MSc Dissertation

Final Report

MSc in Integrated Machine Learning System

Student: Youpeng Yu

Student number: 19132770

Project Title:

Towards an AI recycling bin: Improving image processing for waste classification

Supervisor: Dr Ryan Grammenos

University College London

Dept. of Electronic and Electrical Engineering

2020/2021

Abstract

Nowadays, the increasing amount of global waste has overwhelmed the waste industry. Various types of technologies have been adopted to help the waste industry collect, transport, and dispose of waste. In 2019, the IBM Wastenet team designed an AI recycling bin using the image classification technology to classify waste into recyclables and non-recyclables. This project works on improving the IBM Wastenet project from the image processing aspect to improve the classification accuracy, and further classify recyclables into five recycling waste types (including cardboard, glass, metal, paper, and plastic). However, little research has been done to develop a systematic approach to train waste image classification models. This project aims to develop a systematic approach to split datasets, tune training parameters, and choose data augmentation techniques.

Transfer learning techniques are used extensively in waste image classification projects to achieve higher classification accuracy, where the model structure and pre-trained weights are used to transfer basic image knowledge, such as colours, lines, and shapes. Moreover, statistical analysis of mean accuracy is performed to choose between different data split ratios, sampling strategies, and cross-validation techniques. Next, various training parameters, including learning rate schedulers, patience epochs, layers freezing, loss function, final classifiers, batch sizes, and learning rates, are tested and compared using the 10-fold cross validation average accuracy. Also, different data augmentation techniques, including flipping, rotation, shearing, zooming, and brightness control, are tested and compared using the 10-fold cross validation average accuracy.

The benchmark model developed by splitting the dataset and tuning training parameters achieves 91.21% test accuracy. In comparison, the benchmark model with a set of data augmentation techniques applied, the final model, achieves 95.40% test accuracy. Lastly, this model can perform real-time classification using the webcam of a computer and achieve great classification results.

Acknowledgements

The author is extremely grateful to Dr Ryan Grammenos, who patiently answered all questions and provided support and supervision from the start of this project until the end. The author is also grateful to Mindy Yang and Gary Thung, who collected and provided the TrashNet dataset. Next, the author would like to acknowledge the assistance provided by Dave Locke, Kaloyan Nikolaev Yordanov, and the IBM Wastnet team. Lastly, the author would like to appreciate all the work, waste image classification research, and their authors: Mindy Yang, White Gary, Ruiz Victoria, Bircanoglu Cenk, Özkaya Umut, and Vo Anh H.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Notations

IBM – International Business Machines

MSW – Municipal Solid Waste

AI – Artificial Intelligence

CNN – Convolutional Neural Network

ILSVRC – The ImageNet Large Scale Visual Recognition Challenge

MSc – Master of Science

GDP – Gross Domestic Product

ICT – Information and Communication Technologies

GPS – Global Positioning System

GIS – Geographic Information System

RFID – Radio Frequency Identification

UK – United Kingdom

ANN – Artificial Neural Network

RF – Random Forest

SVM – Support Vector Machine

KNN – K-Nearest Neighbour

FCNN – Fully-Connected Neural Network

RGB – Red, Green, and Blue

CV – Cross Validation

RS – Random Sampling

ANOVA – Analysis of Variance

CE – Cross Entropy

SGD – Stochastic Gradient Descent

MBGD – Mini-Batch Gradient Descent

BGD – Batch Gradient Descent

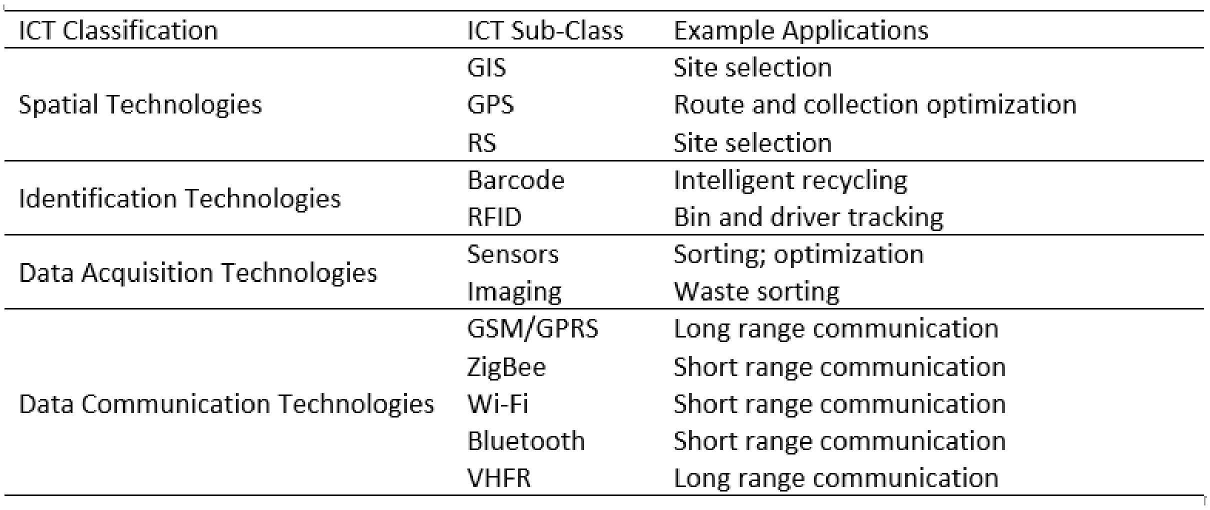
ADAM – Adaptive Moment Estimation

1. Introduction

This Master of Science (MSc) project works on improving the International Business Machines (IBM) Wastenet project from the image processing aspect to improve the classification accuracy, and further classify recyclables into five recycling waste types (including cardboard, glass, metal, paper, and plastic). To be specific, the IBM Wastenet project was developed in 2019 by the IBM Wastenet team and focused on designing an Artificial Intelligence (AI) recycling bin using the image classification technology. Programming code of this MSc project is available on the GitHub project repository [1]. Next, Chapter 1, the introduction, is split into two sections, section 1.1 gives the background information about waste management and the current classification approach, and section 1.2 shows how this MSc research paper is structured.

