measures the mean Average Precision over a range of confidence thresholds from 0.5 to 0.95, with a step size of 0.05. This metric comprehensively evaluates the model's performance across different confidence levels, considering the varying precision and recall values at each threshold and offering a more detailed assessment of the model's detection capabilities.

4.2. E-waste Classification Results

In this study, we have conducted hyperparameter tuning for the e-waste classification system. The selected hyperparameters include an image size of 640, a batch size of 16, and a total of 100 epochs. These values have been shown to yield the best results in our experiments.

4.2.1. YOLOv5 Batch Size

Regarding batch size, it is essential to consider the specific requirements and available resources. Larger batch size can enhance training efficiency by enabling the model to process more images in parallel, making better use of the computational resources. However, it is crucial to note that a larger batch size also requires more GPU memory. If the available GPU memory is insufficient, it may be necessary to reduce the batch size accordingly.

Determining the optimal batch size depends on various factors, including the available hardware resources, the size of the input images, and the complexity of the model. In the study, values 8 and 16 have been implemented to test the accuracy of the YOLOv5 model depending on the hardware capabilities to assess the accuracy of the YOLOv5 model based on our hardware capabilities. Table 6 summarizes the results of YOLOv5 model training with different batches.

Table 6. Results for YOLOv5 Model Training with Different Batch (P-Precision, R-Recall)

Batch	Image Size	Epochs	P	R	mAP@.5	mAP@.5:.95
16	640	5	0.7366	0.6682	0.7312	0.4571
8	640	5	0.5489	0.6152	0.5697	0.3117

For batch size 16, the model achieved a precision of 0.7366, indicating that approximately 73.66% of the detected objects were correctly identified. The recall value, representing the proportion of correctly detected objects out of the real ground truth objects, was 0.6682. The mean Average Precision (mAP) at a confidence threshold of 0.5 was 0.7312, demonstrating the model's ability to detect objects across different classes accurately. The mAP at a stricter confidence threshold of 0.5 to 0.95 was 0.4571, reflecting the model's performance in detecting higher-confidence objects.

In contrast, when the batch size was reduced to 8, the precision decreased to 0.5489, indicating a lower percentage of correctly detected objects. The recall value increased slightly to 0.6152, implying a better proportion of correctly detected objects out of the ground truth. The mAP at a confidence threshold of 0.5 dropped to 0.5697, indicating a decrease in overall detection performance. Similarly, the mAP at the stricter confidence threshold of 0.5 to 0.95 decreased to 0.3117.

The results illustrate the relationship between the batch size and precision, recall, and mAP performance metrics. Generally, a larger batch size can lead to higher precision, recall, and mAP scores, indicating better object detection capabilities. However, it is important to consider the trade-off between batch size and computational resources, which batch size of 16 takes longer to train the model. Optimal performance can be achieved by carefully selecting the batch size based on the specific requirements and limitations of the training process.

4.2.2. YOLOv5 Epochs Number

The determination of the number of epochs is influenced by several factors, including the dataset's complexity, the convergence rate of the model, and the available computational resources, according to a study conducted by (Ieamsaard et al., 2021). It is essential to consider these factors when selecting the appropriate number of epochs for training the model. When deciding on the number of epochs, the following considerations can be considered:

- Dataset Size: The size of the dataset influences the number of epochs needed for the model to learn from the data effectively. Larger datasets may require more epochs for the model to capture the underlying patterns accurately. On the other hand, smaller datasets run the risk of overfitting if trained for too many epochs.
- Model Convergence: Monitoring the training progress and performance metrics, such as loss and accuracy, helps us understand the model's convergence. When the performance metrics

stabilize or show minimal improvement, it indicates that the model has reached a point of convergence. In such cases, training for additional epochs may not yield significant benefits.

- Computational Resources: Considering the available computational resources is crucial since increasing the number of epochs directly affects the training time. Striking a balance between the number of epochs required for convergence and the available resources ensures an efficient training process without unnecessary delays.
- Experimentation and Validation: Conducting experiments with different numbers of epochs is valuable to identify the optimal value for the specific task at hand. We can make informed decisions about further training requirements by training the model for a defined number of epochs, evaluating its performance on a separate validation set, and analyzing the results.

In this study, the YOLOv5 model is trained using 100 epochs, keeping the remaining hyperparameters constant, as shown in Table 7.

