Real-Time Waste Classification Modeling for Embedded Systems Based on Deep Convolutional Neural Networks

SORAWIT THOKRAIRAK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR MASTER DEGREE OF SCIENCE

IN DATA SCIENCE

FACULTY OF INFORMATICS

BURAPHA UNIVERSITY

2022

การพัฒนาโมเดลในการจำแนกขยะสำหรับระบบสมองกลฝังตัวบนพื้นฐานของโครงข่ายประสาทแบบคอนโวลูชันเชิงลึก

สรวิชญ์ โตไกรรักษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการข้อมูล
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
2565

Real-Time Waste Classification Modeling for Embedded Systems Based on Deep Convolutional Neural Networks

SORAWIT THOKRAIRAK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR MASTER DEGREE OF SCIENCE
IN DATA SCIENCE
FACULTY OF INFORMATICS
BURAPHA UNIVERSITY
2022

The Thesis of Sorawit Thokrairak has been approved by the examining committee to be partial fulfillment of the requirements for the Master Degree of Science in Data Science of Burapha University

Advisory Committee

Principal advisor

....

(Prajaks Jitngernmadan)

Examining Committee

.... Principal examiner
(Assistant Professor Rattana Wetprasit)

.... Member
(Assistant Professor Krisana Chinnasarn)

.... Member
(Assistant Professor Prajaks Jitngernmadan)

.... Dean of the Faculty of Informatics
(Assistant Professor Krisana Chinnasarn)
....

This Thesis has been approved by Graduate School Burapha University to be partial fulfillment of the requirements for the Master Degree of Science in Data Science of Burapha University

.... Dean of Graduate School
(Associate Professor Dr. Nujjaree Chaimongkol)
....

63910159: MAJOR: DATA SCIENCE; M.Sc. (DATA SCIENCE)

KEYWORDS: Classification, SSD-MobileNet, Embedded System, COCO, AI Model

SORAWIT THOKRAIRAK : REAL-TIME WASTE CLASSIFICATION MODELING FOR EMBEDDED SYSTEMS BASED ON DEEP CONVOLUTIONAL NEURAL NETWORKS. ADVISORY COMMITTEE: PRAJAKS JITNGERNMADAN, Ph.D. 2022.

Machine Learning technology grows in the field of automatic waste sorting machines equipped with an intelligent unit. This intelligent unit runs on an embedded system that mostly has lower computation power (both CPU and GPU) and lower RAM. However, to archive a higher accuracy rate on classification one has to use a sophisticated classification AI model, which needs less computational power. We considered our experiment using 3 pre-trained models based on COCO dataset, namely the ssd_mobilenet_v2_coco, the ssd_inception_v2_coco, and the ssd_resnet_50_fpn_coco due to their good quality. The results show that although the AI model based on the ssd_resnet_50_fpn_coco has the highest accuracy (99.75%), it consumes the most computational power. In contrast, the one based on the ssd_mobilenet_v2_coco has acceptable accuracy (98.83%) and it consumes the lowest computational power. We decided that the most suitable AI model for embedded systems is the one that is trained with the pre-trained ssd_mobilenet_v2_coco model.

ACKNOWLEDGEMENTS

This work is done at Digital Media and Interaction Research Laboratory (DMI) at Faculty of Informatics, Burapha University. We are deeply grateful for the grant and supports from Faculty of Informatics.

Sorawit Thokrairak

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1
INTRODUCTION

Problem Descriptions

Another factor affecting waste management in Thailand is the problem of people who do not understand the importance of garbage separation. Because most individuals are still unaware of the need of public awareness, these wastes contribute to environmental issues. According to a survey of the Bangsaen beach area in Chon Buri province, Thailand, which is a popular tourist site with a large number of visitors, the majority of the rubbish problems are created by visitors. The majority of garbage bins in Thailand are unclassified, making it impossible for dumpers to differentiate rubbish from the source. This issue is exacerbated by the fact that most individuals are unaware of the need for trash separation. The concept of automatic waste sorting is another technique to use artificial intelligence technology to successfully address the problem of waste sorting. This technology will manage in identifying the type of garbage and directing the waste sorting machine. The idea of using artificial intelligence to sort waste will allow us to make it into an automated trash can or a factory waste sorting system. However, the fundamental issue with applying artificial intelligence to automated garbage cans is the requirement for a large amount of computer power in order for artificial intelligence to execute the work of waste separation. As a result, we aim to merge artificial intelligence technology with embedded systems in this study. An embedded system is a device that may be utilized to make a waste sorting machine since it contains a processor. It enables artificial intelligence to be processed by the system. Because embedded systems can be utilized instead of computers, they can lower the cost of manufacturing garbage sorting equipment. Furthermore, embedded systems are smaller devices than computers, making them easier to use. Contributing to the development of a variety of automatic machinery.

Statement of the Problem

The embedded system's processing efficiency is inferior to that of a computer, we must select and create an artificial intelligence model that is appropriate for the embedded system. In general, an embedded system consists of a limited computational power chip as known as central processing unit (CPU) and random memory access (RAM). These components are packed in a tiny chip called microprocessor and being used in simple electronic systems. In some best cases, they may have a graphical processing unit (GPU) with extra cost. This general embedded system environment is common in most devices ranging from washing machines to vending machines. Up until now, their usage and capability are suitable for consumer electronic devices and they function smoothly without any problems. However, the demand in Deep Learning (DL) and Artificial Intelligence (AI) is getting higher and higher. Nowadays, the people want to put a somehow intelligent agent into a system and let him do the dirty work for them, especially when it comes to a complex decision making with tons of data. Even in a device equipped with an embedded system should be "upgraded" with an AI model. In terms of being able to use an AI model, a device has to have the high computational power, i.e., high performance in CPU and RAM. And in the best case, a device has to have a GPU, too. These requirements are, of course, against nature of a traditional embedded system. Even though there are powerful embedded systems in the market, such as Raspberry Pi, Banana Pi, ASUS Tinker Board 2S, etc., or even the Jetson Nano Developer Kit (Corporation, 2022), which claims to support multiple neural networks running in parallel, the AI model still has to be small and run on limited computational power in case we want to archive the best results.

Research Objectives

1. To find the best AI model for waste separation that run on embedded systems.
2. To compare object detection models used in Single Shot Detector concept.

Research Questions

How to separate solid wastes based on object detection using the best AI model that can run on embedded systems effectively.

Research Methodology

In another work, we attempt to invent an automatic waste sorting bin based on an AI model. The target is to separate 6 types of waste using image classification and object detection. The 6 types of waste are plastic bottles, glass bottles, metal cans, plastic bags, food containers, and plastic coffee cups. These 6 types of waste are mostly found in the near of Bangsaen beach, Chon Buri, Thailand. Here we face the waste problem, especially the recycle-able plastic waste that is littered in the same bin with other wastes. To solve this problem in sophisticated way, we invent an automatic waste sorting bin based on an embedded system equipped with a suitable AI model. This AI model is based on image classification and object detection. The embedded system we proposed is the Jetson Nano mentioned above.

This work focuses on finding the most suitable AI model that can be used on an embedded system (Jetson Nano in this case). We selected 3 pre-trained models based on COCO dataset due to their good quality and widely use. The 3 models are ssd_mobilenet_v2_coco, ssd_inception_v2_coco, and ssd_resnet_50_fpn_coco (Corporation, 2022) . Each pre-trained model will be trained with the prepared dataset of those 6 types of waste. The best results of accuracy rate will be identified, along with other parameters such as frame rate (frame per second: fps) running on Jetson Nano, CPU usage, GPU usage, and RAM usage. The most suitable AI model will be selected. This suitable model has to fulfil the requirements of embedded systems, namely archiving the highest accuracy rate while running on a system with limitation of computational power and architecture. The dataset of 6 types of the waste is organized into 1,000 images each, i.e. 6,000 images altogether. These 1,000 images for each waste type are divided into 800 images (80%) for the training phase and 200 images (20%) for the testing phase.

Threats to Validity

The data used in the training was collected only images of clean, uncontaminated and properly shaped waste. This model may not be suitable for separating dirty or malformed waste.

This model was developed under one piece-by-piece waste detection condition. This model produces good results only when used in conjunction with an application model for separating waste one piece at a time.

Work phases and Timeline

Table 1 Timeline of the research

Tasks	Time Periods							
	Year1				Year 2			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
1. Problem Definition	↔	↔	↔					
2. Literature Review	↔	↔	↔	↔	↔			
3. Methodology and Solution Conception Framework Design and Testing		↔	↔	↔				
4. Proposal Presentation and Approval					↔			
5. Institutional Review Board (IRB) Submission and Approval						↔		
6. Data Collection and Analysis							↔	
7. Test, Feedback, and Improvement							↔	
8. Academic International Conference Participation and Publication Presentation							↔	
9. Thesis Defense Examination								↔

CHAPTER 2
THEORIES AND RELATED WORKS

State of the Art

The work of Melinte et al. is proposed to develop object detectors for the municipal waste identification based on deep convolutional neural network (CNN) (Melinte, Travediu, & Dumitriu, 2020). In this work, they tried to identify the waste in real-time. The proposed recycling wastes are paper, metal, plastic, and glass. The tested models were: SSD-MobileNetV2, Faster R-CNN, MobileNet, MobileNetV2, and InceptionV2. The results show that there are 3 models that have following accuracy: the accuracy of SSD-MobileNetV2 is 97.63%, the accuracy of ResNet50 is 87% and the accuracy of InceptionV2 is 81.6%. These models show the acceptable accuracy rate, especially the SSD-MobileNetV2 one. However, the implementation of these models requires also a high-performance computer.

There is also research that talks about SSD-MobileNet in use with the subject of dental instruments that is Ali et al. researched on dental instrument recognition based on SSD-MobileNet object recognition (Ali, Khursheed, Fatima, Shuja, & Noor, 2019). They proposed to solve the problem of identifying the real-time object recognition of dental instruments by used SSD-MobileNet model. The task is to do the object recognition for dental instruments like spatula, elevator, mouth mirror, etc. They achieved the accuracy of 98.8%. However, the results have some limitations because small objects are difficult to detect. In order to create a model for the classification of waste, there is another research that uses a model like ResNet50.