1.1 Background

Waste management has become a global challenge due to the increasing amount of municipal solid waste (MSW). MSW consists of everyday items consumed by households such as clothing, food, and electronics [1]. The world bank group estimated that the world waste generation would experience a 70 percent increase between 2016 and 2050, from 2.01 billion tonnes to 3.40 billion [2]. There are two main reasons behind the dramatic increase in world waste generation: GDP growth and population growth [2]. More specifically, the amount of total waste generated in a country rises when this country has more citizens with higher purchasing power. As a result, information and communication technologies (ICTs) are adopted in many regions to help the waste industry manage the increasing amount of waste.



ICT Classification	ICT Sub-Class	Example Applications
Spatial Technologies	GIS	Site selection
	GPS	Route and collection optimization
	RS	Site selection
Identification Technologies	Barcode	Intelligent recycling
	RFID	Bin and driver tracking
Data Acquisition Technologies	Sensors	Sorting; optimization
	Imaging	Waste sorting
Data Communication Technologies	GSM/GPRS	Long range communication
	ZigBee	Short range communication
	Wi-Fi	Short range communication
	Bluetooth	Short range communication
	VHFR	Long range communication

Table 1-1. Various ICTs and their example applications in waste management [3]

Table 1-1 gives information about various ICTs that often used in waste management systems. These ICTs help waste industries improve efficiencies in collecting, transporting, and sorting waste. Data communication technologies are used as complementary technologies in the other three ICTs to enable remote communication. Spatial technologies are developed to collect, store, analyse, and integrate spatial data [3]. For instance, the use of geographical

[1] https://github.com/alleno-yu/MSc_Final_Project

positioning systems(GPS) in combination with geographical information systems(GIS) can help locate and track collection vehicles and trash bins [3].

Next, identification technologies are used to track and identify waste related objects such as trash bins, garbage, and waste transport vehicles. For example, radio-frequency identification(RFID) technology is extensively used to monitor collection time, track waste bins, and record driver activities [3]. Lastly, data acquisition technologies can accelerate the data collection process by using sensors and cameras. This MSc project works on improving the image processing aspect of the IBM Wastenet project, employing data acquisition technologies to help people identify the recyclability of waste.

In 2019, the IBM Wastenet team worked with Marwell zoo and developed both a mobile application and a bin assistant - Raspberry Pi attached to the bin. To be specific, images of waste objects taken by mobile phone cameras and Raspberry Pi cameras are sent to the IBM Watson platform to perform image recognition tasks. After processing, recyclability results of waste objects are returned to either the mobile application or Raspberry Pi and displayed [4]. The Wastenet classification model is trained over hundreds of waste product images [5], making binary predictions between recyclable and non-recyclable. Unfortunately, the IBM Wastenet project's training data and model information was not available until the end of this MSc project.

The current approach to the image-based waste classification projects is implementing transfer learning techniques on different well-known Convolutional Neural Network (CNN) architectures. To be specific, these well-known CNN architectures obtained great classification results in ImageNet Challenge (ILSVRC), where ILSVRC is a large-scale object recognition competition. ILSVRC models are trained on the ImageNet dataset, which contains over 14 million images across 21 thousand classes [6]. CNN architectures such as GoogleNet, VGG, and AlexNet are excellent choices of pre-trained models since they have successfully captured general features of various objects. Next, these CNN architectures with pre-trained weights are retrained on the waste image datasets. The most commonly used waste image dataset is TrashNet, which was published in 2017 and consists of 2527 waste images across six classes [7].

In 2017, the authors in [8] developed an eleven layer network that is similar to AlexNet with different hyperparameters. This CNN architecture is then trained using the TrashNet dataset and achieved 22% test accuracy. In 2018, the authors in [9] implemented transfer learning techniques on different well-known CNN architectures, including ResNet50, MobileNet, InceptionResNetV2, DenseNet121, and Xception. The best-performed model, DenseNet121, is retrained using the TrashNet dataset with pre-trained weights and achieved 95% test accuracy. In the same year, the authors in [10] used GoogleNet as fine-tuned models and obtained a classification accuracy of 97.86% on the TrashNet dataset.

Overall, the classification accuracy of the TrashNet dataset trained models has increased dramatically from 22% to 97.86%. However, these models' performance may worsen when the difference between training set data and test set data is significant. For example, the lighting level difference between test images and training images can lead to less accurate predictions. Similar practical problems such as waste contamination, shape-shifting material, positioning of waste, and foreign objects may also affect the classification models.

1.2 Research paper structure

This research paper is organized into five chapters, including introduction for Chapter 1, literature review for Chapter 2, research proposal for Chapter 3, result analysis and discussion for Chapter 4, and conclusion for Chapter 5.

An overview of waste image classification is given out at the beginning of the literature review chapter, then background theories and definitions are introduced, and related waste classification papers are reviewed and discussed at the end of Chapter 2. Next, Chapter 3, the research proposal chapter consists of five sections, motivation, aims and objective, research methodology, work plan, and expected outcome and impact. Experimental results are then analysed and discussed in Chapter 4, including dataset split, training parameter, and data augmentation techniques. Lastly, the conclusion is given out in Chapter 5 to summarise this MSc project.