Table 7. Results for YOLOv5 Model Training with Different Epochs (P-Precision, R-Recall)

Epochs	Image Size	Batch	P	R	mAP@.5	mAP@.5:.95
1	640	16	0.3430	0.5382	0.3276	0.1838
10	640	16	0.8013	0.7284	0.8002	0.5412
20	640	16	0.7585	0.6807	0.7586	0.5018
30	640	16	0.8508	0.7661	0.8452	0.6249
40	640	16	0.8699	0.8049	0.8676	0.6690
50	640	16	0.8668	0.8122	0.8706	0.6690
60	640	16	0.8983	0.8264	0.8810	0.6893
70	640	16	0.9028	0.8219	0.8806	0.7006
80	640	16	0.9080	0.8322	0.8943	0.7147
90	640	16	0.8979	0.8309	0.8833	0.7146
100	640	16	0.9032	0.8399	0.8889	0.7218

Upon analyzing the results in Table 7, it becomes evident that the model achieves the highest performance when trained for 100 epochs. As the number of epochs increases, there is a consistent improvement in precision, recall, "mAP@.5," and "mAP@.5:.95." This improvement signifies the model's enhanced ability to minimize false positive detections and offers more precise predictions. It is worth noting that the recall is slightly lower in 100 epochs compared to 80 epochs, but the "mAP@.5:.95" metric is higher. This implies that the model becomes more adept at accurately detecting objects with varying degrees of overlap, spanning a range of Intersections over Union (IoU) thresholds. Additionally, increasing the number of epochs beyond 100 either maintains or slightly improves accuracy, but at the cost of significantly longer training time. Therefore, it is advisable to balance achieving satisfactory accuracy and optimizing training time by selecting an appropriate number of epochs for the e-waste classification task.

4.2.3. Visualization of Pre-trained YOLOv5 Model

Once the YOLOv5 training process is completed, we can use visualization techniques to understand how the model performs and evaluate its effectiveness. In this study, we used different visualizations to analyze and showcase our results. These visualizations helped us gain valuable insights into the model's performance and allowed us to present our findings clearly and understandably.

Fig. 4 provides a precision curve that shows the precision values at different confidence thresholds. Precision measures how many correct optimistic predictions the model made from all instances predicted as positive. This curve helps us understand the relationship between precision and confidence thresholds, and it helps us determine the best threshold for making predictions. Looking at the diagram, the result "all classes 1.00 at 0.993" means that at a specific confidence threshold of 0.993, the model achieved a precision of 1.00 (or 100%) for all classes. This indicates that the YOLOv5 model performed exceptionally well, showing accurate and reliable class predictions.

In Fig. 5, we can see the precision-recall curve, which shows how the model performs in precision and recall at different confidence thresholds. It gives us insights into how well the model balances precision and recall. The result "all classes 0.889 mAP@0.5" indicates that the model achieved an average precision of 0.889 across all classes at a specific Intersection over the Union (IoU) threshold of 0.5. This means that the model's predictions were accurate and reliable, achieving a relatively high average precision for all classes. The mAP value of 0.889 suggests that the YOLOv5 model effectively balanced precision and recall during object detection, resulting in accurate predictions with minimal false positives.