Adedeji et al. published a work related to an intelligent waste classification system based on convolution neural network (CNN) (Adedeji & Wang, 2019). Their work concentrated on a waste classification system that is able to separate different types of waste using ResNet50 as a waste classification model. The result of the trained model has the accuracy of 87%. The implementation of such a system requires a high-performance computer with high processing power. Therefore, this model is not suitable for small devices such as mobile phones or embedded systems.

In the work of Y. Pang, Y. Yuan, X. Li, and J. Pan suggested to Histograms of Oriented Gradients (HOG) [3]. This research's presentation of the effectiveness of human detection algorithms is an intriguing aspect. This algorithm can detect humans regardless of the lighting conditions and backgrounds of indoor and outdoor images. This demonstrates that the algorithm can function in environments other than those specified when it was created.

Another work from R. G. Dawod and C. Dobre (Dawod & Dobre, 2022) is related to Classification of plant diseases based on images using CNN. This paper discusses the factor that affects classification, specifically light in field tests. Compared to the laboratory results of this study, the model's accuracy was diminished by the light and shadow factors.

The work from C. Bircanoglu, M. Atay, F. Beser, O. Genc, and M. A. Kizrak employed a model known as a RecycleNet (Bircanoğlu, Atay, Beşer, Ö, & Kızrak, 2018). Before this research, a model for the waste generation was selected. They compared them with other CNN models and concluded that the RecycleNet model performed better than the other models. The RecycleNet model provides a waste separation accuracy of up to 95%, and they also discussed the factors that affect their waste classification accuracy: environmental factors such as lighting.

The work of R. A. Aral, S. R. Keskin, M. Kaya, and M. Haciomeroglu concentrated on classifying recyclable garbage (Aral, Keskin, Kaya, & Hacıömeroğlu, 2018) . They employed a model known as a Densenet121, DenseNet169, InceptionResnetV2, MobileNet (Howard et al., 2017) , and Xception. The experimental findings of this study indicated that DenseNet121 is 95% accurate. The study also discussed the background of the training data, that is some real-time models produced less accurate results because the majority of the images used have white backgrounds, as well as datasets that may be too small.

The work of Z. Chunxiang, Q. Jiacheng, and W. Binrui (Chunxiang, Jiacheng, & Wang, 2022) employed a model known as a YOLOX and YOLOv5. They classified the detections into seventeen categories. The 5,000 images were utilized in this research for all of the image practices. This study presented a real-world test, it was concluded that the model would perform admirably when an object has no obstructions. In the real world, the lighting, shadows, and angles of the images also contributed to the model's identification

Gap Identification

The development of the waste classification model in the past was the use of the model on the computer device, which has high computing efficiency, but in the application to create a smart trash can if using a computer, it will cost a lot to build. In this research, there is also a difference in developing a model to classify waste to be able to use on embedded systems and can be applied to create a smart bin. This can effectively reduce the cost of building a smart bin.

Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) in this context can be understood as a simulation of the human vision. The principle work of the Convolutional Neural Network is to divide the image into pixels and then combines these groups of pixels together, in order to analyze the matrix of the image. If the image is black and white, the matrix value is 2x2. However, if the image is a colorized image, the matrix value is 3x3. The CNN consists of 4 steps, which are Step 1 Convolution; in this step, the matrix between the input image and feature detector will be multiplied and the result is the feature map. When multiplying all the matrices, many feature maps will be created. These feature maps are called convolutional layer. Step 2 Max pooling; this step is to filter the maximum values in the same matrix area to get a pooled feature map. Step 3 Flattening; this step is a feature map pooling. The feature map is obtained from the previous step. The data is stored into a single column in terms of easy data analysis. Step 4 Full connection; in this step the flattening will be brought into the Deep Learning model and then results the output model.

Dataset COCO

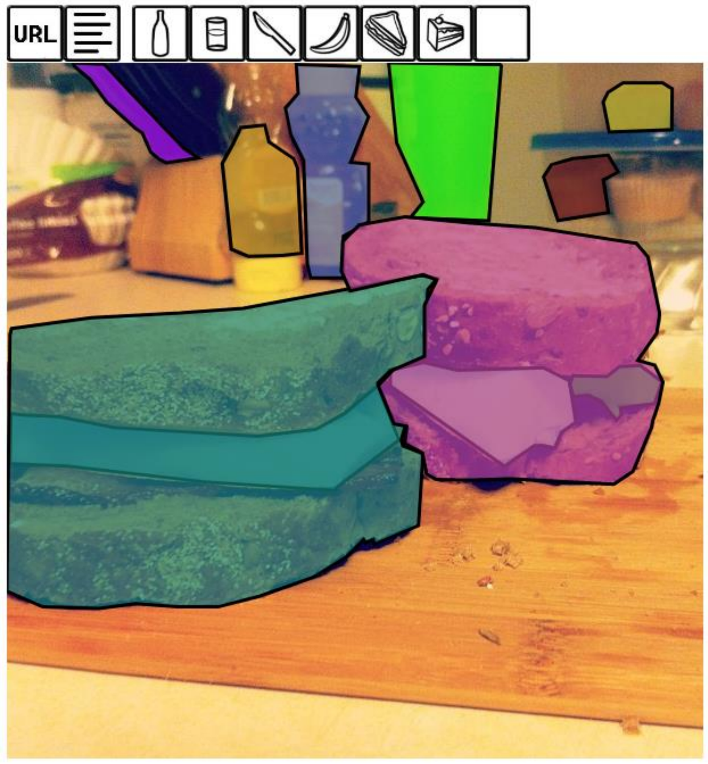

Figure 1Example image from dataset coco (1)

Figure 2 Example image from dataset coco (2)

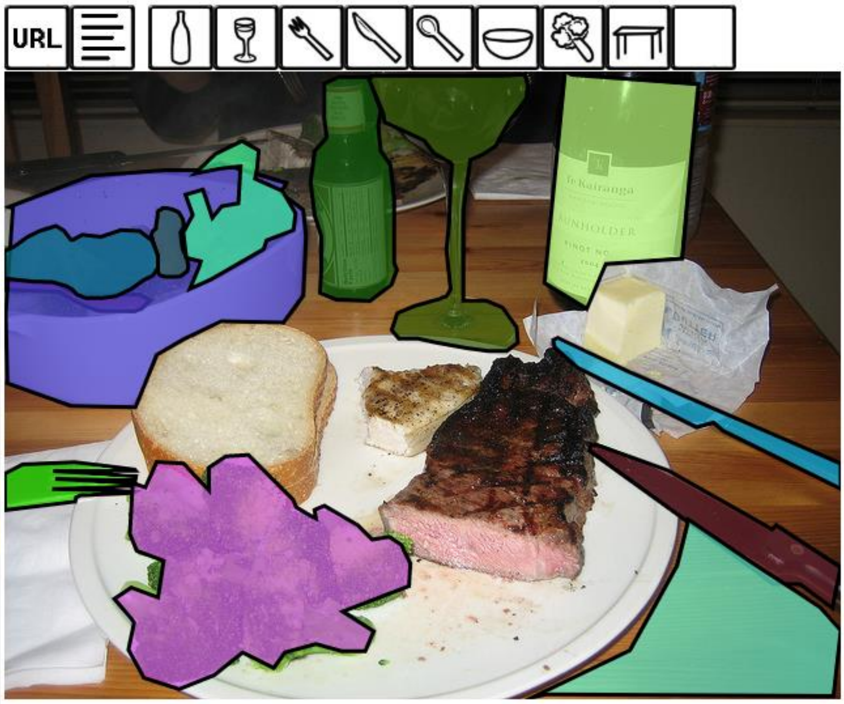

Figure 3 Example image from dataset coco (3)

Figure 4 Example image from dataset coco (4)

Figure 5 Example image from dataset coco (5)

The COCO dataset (Lin et al., 2014), which comprises around 328,000 images with 2.5 million manually segmented object instances and 91 object categories, was created for object detection applications. This is an image dataset that has been categorized. This is certainly beneficial for building additional applications like Object segmentation, Recognition in Context, Superpixel stuff segmentation, and so on. This dataset also keeps the original image and stores descriptions of each data in json files. It's available in png, jpg, and tif formats. The dataset is pre-labeled to the data, which is a key technique for preparing the data before training the model in the evaluation, making it simple to use. COCO datasets are also employed in a wide range of research assignments since they are simple to use and ideal for beginners without their own databases.

SSD-Mobilenet V2

The SSD-MobileNet model (Howard et al., 2017) creation approach is designed for mobile and embedded vision applications. The core layer of this MobileNet is the so-called Depthwise Separable Convolution, which is a form of factorized convolutions. This factorized convolution is a standard convolution for a Depthwise convolution and a 1×1 convolution called a pointwise convolution. In the MobileNet, the Depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs of the Depthwise convolution. Depthwise separable convolutions are a combination of depthwise and pointwise convolutions. This method seeks to make a model lighter while keeping the quantity of information learned in each convolution constant.

SSD-Inception V2

The factorization technique is used in Inception-v2 (Ghoury, Sungur, & Durdu, 2019). To boost computational speed, the 7 x 7 convolution has been factorized into three 3 x 3 convolutions. In the network's inception section, there are three typical inception modules, each with 288 filters. Using the grid reduction approach, the 35 x 35 grid with 288 filters is reduced to a 17 x 17 grid with 768 filters. The factorized inception modules are then repeated five times. It is converted to an 8 x 8 x 1280 grid using grid reduction. There are two Inception modules at the coarsest 8 x 8 level, and each tile has a concatenated output filter bank with a size of 2048. Remove the module's bottleneck filter banks by making them wider rather than deeper.

SSD-Resnet-50-FPN

SSD ResNet-50 FPN (Patel et al., 2021), Unlike other two-stage detectors, the Single Shot Detector (SSD) is one of the most popular and fastest object detection models available, capable of detecting several objects in a single shot. Single-stage detectors, on the other hand, are known to have a severe foreground-background class imbalance problem, which leads to the assumption that these models do not perform well.