2. Literature Review

2.1 Waste image classification overview

The United Kingdom (UK) Government defines waste as materials discarded by the producers or holders [11]. More specifically, discarding includes activities such as throwing away and recycling [11]. In general, waste can be classified into five types, "construction and demolition waste, packaging waste and recyclables, electronic and electrical equipment, vehicle and oily wastes, and healthcare and related wastes [12]." Among these five waste types, "packaging waste and recyclables" waste type is IBM Wastenet project's focus. According to the IBM Wastenet team, the Wastenet project aims at improving the UK's low households recycling rate (45.5% in 2017) [5]. Recyclables can be defined as waste materials which are reprocessed into products through recycling and recovery operations [13]. Common recycling waste types are paper, cardboard, plastic, metal, wood, and glass [14].

Waste must be classified before sending for recycling since the recycling process can vary depending on the recycling waste types. For example, paper waste recycling process involves cleaning, deinking, refining, and papermaking [15]. On the other hand, metal waste recycling involves operations such as shredding, melting, purification, solidifying, and transportation [16]. The IBM Wastenet project focuses on helping people throwing rubbish in the correct bins [4]. Recycling bin colours standard provided by the Manchester City Council is used here for illustration: blue bins can take waste such as cardboard, waste paper, and boxes; brown bins can take plastic bottles, food tins, glass bottles, and cans; green bins can take food waste, grass cuttings, and cut flowers [17].

The author in [18] categorizes automated sorting techniques for recycling waste into direct sorting and indirect sorting methods. Direct sorting methods can recognize and separate different recycling waste types using material properties only [18]. On the other hand, indirect sorting methods can only identify different recycling waste types but need the help of robotic arms to separate them [18]. The IBM Wastenet project performs recycling waste sorting using indirect sorting method – "optic based sorting". Optic based sorting method utilizes camera-based sensors to take waste images, and identify recycling waste types through visual cues, such as colour, shape, and texture [18]. The development of machine learning enables image-based recycling waste classification at the edge with various advantages: better privacy and safety, lower costs, and faster predictions.

Section 2.1 gives a general overview of waste types, recycling waste types, and recycling waste classification methods which are related to the IBM Wastenet project. The literature review section is structured as follows. In section 2.2, background theories and definitions are introduced and explained. In section 2.3, recycling waste classification projects are discussed and critically reviewed. In section 2.4, issues regarding other waste classification projects and their solutions are presented. In section 2.5, research gaps are identified.

2.2 Background theories

In section 2.2, background theories and definitions related to the recycling waste image classification are introduced. These background theories can be split into five subsections: image classification algorithms, transfer learning techniques, data splitting, training parameters, and data augmentation techniques.

2.2.1 Image classification algorithms

A. Machine learning

The development of machine learning enables computers to behave as humans do [19]. In this project, machine learning gives computers the ability to classify recycling waste into different types based on waste images. Machine learning algorithms can be split into two categories: supervised learning and unsupervised learning. The key difference between them is the availability of true labels. For images, supervised learning and unsupervised learning algorithms are targeting classification and clustering problems, respectively.

$$Y = f(X) \qquad \text{(Equation 2.1)}$$

Equation 2.1 gives the basic idea of supervised learning, where Y is true labels and X is training set images. To be specific, supervised learning is to estimate the mapping function (f) between images and labels, and the estimated mapping function is then used to classify new images [20]. On the other hand, unsupervised learning is to estimate the mapping function without image labels through analysing input data's underlying structures [20]. In this project, supervised learning algorithms are chosen since true labels are available for an estimation of accurate mapping function. Common supervised learning algorithms are Artificial Neural Network (ANN), random forest (RF), support vector machines (SVM), K-nearest neighbour (KNN).

B. Artificial neural network (ANN)

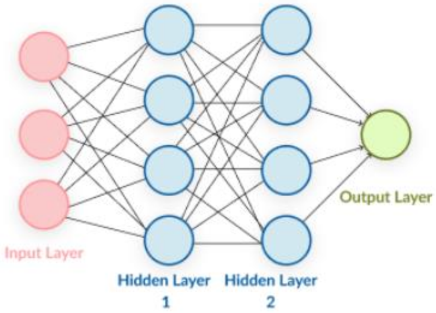


Figure 2-1. example illustration of traditional Artificial Neural Network [21]

Figure 2-1 shows a simplified example of traditional ANN – fully connected neural network (FCNN), where all neurons in the one layer are connected to neurons in the next layer [22].

The connections between neurons in different layers are assigned with weights, and the weighted sum of all neurons in one layer is the input to the next layer neurons [23]. From Figure 2-1, ANN consists of an input layer, multiple hidden layers, and an output layer, where each layer is made of at least one neuron. For image classification tasks, images are fed into the input layer neurons, and neurons in the output layer classify images into different classes [23].