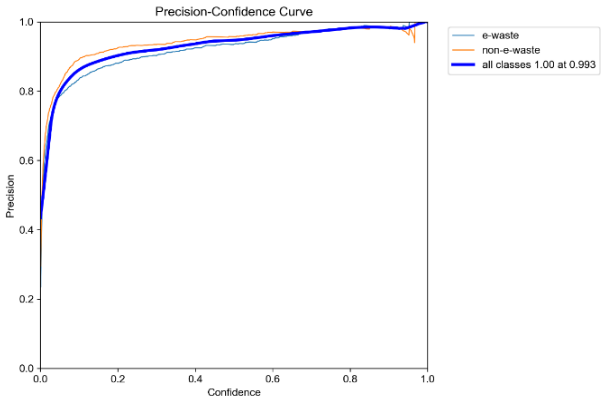

Fig. 4. Precision Curve for YOLOv5 Model

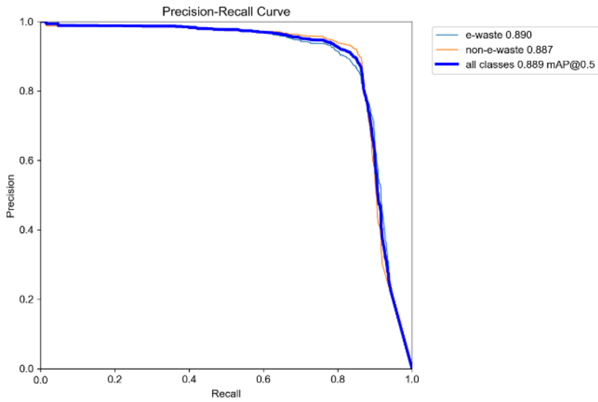

Fig. 5. Precision-recall Curve for YOLOv5 Model

In Fig. 6, the visualization includes metrics that show the model's performance and improvement over epochs. In the top section, we have metrics related to training, such as Box Loss, Object Loss, Class Loss, Precision, and Recall. As the epochs increase, the losses decrease, indicating improved accuracy in localizing objects, detecting objects, and classifying them correctly. Increasing trends in Precision and Recall suggest better optimistic predictions and the ability to capture positive instances accurately. Moving to the bottom section, we have metrics related to validation. Validation Box Loss and Object Loss also show decreasing trends, indicating improved accuracy in localizing and detecting objects during validation. Validation Class Loss decreases, showing improved accuracy in assigning the correct class labels to detected objects. The metric mAP_0.5 represents the mean Average Precision at an IoU threshold of 0.5 during validation, and an increasing trend in this metric indicates improved detection accuracy at this threshold. Finally, the metric mAP_0.5:0.95 represents the mean Average Precision across a range of IoU thresholds, showing improved performance in handling objects with varying levels of overlap.

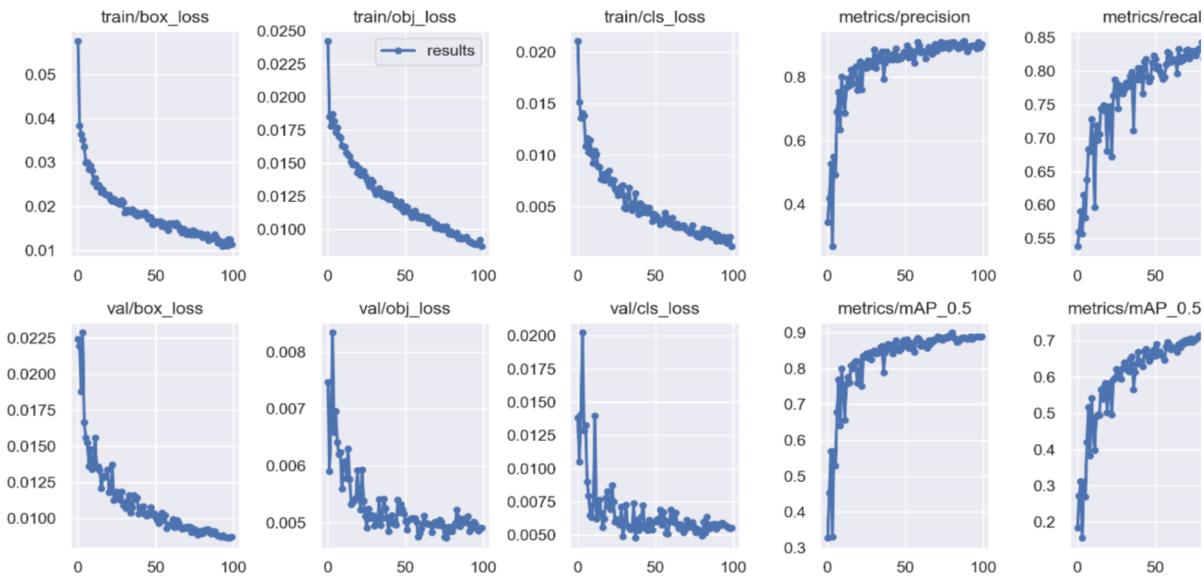

Fig. 6. Training Results for YOLOv5 Model

4.2.4. Testing of E-waste Classification System

During the testing phase of the E-waste classification system, Figures 7 to 11 showcase the detection outcomes for five different test results. The first two test results demonstrate the system's successful identification of e-waste items, specifically a phone and a keyboard. The system accurately recognizes non-e-waste objects, namely metal cans, and papers, in the following two test results. The fifth test result demonstrates its capability to detect both e-waste and non-e-waste simultaneously. These results collectively highlight the system's high level of accuracy in effectively categorizing various scenarios for sustainable recycling.