As a result, Facebook AI Research unveiled the RetinaNet, a revolutionary one-stage object detector that boosts prediction accuracy by using focal loss. In training, the primary purpose of focused loss is to suppress simple samples and concentrate on hard negative samples. ResNet and Feature Pyramid Network (FPN) serve as the backbone, and two subnetworks for classification and bounding box regression are integrated to form the main model, RetinaNet, which outperforms several prominent two-stage detectors when evaluated on the COCO dataset.

Confusion Matrix for Multi-Class Classification

To identify the accuracy rate of each model, the Confusion Matrix [6–8] is used. In our case, the Confusion Matrix for Multi-Classification is the best option due to the 6 types of waste we proposed. A Confusion Matrix is the way to visualize the performance of the classification model in a tabular view. Each entry in a Confusion Matrix shows the number of predictions that are the results of the model. It indicates that the model classified the classes correctly or incorrectly.

The simple and most known form of a confusion matrix is the confusion matrix for binary classification which contains 2 classes. A confusion matrix that describes the classification performance can be visualized as shown in Figure 6. The number of predictions are visualized in the tabular way, showing relations.

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 6 Contingency table

Where:

- True Positive (TP): TP is the number of predictions where an actual image is classified correctly.

- True Negative (TN): TN is the number of predictions where a not-actual image is classified correctly.
- False Positive (FP): FP is the number of predictions where a not-actual image is classified incorrectly.
- False Negative (FN): FN is the number of predictions where an actual image is classified incorrectly.

This confusion matrix can be used as an evaluation criteria for machine learning model due to its simplicity but very efficient for performance measurement. Some common performance measures includes:

1. Accuracy: This value indicates the overall accuracy of the trained model. It can be calculated by means of the fraction of the total samples that were correctly classified by the classifier divided by the total samples, as following:

$$\frac{(TP + TN)}{(TP + TN + FP + FN)}$$

2. Precision: This value indicates how often the classifier classifies an actual image and the result of the classification is correct. The following formula calculates the precision:

$$\frac{(TP)}{(TP + FP)}$$

3. Recall: This value indicates how often the classifier classifies an actual image as an actual image. The following formula calculates the recall:

$$\frac{(TP)}{(TP + FN)}$$

4. F1-score: This value is a combination of precision and recall into a single measure. It can be described as the harmonic mean of precision and recall in Mathematics. It can be calculated as follows:

$$2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = \frac{2TP}{2TP + FP + FN}$$

5. In our experiment, we proposed 6 waste types which give us 6 classes to be calculated. The confusion matrix with multi-class classification have to be used instead of binary classification. This kind of confusion matrix has no positive or negative classes. The calculation can be done by finding TP, TN, FP, and FN for each individual class. Figure 7 shows an example of a confusion matrix for 3-class classification.

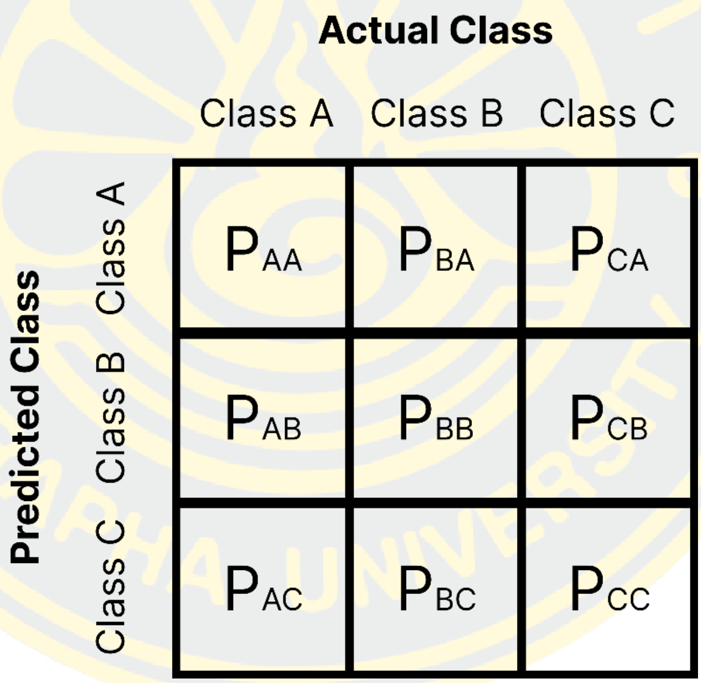

Figure 7 Confusion matrix

The solution for creating a Multi-Class Confusion matrix is similar in principle to a normal Confusion matrix but differs in the way the results are read from the Confusion matrix.

Example:

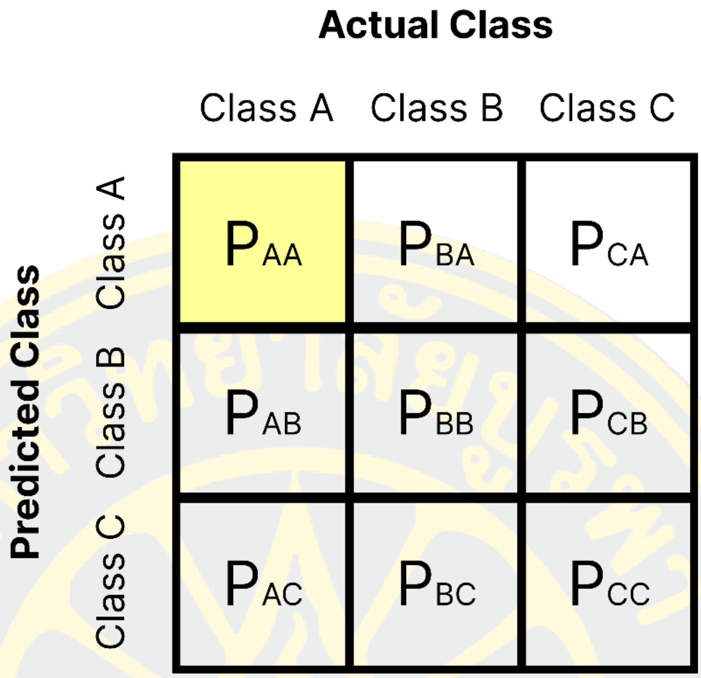

Figure 8 TP value selection

TP of class A = P_{AA}

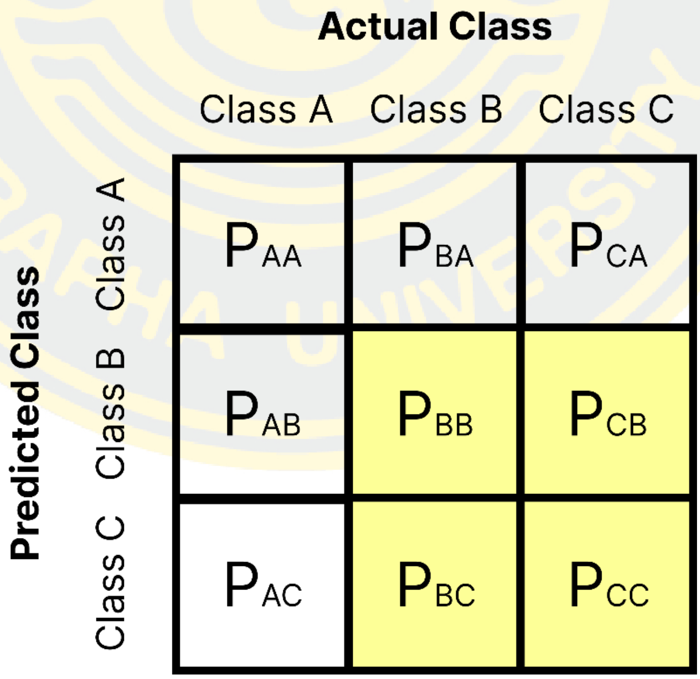

Figure 9 TN value selection

TN of class A = $P_{BB} + P_{CB} + P_{BC} + P_{CC}$

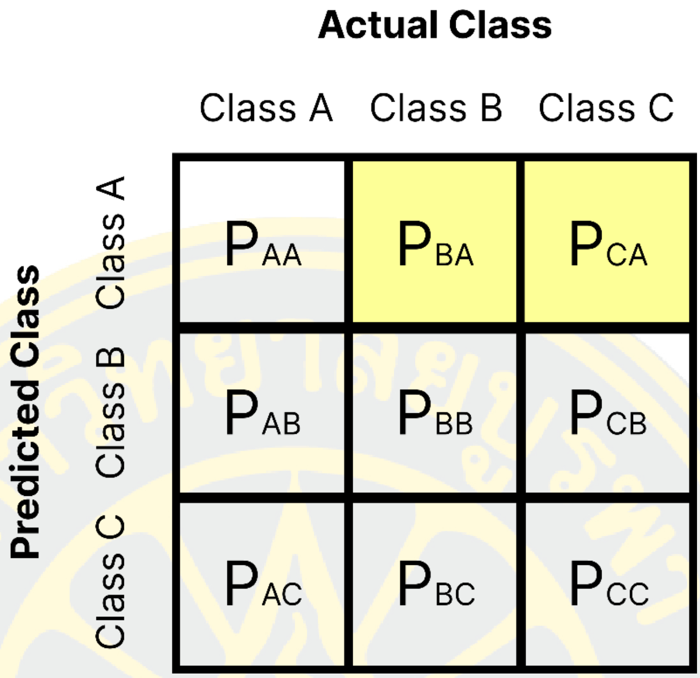

Figure 10 FP value selection

FP of class A = $P_{BA} + P_{CA}$

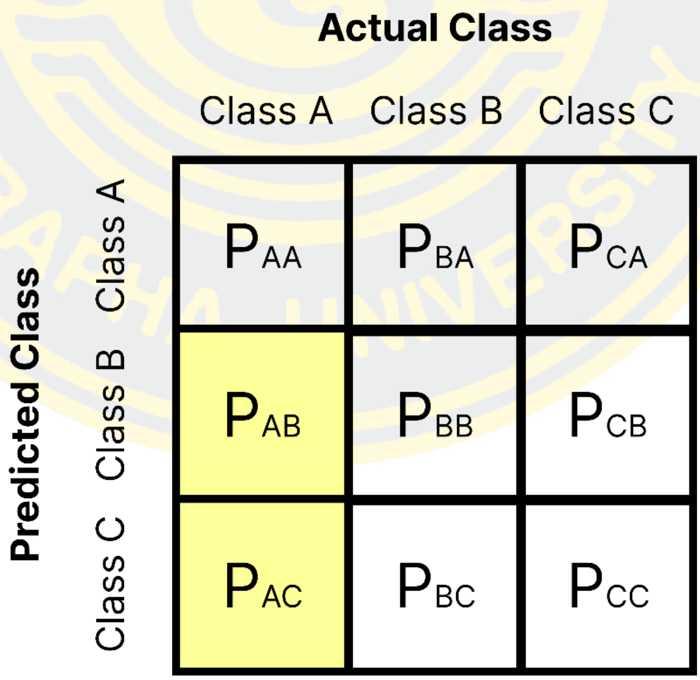

Figure 11 FN value selection

FN of class A = $P_{AB} + P_{AC}$

CHAPTER 3
METHODOLOGY AND ANALYSIS

Overview

The concept of this experiment is based on the training and testing for creating an AI model that is based on a pre-trained model. The overview of experiment method is shown in the figure 12 below.