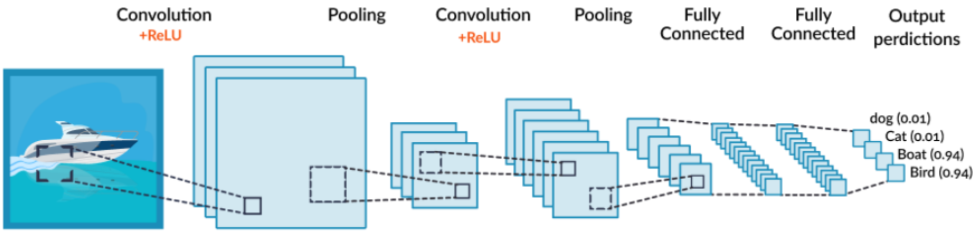


Figure 2-2. example illustration of Convolutional Neural Network [21]

Convolutional Neural Network (CNN) is a category of ANN which performs well in computer vision tasks. The major difference between traditional ANN and CNN is layer dimensionalities. CNN has a three-dimensional layer structure which is designed for analysing Red, Green, and Blue (RGB) images, where traditional ANN has an only one-dimensional layer structure [24]. As shown in Figure 2-2, CNN consists of three types of layer: convolutional layer, pooling layer, and fully connected layer. The input image is split into many small regions, where convolutional layer neurons are grouped and process the corresponding image region [24]. Moreover, the convolutional layer processed image is then downsampled through pooling layer. The most used pooling layer in CNN is the max-pooling layer, where only the maximum number in each pool is reserved. Finally, fully connected layers are connected to the final output layer to process the extracted features and output different class probabilities

C. *Random forest (RF)*

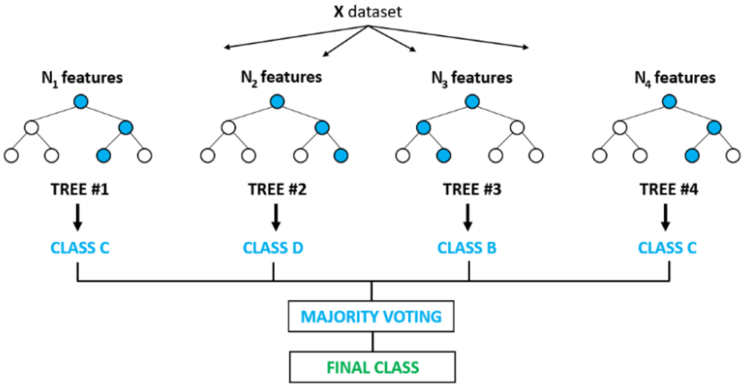
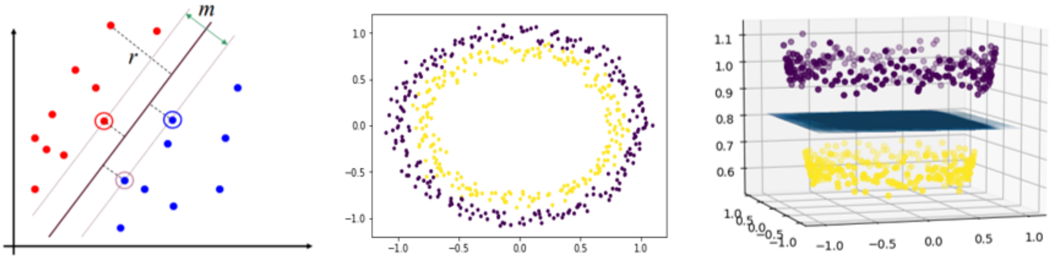


Figure 2-3. example illustration of Random Forest [25]

Figure 2-3 shows an example of random forest model which consists of four decision trees. These decision trees are constructed from the bootstrapped dataset, a majority vote on predictions from decision trees is then presented as the random forest model's prediction. As shown in Figure 2-3, class C is random forest model's prediction since it is the most predicted class. However, random forest classification algorithm cannot perform image classification tasks without the help of feature extraction techniques.

D. Support vector machines (SVM)



(a) (b) (c)

Figure 2-4. example illustration of, (a): linear Support Vector Machines (SVM) [26], (b): non-linear dataset [27], (c): high-dimensional linear SVM [27]

Support Vector Machines (SVM) is a discriminative classifier that separates data points by constructing a hyperplane [28]. Figure 2-4 (a) illustrates how linear Support Vector Machines (SVM) performs binary classification tasks on a linear-separable dataset.

$$r = \frac{W^T X_i + b}{\|W\|} \qquad \text{(Equation 2.2)}$$

To be specific, the closest data points are first determined by comparing distances measured between data points and the hyperplane. Distances between data points and the separator can be measured using equation 2.2, where W is a weight vector, X_i is a input vector, and b is bias [29]. Next, the hyperplane is optimized by pushing data points of both classes as far as possible [26]. In other words, the margin m of the hyperplane is maximized. After constructing the hyperplane, data points located on the hyperplane are support vectors.

Linear SVM cannot be applied to non-linear separable datasets directly, shown in Figure 2-4 (b). However, various kernel functions, such as radial kernel functions, sigmoid, and polynomial, can be used to project non-linear datasets into a high-dimensional feature space. As shown in Figure 2-4 (c), the non-linear dataset becomes linear separable when projected into a higher dimensional space and can be separated by linear SVM using a hyperplane. SVM also needs the help of feature extraction techniques to perform image classification tasks since SVM cannot extract important features from images.

E. K-nearest neighbour (KNN)

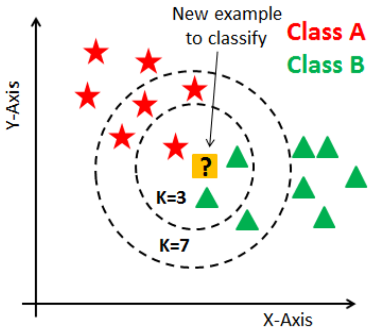


Figure 2-5. example illustration of K-Nearest Neighbour [30]

K-Nearest Neighbour (KNN) classification algorithm stores and plots all training data points, where k represents the number of closest data points. KNN model predicts new instances by identifying k closest data points, the majority class of k closest data points is chosen as the model prediction [30].