Fig.7. Test Result 1

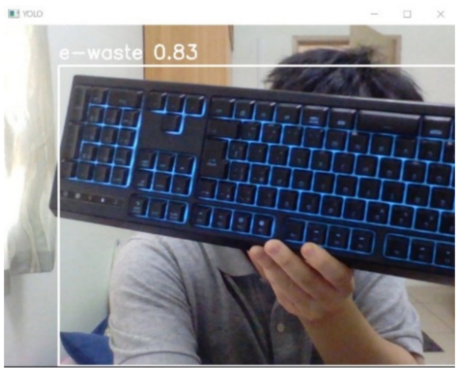

Fig.8. Test Result 2

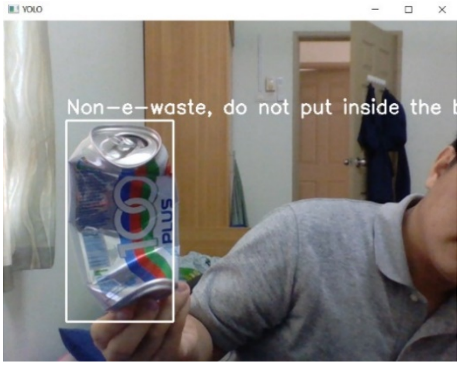

Fig.9. Test Result 3

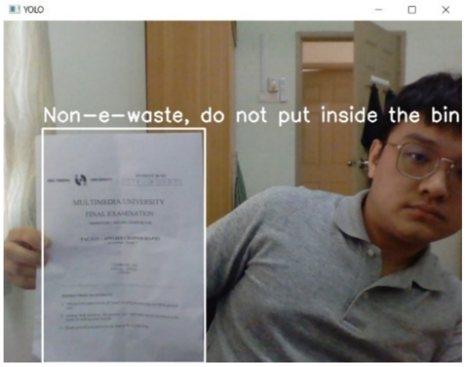

Fig.10. Test Result 4

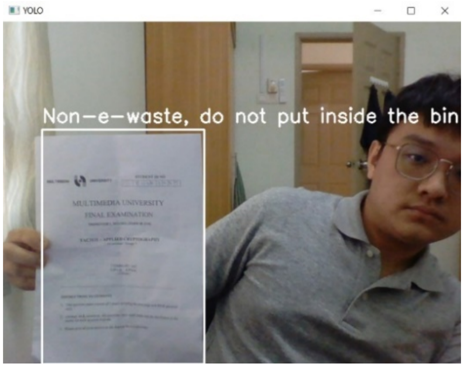

Fig.11. Test Result 5

5. Conclusion

The main goal of this study is to increase MMU students' awareness regarding recycling e-waste for sustainable recycling. The proposed system allows the user to classify e-waste into two categories, non-e-waste, and e-waste, using the YOLOv5 model. Upon completing the training process with 100 epochs and a batch size 16, the YOLOv5 model yielded the following performance metrics: Precision: 0.9032, Recall: 0.8399, mAP@.5: 0.8889, and mAP@.5:.95: 0.7218. Overall, the YOLOv5 performs well as the model has a low false negative rate and can effectively capture many positive instances. Subsequently, the pre-trained YOLOv5 model is integrated into Flask to develop a user interface that enables users to perform two primary actions: uploading images for object detection and initiating live detection functionality. Once the system detects an input of e-waste, the system will proceed to store the corresponding image in the dataset, which aims to continuously enhance the model's accuracy by accumulating more data from the smart bin.

In future work, we will gather additional data from various angles and include multiple classes to solve the system limitation of the camera detection that sometimes fails to identify waste that is too far or too close. This will enable the system to have a more comprehensive set of reference data and reduce instances of missed detection. Additionally, we aim to expand the dataset to include more diverse types of e-waste, allowing the system to learn patterns and classify e-waste into more specific categories. In conclusion, the focus will be on collecting a more comprehensive range of data to facilitate better learning and more precise categorization of e-waste within the system.

References

Adedeji, O., & Wang, Z. (2019). Intelligent waste classification system using deep learning convolutional neural network. *Procedia Manufacturing*, *35*, 607–612. https://doi.org/10.1016/j.promfg.2019.05.086

Dassi, R., Kamal, S., & Babu, B. S. (2021). E-Waste Detection and Collection assistance system using YOLOv5. CSITSS 2021 - 2021 5th International Conference on Computational Systems and Information Technology for Sustainable Solutions, Proceedings. https://doi.org/10.1109/CSITSS54238.2021.9682806

Electronics Object Image Dataset | Computer Parts | Kaggle. (n.d.). Retrieved December 15, 2022, from https://www.kaggle.com/datasets/dataclusterlabs/electronics-mouse-keyboard-image-dataset