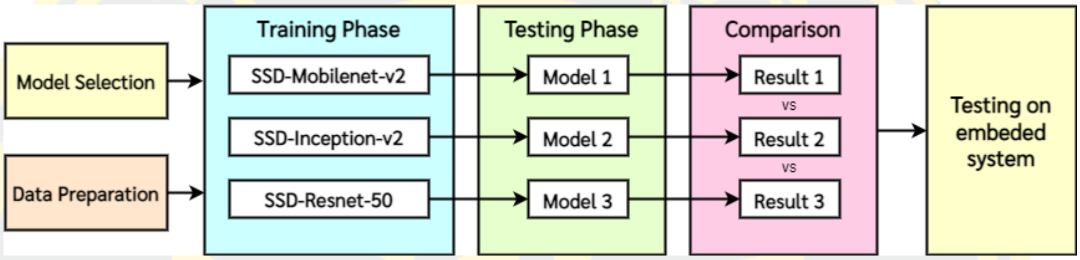

Figure 12 Overview

From Figure 12, we first considered using 3 pre-trained models base on COCO dataset, which are ssd_mobilenet_v2_coco, ssd_inception_v2_coco, and ssd_resnet_50_fpn_coco. These 3 models have a good quality and detect the object relatively fast. At this step, we also prepare our own dataset for training and testing phase, which includes 1,000 images for each waste type (6,000 images altogether). The details of dataset preparation will be explained in the next topic. Then, 80% of images of each waste type will be trained with each pre-trained model. In this phase, we control the time of training to be 48 hours for each pre-trained type. At this step, we collect the relevant parameters, including training steps, start loss, and stop loss. In terms of getting accuracy rate, we have to test the trained models with the remaining 20% of images of each waste type. Then the trained models with our own dataset can be compared in terms of accuracy rate. The one that has the highest accuracy rate is a good candidate for our proposed automatic waste sorting bin. However, there are other factors that have to be taken into account such as frame rate, CPU usage, GPU usage, and RAM usage, when running on Jetson Nano. Therefore, the last step is to test the trained models on the real system, namely the Jetson Nano environment. The mentioned parameters can be then observed and collected.

Comparison

This phase compares the findings of the Confusion Matrix from the test results of the three models with the results of the Accuracy, Precision, Recall, and F1-score calculations. These three models will allow us to determine whether our trained models can be put into practice and whether their accuracy is acceptable for usage in waste sorting applications, as well as compare which models are the most suitable.

Actual Calss

Predicted Calss	Plastic Bottles	Glass Bottles	Metal Cans	Plastic Bags	Food Containers	Plastic Coffee Cups
Plastic Bottles	P_{pp}	P_{gp}	P_{mp}	P_{bp}	P_{fp}	P_{cp}
Glass Bottles	P_{pg}	P_{gg}	P_{mg}	P_{bg}	P_{fg}	P_{cg}
Metal Cans	P_{pm}	P_{gm}	Pmm	P_{bm}	P_{fm}	P_{cm}
Plastic Bags	P_{pb}	P_{gb}	P_{mb}	P_{bb}	P_{fb}	P_{cb}
Food Containers	P_{pf}	P_{gf}	P_{mf}	P_{bf}	P_{ff}	P_{cf}
Plastic Coffee Cups	P_{pc}	P_{gc}	P_{mc}	P_{bc}	P_{fc}	P_{cc}

Figure 13 Confusion matrix of 6 class

We will create a Confusion matrix of 6 classes consisting of classes.

Determine the value:

P = Plastic Bottles

G = Glass Bottles

M = Metal Cans

B = Plastic Bags

F = Food Containers

C = Plastic Coffee Cups

After the variables are defined, we will evaluate the TP, TN, FP, FN values and use these values to calculate the Accuracy, Precision, Recall, and F_1-score for each class plastic bottles as follows:

$$TP_{\text{Plastic Bottle}} = P_{pp}$$
$$TN_{\text{Plastic Bottle}} = P_{gg}+P_{mg}+P_{bg}+P_{fg}+P_{cg}+P_{gm}+P_{mm}+P_{bm}+P_{fm}+P_{cm}+P_{gb}+P_{mb}+P_{bb}+P_{fb}+P_{cb}+P_{gf}+P_{mf}+P_{bf}+P_{ff}+P_{cf}+P_{gc}+P_{mc}+P_{bc}+P_{fc}+P_{cc}$$
$$FP_{\text{Plastic Bottle}} = P_{gp}+P_{mp}+P_{bp}+P_{fp}+P_{cp}$$
$$FN_{\text{Plastic Bottle}} = P_{pg}+P_{pm}+P_{pb}+P_{pf}+P_{pc}$$

Model selection

Selecting a model for waste separation necessitates selecting an Object detection model. We used the models SSD Mobilenet v2 coco, SSD inception v2 coco, and SSD resnet 50 fpn coco in this investigation. All three of these models are pre-trained models based on the COCO dataset, which has data from the Object classes corresponding to the trash types we want to identify, making these three pre-trained models a good fit to implement. These three models are also popular for the Nvidia Jetson Nano, which we used in our study.

Data Preparation

In terms of creating a suitable AI model, one has to refining a pre-trained model with a proper dataset. The dataset used in this work is done using the steps shown in the Figure 14 below.

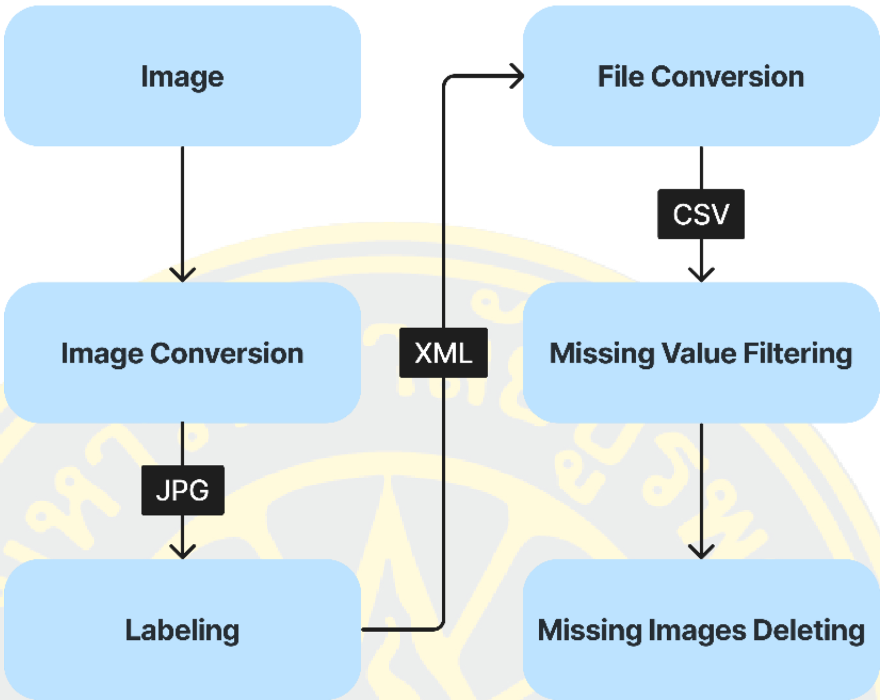

Figure 14 Data preparation

The dataset preparation steps start from acquiring proposed images from various sources such as on the Internet, real images taken from the target area, real images taken in the lab, etc. The figure 15-20 depicts 6 types of solid waste used in this work.

Figure 15 Example image of plastic bottle

Figure 16 Example image of glass bottle

Figure 17 Example image of metal cans

Figure 18 Example image of plastic bags

Figure 19 Example image of food containers

Figure 20 Example image of plastic coffee cups

Since we propose 6,000 images from 6 waste types, we have to have the raw images more than 6,000 because they will be filtered out in case they lack of some features. The 6 waste types are plastic bottles, glass bottles, metal cans, plastic bags, food containers, and plastic coffee cups. The next step is to convert the images into JPG format. After that, the images will be labeled with actual type of waste, e.g. an image of a plastic bottle will be labeled with “plastic bottle”, an image of a metal can will be labeled with “metal can”, respectively. Figure 21-24 indicates the Labeling step. Then the images will be converted to XML files and after that they will be converted to CSV file format. In this CSV file format, the images will be filtered out if they lack of required features. With this information, we then can remove that specific image from the dataset described at the beginning.