As shown in Figure 2-5, the KNN model predicts new instance as class B when k=3, and class A when k=7. KNN model is fast to train but has long prediction time. More specifically, KNN model is trained by simply storing all training data points, and makes predictions by calculating distances between new instances and all training data points [30]. Same as SVM and random forest, KNN also needs the help of feature extraction techniques to perform image classification tasks.

2.2.2 Transfer Learning techniques

Transfer learning is a machine learning technique that helps improve the learning of a target task by transferring knowledge learnt from one or more related source tasks [31]. This technique is often used when there is a lack of training data in the target domain.

$$\text{domain } D = \{X, P(X)\};\ \text{task } T = \{Y, f_T(\cdot)\} \qquad \text{(Equation 2.3)}$$

As illustrated in equation 2.3, the author in [32] gives definitions on domain and task. A domain D consists of two components, feature space X and probability distribution of feature space $P(X)$ [32]. Also, a task T consists of label space Y and predictive function $f_T(\cdot)$, where the predictive function is learnt from the training data [32].

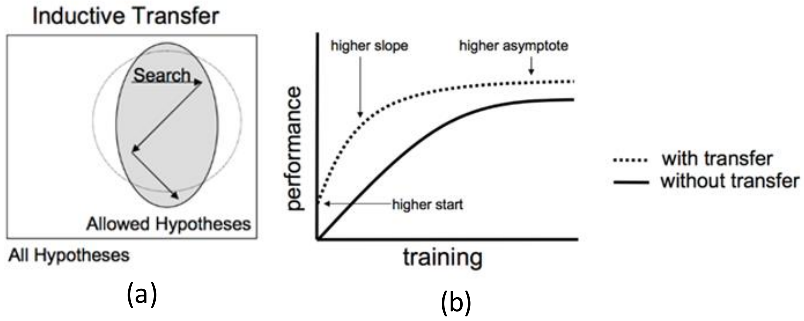


Figure 2-6. example illustration of, (a): inductive transfer [33], (b): potential performance improvements in inductive transfer [33]

Transfer learning can be split into three categories, unsupervised transfer learning, transductive transfer learning, and inductive transfer learning [32]. Among these categories, Inductive transfer learning is the specific form when applying transfer learning techniques to deep learning neural network [33]. Source and target tasks are different but related in inductive transfer learning; also, source and target domains can either be the same or different but related [32].

As shown in Figure 2-6 (a), inductive transfer learning defines and narrows the search scope from all hypothesis to allowed hypothesis [33]. All hypothesis is the search scope when models are initialised randomly, and allowed hypothesis is defined when models are initialised with parameters from source task models [33]. The allowed hypothesis is narrowed in a beneficial way if source tasks and target tasks are related, which improves target task model performance, shown in Figure 2-6 (b) [33]. More specifically, the target task model has better initial performance since knowledge of the source task model is inherited. Furthermore, the target task model learns quicker since the search scope is narrowed in a beneficial way. With the correct optimisation direction, the trained target model performs better than the model with no transfer learning techniques applied.

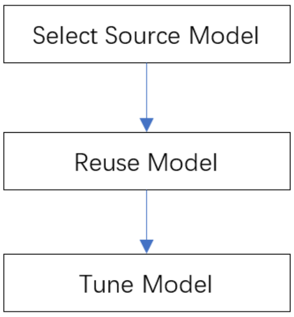


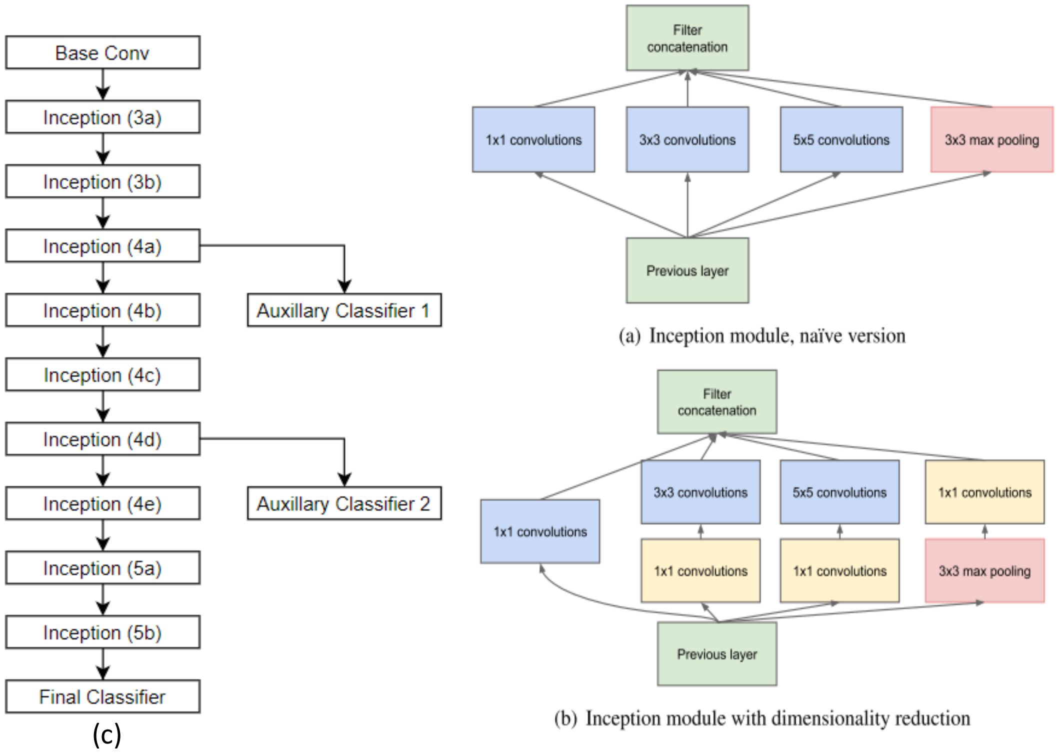
Figure 2-7. flowchart of transfer learning approach [33]

Figure 2-7 shows the steps of applying transfer learning to deep learning neural networks. Firstly, a related source task model needs to be selected to transfer general and useful knowledge from source to target tasks [33]. For image classification models, ImageNet dataset is a good example of source domain since it consists of 14 million images across 1000 classes [34]. Models trained using the ImageNet dataset are assumed with general and related features extracted which are good source task models choices.