Fieno Silva, A., Soares, E. A., Costa, B. S., S Bernardes, A. C., A Pereira, J. V, Zampa, V. H., Pereira, V. A., Matos, G. F., Soares, C. L., & Silva, A. F. (2018). Artificial Intelligence in Automated Sorting in Trash Recycling. *XV Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*. https://www.researchgate.net/publication/326994757

Ieamsaard, J., Charoensook, S. N., & Yammen, S. (2021). Deep Learning-based Face Mask Detection Using YoloV5. *Proceeding of the 2021 9th International Electrical Engineering Congress, IEECON 2021*, 428–431. https://doi.org/10.1109/IEECON51072.2021.9440346

Kushal Samir Mehta, Nishant Bhattacharya, Jyotsna Prasad, Kaala Lakshmi .S, K. V. Subramaniam, & Dinkar Sitaram. (2017). Identifying Uncollected Garbage in Urban Areas using Crowdsourcing and Machine Learning. *IEEE Region 10 Symposium (TENSYMP)*.

Majchrowska, S., Mikołajczyk, A., Ferlin, M., Klawikowska, Z., Plantykow, M.A., Kwasigroch, A., Majek, K. (2022). Deep Learning-Based Waste Detection in Natural and Urban Environments, *Waste Management*, Vol. 138, pp. 274-284, https://doi.org/10.1016/j.wasman.2021.12.001.

Microcontroller Detection | Kaggle. (n.d.). Retrieved December 15, 2022, from https://www.kaggle.com/datasets/tannergi/microcontroller-detection

Nasik Sami, K., Md Afique Amin, Z., & Hassan, R. (2020). Waste Management Using Machine Learning and Deep Learning Algorithms. In *International Journal on Perceptive and Cognitive Computing (IJPCC)* (Vol. 6, Issue 2).

Ng, J.J., Goh, K.O.M., & Tee, C. (2023). Traffic mpact Assessment System using Yolov5 and ByteTrack. *Journal of Informatics and Web Engineering*, 2(2), 168 – 188.

Poo, Z.Y., Ting, C.Y., Loh, Y.P., & Ghauth, K.I. (2023). Multi-Label Classification with Deep Learning for Retail Recommendation. *Journal of Informatics and Web Engineering*, 2(2), 218 – 232.

Qiao, Y., Zhang, Q., Qi, Y., Wan, T., Yang, L., Yu, X. (2023). A Waste Classification Model in Low-Illumination Scenes Based on Convnext, Resources, *Conservation and Recycling*, Vol. 199, 107274, ISSN 0921-3449, https://doi.org/10.1016/j.resconrec.2023.107274.

Sakr, G. E., Mokbel, M., Darwich, A., Khneisser, M. N., & Hadi, A. (2016). Comparing deep learning and support vector machines for autonomous waste sorting. *2016 IEEE International Multidisciplinary Conference on Engineering Technology, IMCET 2016*, 207–212. https://doi.org/10.1109/IMCET.2016.7777453

Satvilkar, M. (2018). Image-Based Trash Classification using Machine Learning Algorithms for Recyclability Status.

Starter: e-waste dataset 93b07fb8-a | Kaggle. (n.d.). Retrieved December 15, 2022, from https://www.kaggle.com/code/kerneler/starter-e-waste-dataset-93b07fb8-a/data

Wu, T., Zhang, H., Peng, W., Lü, F., He, P. (2023). Applications of Convolutional Neural Networks for Intelligent Waste Identification and Recycling: A Review, *Resources, Conservation and Recycling*, Vol. 190, 106813, ISSN 0921-3449, https://doi.org/10.1016/j.resconrec.2022.106813.

Yang, G., Jin, J., Lei, Q., Wang, Y., Zhou, J., Sun, Z., Li, X., & Wang, W. (2021). Garbage Classification System with YOLOV5 Based on Image Recognition. *2021 6th International Conference on Signal and Image Processing, ICSIP 2021*, 11–18. https://doi.org/10.1109/ICSIP52628.2021.9688725

Yang, M., & Thung, G. (2016). Classification of Trash for Recyclability Status.

Zhang, Q., Yang, Q., Zhang, X., Wei, W., Bao, Q., Su, J., Liu, X. (2022). A Multi-Label Waste Detection Model Based on Transfer Learning, *Resources, Conservation and Recycling*, Vol. 181, 106235, ISSN 0921-3449, https://doi.org/10.1016/j.resconrec.2022.106235.