From Figure 21-24, each waste has to have 4 images,

a. The ordinary half-image
b. The ordinary full-image
c. The invert half-image
d. The invert full-image

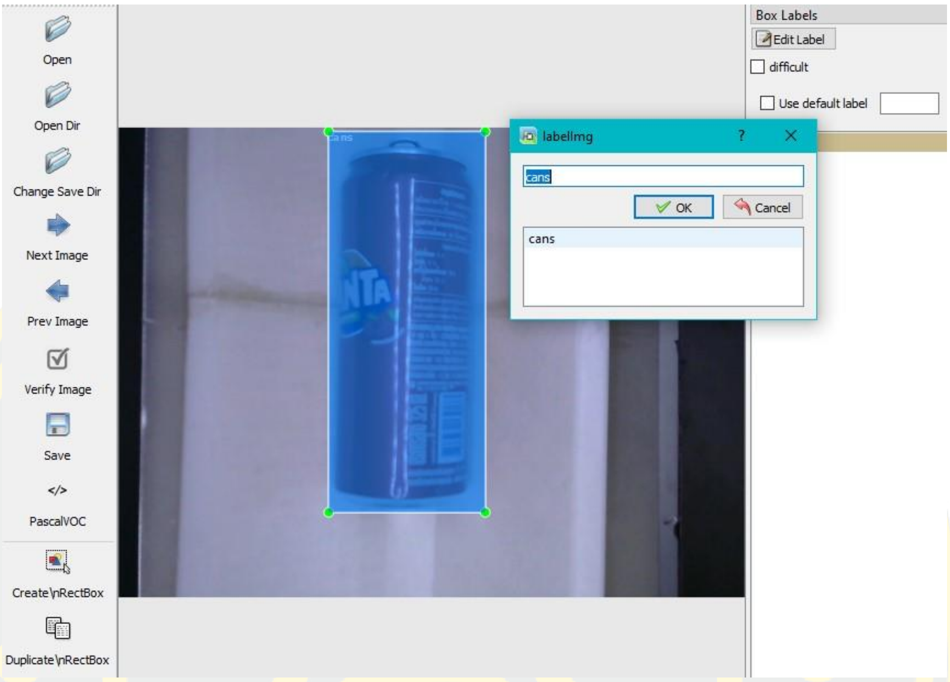

Figure 21 Labeling the ordinary full-image

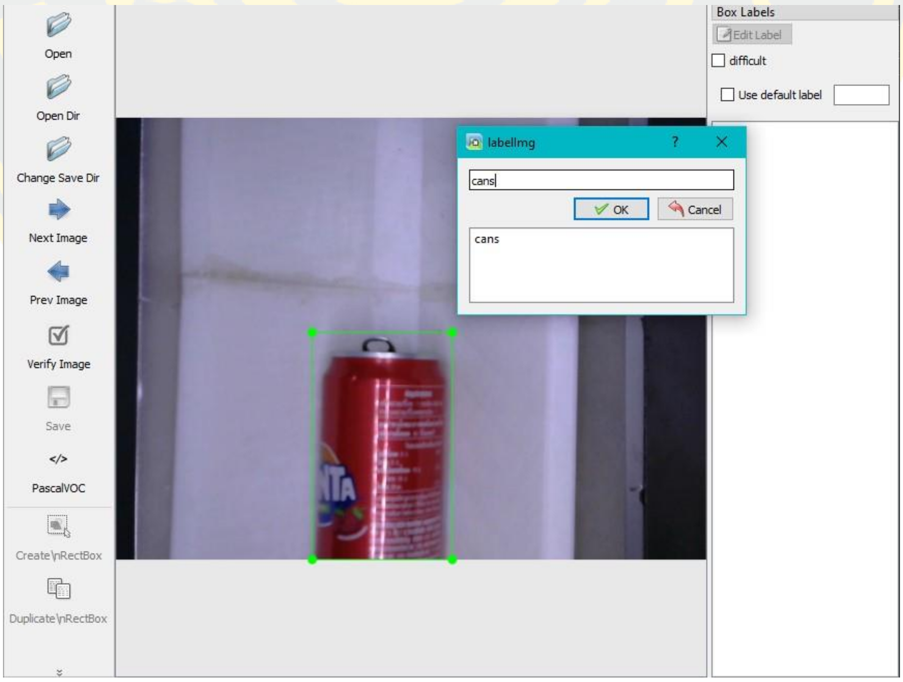

Figure 22 Labeling the ordinary half-image

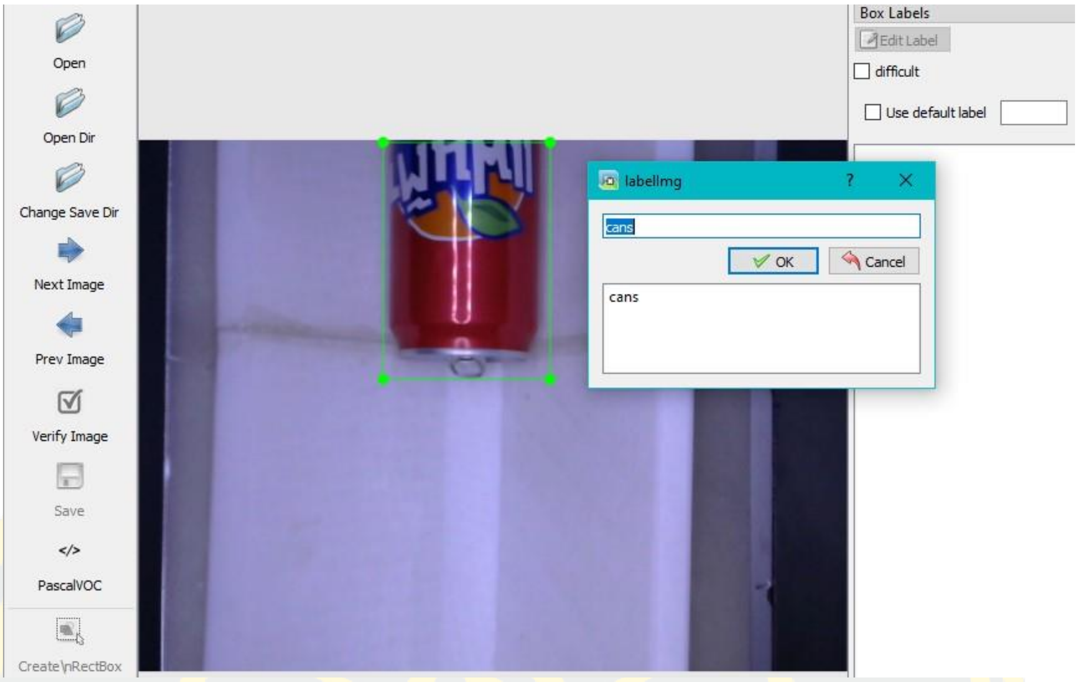

Figure 23 Labeling the invert half-image

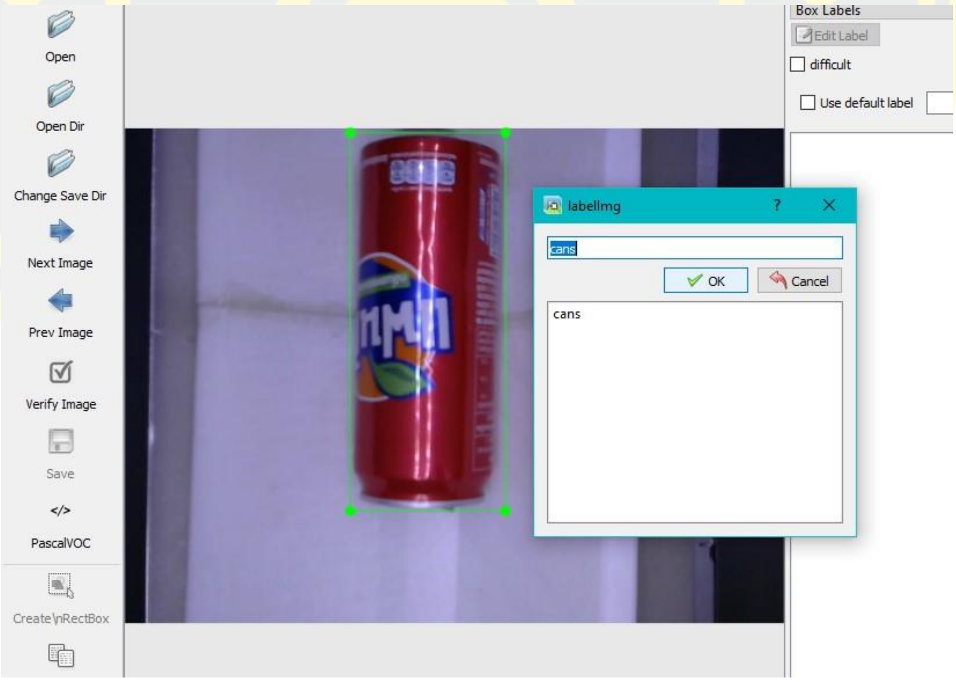

Figure 24 Labeling the invert full-image

These filtered out images form a dataset of the 6 waste types. This dataset is divided into 2 clusters, namely the training dataset in amount of 80% of total images and the testing one in amount if 20% of the total images. In our case, we have 6,000 images in total. Thus, the training dataset contains 800 images for each waste type and the testing dataset contains 200 images respectively. The next step is to insert these images into the training process with respective pre-trained model.

Training Phase

In the training phase, the dataset will be read into the single pre-trained model, one by one. This process of creating a waste classification model is based on TensorFlow training library equipped with a single pre-trained model. The training time for each pre-trained model takes 48 hours. This is our control variable for this experiment. The training hours of 48 come from our prior trial and error experiment. Within these 48 hours the models show significant accuracy rate and can be used for performance comparison test. In this work, we used a computer equipped with the CPU Intel Core i7-10700K, 32 GB RAM and the GPU is GEFORCE RTX 2070 Super 8 GB GDDR5. Furthermore, in this work we use the TensorBoard as a Training Measurement tool. This tool allows us to observe and collect several important training results. In this experiment, we are interested in the number of steps, the loss value of start loss, and the loss value of stop loss, after the training run for 48 hours. The loss value indicates how good the trained model will be, the fewer the better. Figure 25 shows a TensorBoard output of a waste classification model trained with the pre-trained ssd_mobilenet_v2_coco at 200,000 steps and loss value of stop loss is 0.84.