Secondly, pre-trained source task models are reused to perform target tasks. In general, final classifiers of source task models are modified to perform target tasks. For example, the number of nodes in the final classification layer needs to be reduced from 1000 to 5 to transform an ImageNet trained model to perform a five-class classification task. Moreover, target task models are initialised with pre-trained weights to inherit source task models' knowledge. Freezing is often used together with transfer learning to retain extracted low-level features, where freezing is to stop training weights in one or more layers. Finally, target task models are tuned to produce final classification models. Layers of freezing, model structures and other training parameters are determined in this stage to produce the best-performed model.

2.2.3 ImageNet models

ImageNet classification models are good choices for source task models since the ImageNet dataset is large and diverse. Winners of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) are expected to extract general and useful features from the ImageNet dataset. ILSVRC is also called ImageNet competition since all participated models are trained using the ImageNet dataset [35]. For example, AlexNet, GoogleNet, VGG and ResNet are all winner models of ILSVRC at different years. Among these winner models, GoogleNet structure is introduced in detail since it is used throughout this project.



(a) Inception module, naïve version

(b) Inception module with dimensionality reduction

(c)

Figure 2-8. GoogleNet model structure (c) [36], naïve inception module (a) [36], Inception module with dimensionality reduction (b) [36]

Figure 2-8 (c) shows the main components of the GoogleNet model. As can be seen, the GoogleNet model is built from many inception modules and two auxiliary classifiers. Figure 2-8 (a) shows the naïve version inception module, where convolutional layers with size 1x1, 3x3, 5x5 are added in parallel to approximate optimal local sparse structure [36]. Additionally, the 3x3 max-pooling layer is added in parallel to provide downsampling ability [36]. However, the stacking of naïve inception modules can lead to computational blow up since 3x3 and 5x5 convolutions are computationally expensive [36]. Figure 2-8 (b) is the inception module form used in the GoogleNet where 1x1 convolutions are placed before 3x3 and 5x5 convolutions to reduce computations [36].

Moreover, two auxiliary classifiers are added to combat the vanishing gradient problem [36]. In general, loss of the final classifier is calculated and backpropagated to the input layer. However, the gradient of loss reduces dramatically as the number of backpropagated layers increases [37]. In other words, the vanishing gradient problem is often encountered when neural networks are too deep. GoogleNet solves this problem by calculating the final classifier loss together with auxiliary classifier losses. With more information carried out in auxiliary classifiers, gradients of loss are less likely to vanish [38].

2.2.4 Data splitting

A. Dataset

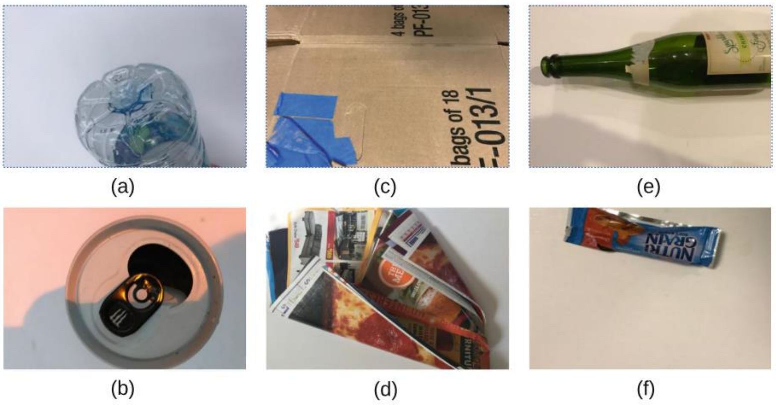


Figure 2-9. Sample images of TrashNet dataset [39], (a): plastic; (b): metal; (c): cardboard; (d): paper; (e): glass; and (f): general trash.

The TrashNet dataset is the only found annotated public recycling waste image dataset, and most recycling waste classification projects are based on this dataset. Figure 2-9 shows six different types of waste contained in the TrashNet dataset. The authors in [8] collect and build this dataset, which contains 2527 images across six waste types (plastic, metal, cardboard, paper, glass, and general trash) [7]. To be specific, TrashNet dataset consists of 501 glass images, 594 paper images, 403 cardboard images, 482 plastic images, 410 metal images, and 137 trash images. These images are taken using iPhones with a white posterboard as the background under sunlight or room light [7]. On the other hand, a private waste image dataset, called VN-Trash, contains 5904 images across three classes, including organic, inorganic, and medical waste.

B. Dataset split

Dataset needs to be split into different sets for the training of machine learning models. The most used dataset split method in machine learning is training/validation/test splits, where a dataset is split into training, validation, and test sets.