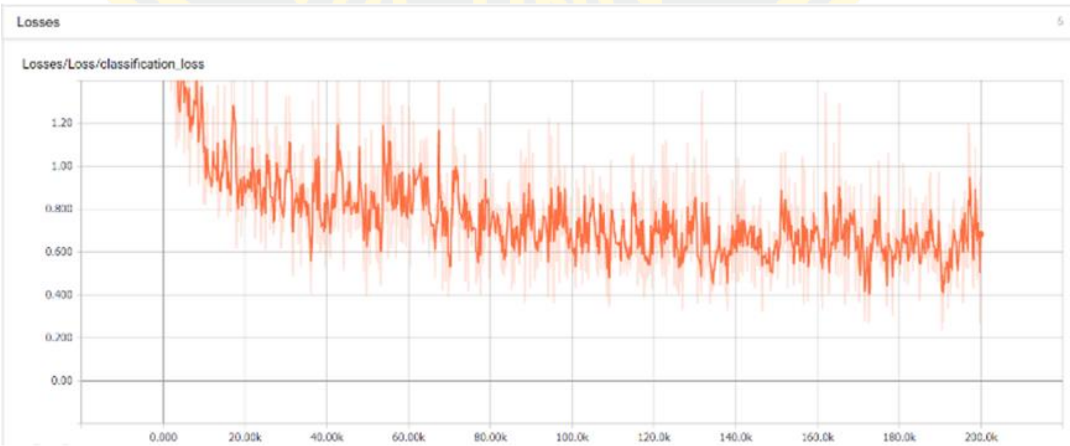

Figure 25 Loss value monitor

Testing Phase

After we've trained the model using the data we've collected, we'll put it to the test by using a Confusion Matrix to assess the model's performance based on the outcomes it predicts. Compare the model's results to the real results from the sorts of photographs we used for the test (20% of the total images gathered). We calculate the values received from the Matrix, such as Accuracy, Precision, Recall, and F1-score, and utilize these numbers to compare the performance of the three models after they are generated.

Testing on Embedded System

After testing the performance of all three models and calculating their accuracy, there was one more thing to consider when selecting the best model for the Nvidia Jetson Nano. which is the embedded system we've chosen for this project? The most essential factor is our live testing with the Nvidia Jetson Nano, in which we compare which model is best suited to an embedded system based on AVG FPS, CPU Usage, GPU Usage, Memory Usage, and CPU TMP while each model is running.

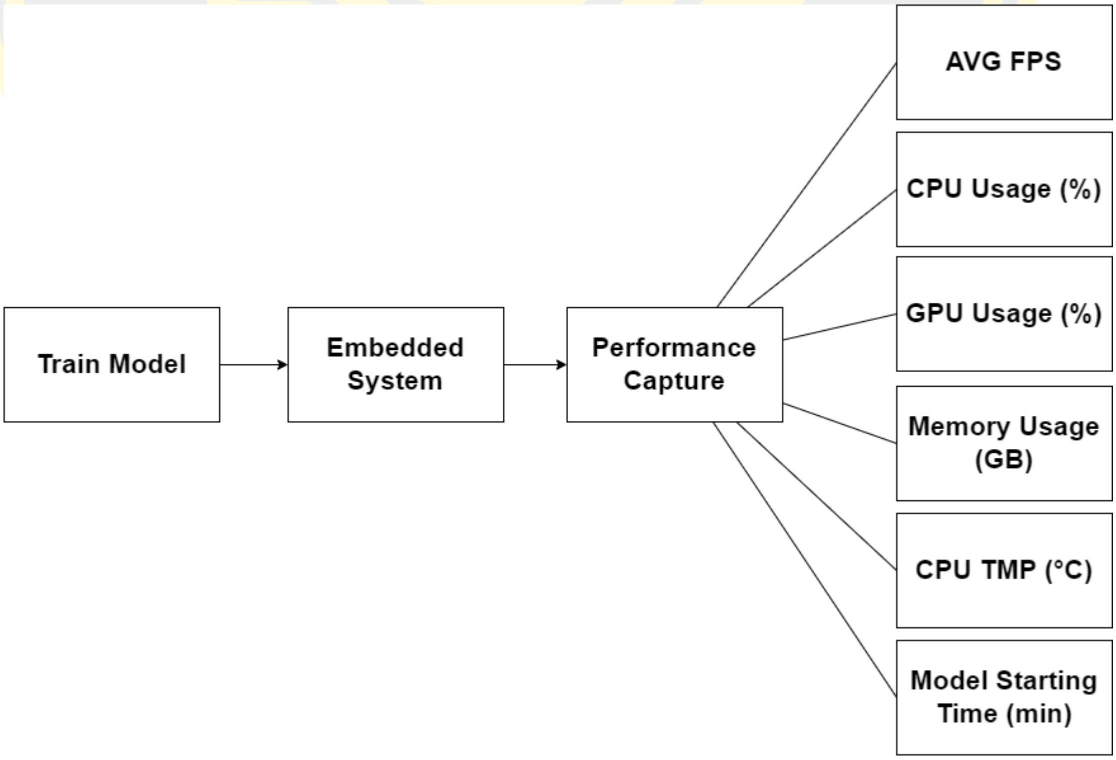

Figure 26 Embedded system performance capture

After we go through the process of training the model that is ready for use on the embedded system, from the figure 26, we will have a step to verify the performance of the model in conjunction with the embedded system. The factors we are interested in are as follows:

a. AVG FPS is the average frame rate from the video. That the model can detect in one second. This value, if there are large numbers, will allow the model to effectively work in the form of motion capture or image inspection in real-time.
b. CPU Usage (%) is the CPU performance utilization value. It is a factor indicating how much the CPU usage model in the waste classification uses the CPU performance of the embedded system. However, excessive CPU usage may result in slower execution of various parts of the embedded system.
c. GPU Usage (%) is the GPU's performance utilization value. It is a factor in how much the embedded system's GPU usage model is used in the waste classification model. However, if the GPU is used, which is the main factor in the performance of the model because The GPU is the primary image processor used by the model, if overused it may affect image processing efficiency. reduced and may cause heat to the image processor.
d. Memory Usage (%) is the value of memory usage performance. It is the main memory used by the embedded system and in addition, other systems running on the embedded system also share memory. As well, if too much memory is used, it may cause the system to work in other parts of the embedded system to stop working.
e. CPU Temp is a value that indicates the temperature. of the heat generated by the computation in the embedded system. Excessive temperature may result in the embedded system stopping working or the device. Has a shorter service life
f. Model Starting Time is a value indicating the start-up time of the model starting from the initialization of the model's required resources until the initial image detection of the model.

Real World Testing

After the performance of all three models was measured and the optimal model results were obtained. Therefore, we tested the model in a real-world scenario to illustrate the model's effectiveness in prototype innovation in addition to lab testing, to assess the model's preparedness for use in real-world contexts.

The concept of this AI model accuracy evaluation in field test environment is to let the users test the system naturally. The users in this case are students and university staffs at Burapha University. The appropriate places are selected for placing smart bins equipped with AI model running on Jetson Nano [9] embedded system for automatically separating recycle wastes. These places are easy to access and lively where many people would walk around or have activities nearby in terms to get those test populations to litter in an automated waste sorting model we developed. The Figure 27 depicts the concept overview for this research.

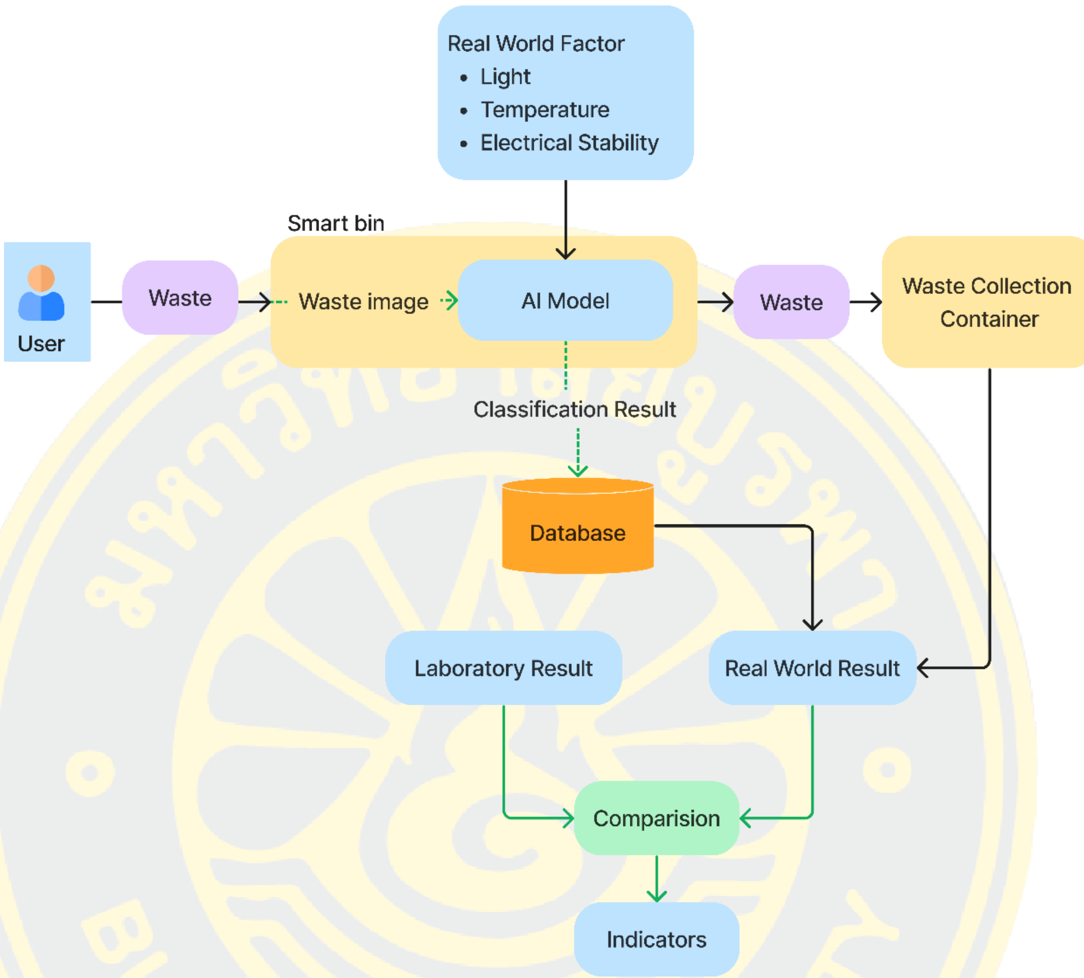

Figure 27 Concept overview

From Figure 27, the measurement of the smart bin's efficacy in terms of accuracy based on the sample population's actual usage of the bin can be accomplished. A user litters a waste into one of the smart bins, the waste image will be made and sent to be processed in the AI model. When the waste sorting AI model examines and determines the type of waste that has been littered, the database will be updated with the names of the littered waste types. Each type of the waste will be stored in its respective waste sorting container. Then, each day, we collect and count the amount of the waste that is discarded into the waste sorting bin. We then compare it to the list of the waste name recorded in the database, as predicted by the model.