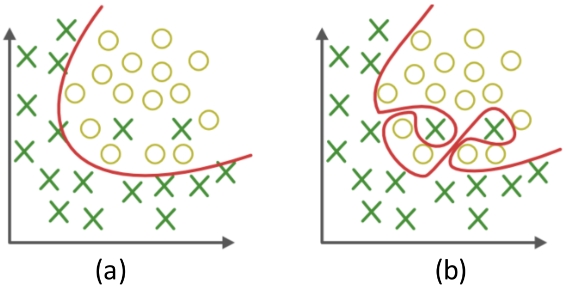


Figure 2-10. example illustration of, (a): Appropriately fitting, (b): Overfitting [40]

For an image classification task, images used to train the classification model are separated into a set, called training set. The training set classification result does not reflect the model's generalizability because of overfitting. Figure 2-10 illustrates the difference between appropriately fitting and overfitting, where the overfitted model fits training data points too well, including noisy data [41]. More specifically, the model (shown in Figure 2-10, a) generalizes better than the model (shown in Figure 2-10, b), but with lower training set classification accuracy. As a result, a separate set, called a validation set, is needed to evaluate the model's performance.

The validation set images are good generalizability indicators since they do not appear in the training set. Therefore, the validation set is used to determine if the model is over-trained or to tune model parameters. The model with the highest validation accuracy or the lowest validation loss is often selected to be the final classification model. Another concern is that the model is manually overfitted to the validation set since parameters are tuned to achieve better validation classification results. In this case, an additional set, called a test set, is needed to evaluate the final model's real performance by showing some images the model has never seen.

C. Cross validation (CV) techniques

Cross validation (CV) techniques are used to split datasets into training/validation/test sets. The author in [42] introduces two commonly used CV types: Hold-out CV and K-fold CV.

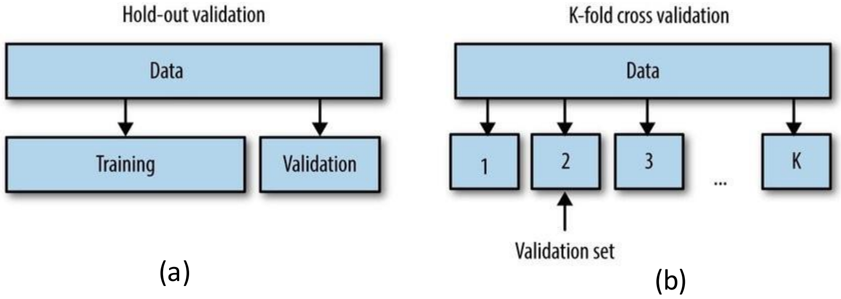


Figure 2-11. example illustrations of, (a): Hold-out CV, (b): K-fold CV [43]

Figure 2-11 (a) and Figure 2-11 (b) illustrate how Hold-out CV and K-fold CV split a dataset into training and validation sets. The main difference between them is the number of splits, where Hold-out CV has one split, and K-fold CV has K splits. Hold-out CV split a dataset into training and validation sets with specified ratios, such as 80% for the training set and 20% for the validation set. On the other hand, K-fold CV first splits a dataset into K subsets (with a label from subset 1 to subset k). For K times, a different subset is selected to be the validation set, and the remaining subsets are combined and used as the training set. As a result, one model is obtained using Hold-out CV, and K models are obtained using K-fold CV.

In comparison, K-fold CV provides more information on a model's generalizability. K-fold CV uses all data as the validation set, but Hold-out CV only uses a small proportion of data. In exchange, K-fold CV models take more computational power and time to train than one Hold-out CV model.

D. Sampling strategies

Simple random sampling (simple RS) and stratified random sampling (stratified RS) are two commonly used sampling strategies in data splitting. The aim of various sampling strategies is to represent the population distribution using a small subset.

$$p(x \in Ts) = \frac{n_s}{n} \qquad \text{(Equation 2.4) [42]}$$

The same probability is given to each data point in a dataset using the simple random sampling strategy [42]. Equation 2.4 gives the sampling probability p of sample x belonging to a sampled subset Ts using the simple RS strategy, where n_s is the sampled subset size, and n is the population size [42].

$$n_h = \frac{N_h}{N} \times n \qquad \text{(Equation 2.5)[44]}$$

Stratified RS divides a dataset into different groups, and randomly draws a proportional size of samples from each group. Equation 2.5 calculates sample size for each group, where n_h is sample size for group h, N_h is the population size for group h, N is the size of population dataset, and n is the total sample size [44]. In comparison, stratified RS takes more computational power and time than simple RS to process, but gives better distribution representation when dealing with uniformly distributed data.

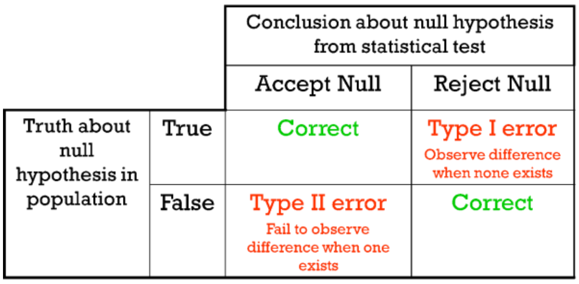
E. Statistics data analysis

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \qquad \text{(Equation 2.6)}$$

The sample mean is often used to compare model performances (such as average classification accuracy and average loss), which can be calculated using equation 2.6. The numerator $\sum_{i=1}^{n} X_i$ in equation 2.6 sums up all samples, and the denominator n is the sample size.

$$S^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{X})^2}{n-1} \qquad \text{(Equation 2.7)}$$

The sample variance is often used to evaluate how far are samples spread out from the sample mean [45], which can be calculated from equation 2.7. The numerator $\sum_{i=1}^{n} (x_i - \bar{X})^2$ in equation 2.7 sums up the squared difference between sample data points and the sample mean, and the $n - 1$ instead of n is used in the denominator to correct the bias created in estimating the population variance [46].