After comparison, we can identify the wrong recognition of the AI model. This information can be analyzed as an accuracy rate of the AI model in the real-world environment. Furthermore, we can compare this real-world accuracy rate and the laboratory accuracy rate. This figure will suggest how intensive the factors affects the accuracy of the originate AI model.

Figure 28 Data prepare labeling XML output

	A	B	C	D	E	F	G	H
1	**filename**	**width**	**height**	**class**	**xmin**	**ymin**	**xmax**	**ymax**
2	BirdyCan (10).JPG	741	749	can	180	55	492	660
3	BirdyCan (11).JPG	840	871	can	230	31	643	841
4	BirdyCan (12).JPG	525	586	can	138	43	383	531
5	BirdyCan (13).JPG	823	747	can	272	172	473	591
6	BirdyCan (14).JPG	845	822	can	298	136	582	741
7	BirdyCan (15).JPG	695	579	can	197	77	390	462
8	BirdyCan (16).JPG	871	761	can	273	52	578	666
9	BirdyCan (17).jpg	200	300	can	34	30	165	285
10	BirdyCan (18).jpg	800	800	can	303	202	497	598
11	BirdyCan (19).jpg	350	350	can	95	22	249	333
12	BirdyCan (20).jpg	458	458	can	132	37	327	424
13	BirdyCan (21).jpg	800	800	can	299	204	497	596
14	BirdyCan (22).jpg	500	500	can	166	70	337	434
15	BirdyCan (23).JPG	797	795	can	233	61	559	735
16	BirdyCan (24).JPG	826	648	can	286	157	481	543
17	BirdyCan (25).JPG	886	641	can	290	94	485	484
18	BirdyCan (26).jpg	600	600	can	169	33	431	566
19	BirdyCan (27).jpg	200	200	can	74	45	125	150
20	BirdyCan (28).jpg	720	400	can	289	54	434	341
21	BirdyCan (29).jpg	800	800	can	221	51	576	753

Figure 29 Data prepare labeling CSV output combined

In the figure 29, when we assign labels to each image of the garbage type, we convert the XML file we obtained from the labeling into a CSV file, which is a format that can be used to train the model.

CHAPTER 4
RESULT

After running the training phase using the 3 pre-trained models, the following results can be shown.

The Training Phase

In the training phase, some performance indicators of model trained with the 3 pre-trained models are collected. Table 2 shows the training results.

Table 2 Training results

Factor	Models		
	SSD-Mobilenet-V2	**SSD-Inception-v2**	**SSD-Resnet-50**
Training Time	48 hrs.	48 hrs.	48 hrs.
Number of Steps	200,000	95,079	56,537
Number at Start Loss	20.32	32.45	2.65
Number at Stop Loss	0.84	0.536	0.053
Loss Difference (%)	-95.8661	-98.3482	-98.0000

From Table 2, the training time is the control variable and the training lasts 48 hours for each model. This indicates that at the same training time, the ssd_inception_v2_coco shows the best result. The percentage of its loss difference is 98.3482%, which means that the loss value of the model declines the fastest. The model accuracy will be then the highest in this case.

The Testing Phase

In the training phase, some performance indicators of model trained with the 3 pre-trained models are collected. Table 2 shows the training results.

1. The result for the ssd_mobilenet_v2_coco model: Table 3 shows the result of the calculated confusion matrix for 6-class classification. In this testing phase, 200 images of 6 waste types were used.

Table 3 Confusion Matrix of 6-class Classification for SSD-Mobilenet-V2

	Plastic Bottle	**Glass Bottle**	**Metal Can**	**Plastic Bag**	**Food Container**	**Coffee Cup**
Plastic Bottle	200	1	0	0	0	0
Glass Bottle	1	198	6	0	0	0
Metal Can	0	0	188	0	0	0
Plastic Bag	0	0	0	200	0	0
Food Container	0	0	1	0	200	0
Coffee Cup	0	0	5	0	0	200

Table 4 Performance Measure Calculation

	Plastic Bottle	**Glass Bottle**	**Metal Can**	**Plastic Bag**	**Food Container**	**Coffee Cup**
TP	200	198	188	200	200	200
TN	998	994	1000	1000	999	995
FP	1	7	0	0	1	5
FN	1	1	12	0	0	0
Precision	0.99502	0.96585	1	1	0.99502	0.97560
Recall	0.99502	0.99497	0.94	1	1	1
F1-score	0.99502	0.98019	0.96907	1	0.99750	0.98765

The total accuracy of these 6 classes together can be then calculated. The total accuracy for the ssd_mobilenet_v2_coco model is 0.988333333 or 98.83%.

2. The result for the ssd_inception_v2_coco model: Table 5 shows the result of the calculated confusion matrix for 6-class classification. As mentioned before, this testing phase also uses 200 images of 6 waste types.

Table 5 Confusion Matrix of 6-class Classification for SSD-Inception-V2

	Plastic Bottle	**Glass Bottle**	**Metal Can**	**Plastic Bag**	**Food Container**	**Coffee Cup**
Plastic Bottle	200	0	0	0	0	0
Glass Bottle	1	199	0	0	0	0
Metal Can	0	0	199	0	0	0
Plastic Bag	0	0	0	200	0	0
Food Container	0	0	1	0	200	2
Coffee Cup	0	0	0	0	0	198

Table 6 Performance Measure Calculation

	Plastic Bottle	**Glass Bottle**	**Metal Can**	**Plastic Bag**	**Food Container**	**Coffee Cup**
TP	200	199	199	200	200	198
TN	999	1000	1000	1000	997	1000
FP	0	1	0	0	3	0
FN	1	0	1	0	0	2
Precision	1	0.995	1	1	0.98522	1
Recall	0.99502	1	0.995	1	1	0.99
F1-score	0.99750	0.99749	0.99749	1	0.99255	0.99497

The total accuracy of these 6 classes together can be then calculated. The total accuracy for the ssd_inception_v2_coco model is 0.996666667 or 99.67%.

3. The result for the ssd_resnet_50_fpn_coco model: Table 7 shows the result of the calculated confusion matrix for 6-class classification. Also in this testing phase, 200 images of 6 waste types were used.

Table 7 Confusion Matrix of 6-class Classification for SSD-Resnet-50

	Plastic Bottle	**Glass Bottle**	**Metal Can**	**Plastic Bag**	**Food Container**	**Coffee Cup**
Plastic Bottle	200	0	0	0	0	0
Glass Bottle	1	199	0	0	0	0
Metal Can	0	0	200	0	0	1
Plastic Bag	0	0	0	200	0	0
Food Container	0	0	0	0	199	0
Coffee Cup	0	0	0	0	1	199

Table 8 Performance Measure Calculation

	Plastic Bottle	Glass Bottle	Metal Can	Plastic Bag	Food Container	Coffee Cup
TP	200	199	200	200	199	199
TN	999	1000	999	1000	1000	999
FP	0	1	1	0	0	1
FN	1	0	0	0	1	1
Precision	1	0.995	0.99502	1	1	0.995
Recall	0.99502	1	1	1	0.995	0.995
F1-score	0.99750	0.99749	0.99750	1	0.99749	0.995

The total accuracy of these 6 classes together can be then calculated. The total accuracy for the ssd_resnet_50_fpn _coco model is 0.9975 or 99.75%.

From the accuracy point of view, the SSD-Resnet-50 model delivers the best result for image classification and object detection. However, these results were produced on a computer that is considered as a high-performance system. It is also still interesting to run these trained models on our Jetson Nano.

The Results on Jetson Nano

Running these trained models on a Jetson Nano board reveals the insight of the real use of models. A Jetson Nano board is designed to serve several Machine Learning tasks such as image classification, object detection, segmentation, etc. Although its performance seems to be higher comparing to other boards such as Raspberry Pi or Banana Pi. The results in the Table 9 show that the model with the highest accuracy may not be suitable for an embedded system.

Table 9 Results of 3 Models Running of Jetson Nano

Factors	SSD-Mobilenet-v2	SSD-Inception-v2	SSD-Resnet-50
AVG FPS	7.71	4.35	4.21
CPU Usage (%)	64	61	68
GPU Usage (%)	71	86	91
Memory Usage (GB)	3.2	3.5	3.7
CPU TMP (°C)	54	67	69
Model Starting Time (min)	3.24	16.06	21.02

From Table 9, the most suitable model for running image classification and object detection on an embedded system is the ssd_mobilenet_v2_coco one. It has the highest number of the average frame rate. Its GPU usage and the Memory usage are the lowest at 71% and 3.2 GB respectively. Furthermore, the model needs starting time only 3.24 minutes where the others need more than 16 minutes to start.

The Results of Field Test

The results of the field test were collected over the time of 73 days. Each location revealed the results as follows:

1. Faculty of Informatics, Burapha University

Table 10 Location 1 Environment Factor

Faculty of Informatics, Burapha University	
Types	**% Accuracy**
Plastic Bottles	68.22
Glass Bottles	58.64
Cans	79.22
Plastic Bags/Food Containers	76.2
Plastic Cups	72.54
AVG % Accuracy	70.96

In real-world testing at location 1 (Faculty of Informatics), Plastic Bags and Food Containers have an accuracy of 76.2%, Cans have an accuracy of 79.22%, Plastic Bottles have an accuracy of 68.22%, Plastic Cups have an accuracy of 72.54% and Glass Bottles have an accuracy of 58.64%. It was determined that the model's average accuracy was approximately 70.96%.

2. Student Affairs, Burapha University

Table 11 Location 2 Environment Factor

Student Affairs, Burapha University	
Types	**% Accuracy**
Plastic Bottles	84.32
Glass Bottles	67.15
Cans	87.42
Plastic Bags/Food Containers	82.31
Plastic Cups	80.25
AVG % Accuracy	80.29

In real-world testing at location 2 (Student Affairs), Plastic Bags and Food Containers have an accuracy of 82.31%, Cans have an accuracy of 87.42%, Plastic Bottles have an accuracy of 84.32%, Plastic Cups have an accuracy of 80.25% and Glass Bottles have an accuracy of 67.15%. It was determined that the model's average accuracy was approximately 80.29%.

3. Division of Educational Service, Burapha University

Table 12 Location 3 Environment Factor

Division of Educational Service, Burapha University	
Types	**% Accuracy**
Plastic Bottles	91.42
Glass Bottles	82.02
Cans	96.25

Division of Educational Service, Burapha University	
Types	**% Accuracy**
Plastic Bags/Food Containers	95.32
Plastic Cups	89.24
AVG % Accuracy	90.85

In real-world testing at location 3 (Division of Educational Service), Plastic Bags and Food Containers have an accuracy of 95.32%, Cans have accuracy of 96.25%, Plastic Bottles have an accuracy of 91.42%, Plastic Cups have an accuracy of 89.24% and Glass Bottles have an accuracy of 82.02%. It was determined that the model's average accuracy was approximately 90.85%.

CHAPTER 5
DISCUSSION AND CONCLUSIONS

Discussion

From studying the COCO dataset and applying it to this research. We have created our own waste dataset. There were six types of waste data collected. From the shape of the bottle to the label format, we found that there were still limitations from the shape of the bottle to the label format. Products are constantly changing if more accurate models are needed. Information may need to be kept new and up-to-date.

To prepare the model data prior to training, this is a long process as we have to collect images of the six types of waste to cover and then we need to define them. Label for all data as well. The method of creating a Label will take a long time. After preparing all the data sets, we will need to share the proportions of the images for use in training and testing the model as well. We will divide the proportions of the images according to the type of waste, by brand and by size, which will be used in Practice and test in equal ratios so that the results of the test do not deviate from any type of waste. In the case where we collect datasets to train the model if there is a different number of images of each type of garbage and imbalance data, we can solve this problem by using oversampling and undersampling methods.

Key Findings

From the field test, it was found that the factors affecting the sensing performance of the model were Light Intensity, Temperature, Electrical Stability, Population, Area. As a result, we discovered that if these factors were controlled according to the model's requirements, the efficiency of the model's waste separation could be improved. For example, in creating a smart bin, the part used to receive waste into the processing system should be sufficiently regulated to allow the model to detect the type of waste received. If the lighting conditions are available, the model will be more efficient in computation.

Innovative application of ready-to-use models for waste segregation, for example smart bins, will provide society with innovations that can help address the problem of waste segregation. By relying on technology that can be easily applied, such as Embedded system from our experiments in this research, we know that this innovation, if further developed that the model is compatible with embedded systems, can create innovations in solving the separation problem. waste such as smartbin can actually be built and used in a real-world environment if it is designed to support both the model and the embedded system.

Conclusion

In this work, we designed and implemented an experiment for finding the most suitable AI model for an embedded system. The AI model should be pre-trained using the coco dataset. These 3 pre-trained models are ssd_mobilenet_v2_coco, ssd_inception_v2_coco, and ssd_resnet_50_fpn_coco. The most important conditions include the computational power consumption, the average frame rate, and the model starting time. From our experiment, we found out that the 3 model trained with pre-trained models mentioned above deliver almost similar accuracy rate. The ssd_mobilenet_v2_coco has the accuracy of 98.83%, the ssd_inception_v2_coco has the accuracy of 99.67%, and the ssd_resnet_50_fpn_coco has the accuracy of 99.75%. However, the ssd_resnet_50_fpn_coco is the one that consumes the most computational power and needs the longest time to start. In contrast, the model with ssd_mobilement_v2_coco delivers enough accuracy and it consumes the fewest computational power. Furthermore, it needs the shortest time to start. From these results we consider that the AI model trained based on ssd_mobilenet_v2_coco model is the most suitable model for an embedded system.

After we tested it in a real environment, the results obtained from the cat model showed that the Division of Educational Service, Burapha University, was the place where the models were most accurate at 90.85%, but compared to the results of the research lab, the accuracy was reduced by 7.98%. Finally, when the model was deployed in all three real-world locations over a period of two months, the total accuracy was 80.70%, while the accuracy was reduced by 18.13% when compared to the laboratory.

Future Work

In the future, the trained model will be used in an automatic waste sorting system. We will collect the data for analyzing its usage in the field test and long run. Furthermore, we want to improve the quality of the model by optimizing the training dataset. There should be more varieties of images such as images with different scales and sizes, images with dirt on their surface, etc. This will improve the recognition rate due to the real dirty waste that is being thrown into the system. And we will revise the factors that affect the automatic waste sorting AI model when deployed in a real-world environment and adding methods to control environmental problems such as light interfering the waste chamber, which causes the model prediction error. We will investigate how different light intensities affect the level of AI model accuracy by using standardized light intensity measurement instruments.

Furthermore, the smart bin will be redesigned to fit in different environment and conditions. We have to concern how to design the inlet chamber for inserting the recycle waste according to users’ behaviors. A new concept for managing unrecognized waste should be considered. This could help reducing the load in the recognition process of the model.

REFERENCES

Adedeji, O., & Wang, Z. (2019). Intelligent waste classification system using deep learning convolutional neural network. *Procedia Manufacturing, 35*, 607-612.

Ali, H., Khursheed, M., Fatima, S. K., Shuja, S. M., & Noor, S. (2019). *Object recognition for dental instruments using SSD-MobileNet.* Paper presented at the 2019 International Conference on Information Science and Communication Technology (ICISCT).

Aral, R. A., Keskin, Ş. R., Kaya, M., & Hacıömeroğlu, M. (2018, 10-13 Dec. 2018). *Classification of TrashNet Dataset Based on Deep Learning Models.* Paper presented at the 2018 IEEE International Conference on Big Data (Big Data).

Bircanoğlu, C., Atay, M., Beşer, F., Ö, G., & Kızrak, M. A. (2018, 3-5 July 2018). *RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks.* Paper presented at the 2018 Innovations in Intelligent Systems and Applications (INISTA).

Chunxiang, Z., Jiacheng, Q., & Wang, B. (2022). YOLOX on Embedded Device With CCTV & TensorRT for Intelligent Multicategories Garbage Identification and Classification. *IEEE Sensors Journal, 22*(16), 16522-16532. doi:10.1109/JSEN.2022.3181794

Corporation, N. (2022). Jetson Nano Developer Kit. Retrieved from https://developer.nvidia.com/embedded/jetson-nano-developer-kit

Dawod, R. G., & Dobre, C. (2022). ResNet interpretation methods applied to the classification of foliar diseases in sunflower. *Journal of Agriculture and Food Research, 9*, 100323. doi:https://doi.org/10.1016/j.jafr.2022.100323

Ghoury, S., Sungur, C., & Durdu, A. (2019). *Real-time diseases detection of grape and grape leaves using faster r-cnn and ssd mobilenet architectures.* Paper presented at the International conference on advanced technologies, computer engineering and science (ICATCES 2019).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . Zitnick, C. L. (2014). *Microsoft coco: Common objects in context.* Paper presented at the European conference on computer vision.

Melinte, D. O., Travediu, A.-M., & Dumitriu, D. N. (2020). Deep convolutional neural networks object detector for real-time waste identification. *Applied Sciences, 10*(20), 7301.

Patel, D., Patel, F., Patel, S., Patel, N., Shah, D., & Patel, V. (2021). *Garbage Detection using Advanced Object Detection Techniques.* Paper presented at the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS).

APPENDICES

Example data of waste image 6 type

Figure 30 Example glass bottle image (1)

Figure 31 Example glass bottle image (2)

Figure 32 Example metal cans image (1)

Figure 33 Example metal cans image (2)

Figure 34 Example plastic coffee cups (1)

Figure 35 Example plastic coffee cups (2)

Figure 36 Example food containers

Link Document

1. Confusion matrix
   https://docs.google.com/spreadsheets/d/1hqLF7kmQZf3mWO79wqQxoahQj-1pulo/edit?usp=share_link&ouid=101471441107704523469&rtpof=true&sd=true
2. Prediction Graph
   https://docs.google.com/spreadsheets/d/1aZkjQYx7ZXdgYJpxlKyBrQDufNu61D2K/edit?usp=share_link&ouid=101471441107704523469&rtpof=true&sd=true

BIOGRAPHY

NAME	Sorawit Thokrairak
DATE OF BIRTH	11 Sep 1997
PLACE OF BIRTH	Saohai, Saraburi
PRESENT ADDRESS	12 Moo.3, Saohai Sub-district, Saohai District, Saraburi, 18160
EDUCATION	Bachelor's degree: Informatics, Information Technology, Burapha University Master's degree: Data Science, Information Technology, Burapha University
AWARDS OR GRANTS	The Twenty-Second National Software Contest (NSC 2020) October-2019 - February-2020 Competition on special topics in the type of IoT in the project “Trash2Cash” The Computing Technology Industry Association (CompTIA) May-2019 - May-2019 Successfully completed the requirements to be recognized as CompTIA Cloud Essentials The 7th ASEAN Undergraduate Conference in Computing March-2019 - March-2019 Presented a research paper "Rewarding Smart Recycle Bin Prototype Designed for Saensuk Sub-District" with the award of Excellent Paper The Twenty-First National Software Contest (NSC 2019) March-2019 - March-2019 Honorable mention on special topics in the type of IoT in the project "Rewarding Smart Recycle Bin" Eastern Code Festival July-2018 - July-2018 The third prize of the Code Marathon competition in the IOS group