		Conclusion about null hypothesis from statistical test	
		Accept Null	Reject Null
Truth about null hypothesis in population	True	Correct	Type I error Observe difference when none exists
	False	Type II error Fail to observe difference when one exists	Correct

Figure 2-12. statistical hypothesis testing [47]

Statistical hypothesis testing methods are used to analyse population data structures by calculating the confidence level of the null hypothesis, where the null hypothesis is the default statement [48]. From Figure 2-12, the condition of rejecting a true null hypothesis is called Type I error, and accepting a false null hypothesis is called Type II error. On the other hand, accepting a true null hypothesis and rejecting a false null hypothesis are the correct decisions.

$$confidence\ level = 1 - significance\ level \qquad \text{(Equation 2.8)}$$

The confidence level is the probability of obtaining the same results if the sampling process is repeated. Common confidence level choices are 90%, 95%, and 99% [49]. The significance level is the probability of rejecting a null hypothesis when it is true, which is the type I error probability. The significance level is 0.05 when the confidence level is selected to be 95%, which can be calculated using equation 2.8. The p-value is the probability of drawing such extreme samples from the population with the assumption of the null hypothesis being true [50]. The decision is made by comparing the p-value and the significance level. To be specific, the null hypothesis is rejected if the p-value is smaller than the significance level and vice versa.

Statistical hypothesis testing methods can be further split into two types: parametric tests and non-parametric tests. The major difference between them is the test assumptions. To be specific, parametric tests have assumptions (such as normal distribution) of the underlying structures of the population data, but non-parametric tests do not. Therefore, normality tests are often carried out before applying hypothesis testing.

$$W = \frac{(\sum_{i=1}^{n} a_i x_i)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \qquad \text{(Equation 2.9)}$$

The Shapiro-Wilk test is a normality test designed for samples with sample sizes smaller than 50 [51]. The Shapiro-Wilk test statistic W is calculated using equation 2.9, where x_i are ordered samples, a_i are constants generated from covariances, variances, and means, and x_i is the sample mean [52]. The null hypothesis of the Shapiro-Wilk test suggests the population data is normally distributed. Furthermore, the p-value is calculated from the sample size and the test statistic using the Shapiro-Wilk tables [53].

$$H_0: \mu_1 = \mu_2 = \mu_3 = \cdots = \mu_k \qquad \text{(Equation 2.10)}$$

$$H_a: \mu_i \neq \mu_j \qquad \text{(Equation 2.11)}$$

The one-way analysis of variance (ANOVA) is a parametric test that can be used to compare population means of two or more groups [54]. Equation 2.10 gives the null hypothesis of the one-way ANOVA: the population means of k groups are equal. Equation 2.11 shows the alternative hypothesis: the population means of at least two groups are statistically and significant different. Furthermore, the one-way ANOVA has three major assumptions: normal distribution, homogeneity of variances, and independent variables [54]. Therefore, normality tests and homogeneity of variances tests need to be carried out before performing ANOVA.

$$H_0: {\sigma_1}^2 = {\sigma_2}^2 = {\sigma_3}^2 = \cdots = {\sigma_k}^2 \qquad \text{(Equation 2.12)}$$

$$H_a: {\sigma_i}^2 \neq {\sigma_j}^2 \qquad \text{(Equation 2.13)}$$

The Levene's test is often used before one-way ANOVA to test the homogeneity of variances assumption. Equation 2.12 gives the null hypothesis of the Levene's test: the population variances for all groups are equal. Equation 2.13 gives the alternative hypothesis: the population variances of at least two groups are statistically and significantly different. The Kruskal-Wallis test can be used to analyse population means if the ANVOA assumptions are violated since the Kruskal-Wallis test is non-parametric.

2.2.5 Training parameters

A. Loss functions

The loss function measures how well a trained model fits the dataset, and machine learning models are trained by optimizing the loss functions to achieve better-fitted models. The loss function produces a large number if the model cannot make correct predictions on a dataset, and the loss is small if the model can. Loss functions can be categorized into two types: classification loss functions and regression loss functions. Only classification loss functions are introduced since this project is a multi-class classification task. Common classification loss functions are cross-entropy (CE) loss and hinge loss.

$$CE\ Loss = -\textstyle\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \qquad \text{(Equation 2.14)}$$

The cross-entropy loss computes the natural logarithm loss of the predicted probability [55]. Equation 2.14 calculates the multi-class classification CE loss, where c is class label, o is true label, y as a binary indicator (0,1) is 0 when o, c does not match and 1 when o, c matches, and p is the predicted probability. In other words, the natural logarithm loss of the correct class predicted probability is computed as the cross-entropy loss. The Softmax function should be used as the final layer activation function to obtain the predicted probability for each class.

$$Hinge\ Loss = \max\ (0{,}1 - y.\hat{y}) \qquad \text{(Equation 2.15)}$$

The hinge loss maximises the classification margin, which is used in the SVM classification algorithm. Equation 2.15 calculates the binary hinge loss, where y is the true label (1, -1), and $\hat{y}$ is the raw output of the prediction function. The hinge loss increases linearly if $\hat{y}$ and y have the opposite sign, which indicates misclassification. On the other hand, the hinge loss is zero if $\hat{y}$ is larger than 1 and $\hat{y}$ and y have the same sign. In other words, the hinge loss is zero only if the model classifies data points correctly with enough confidence. The hinge loss can be used in multi-class classification models by treating multiple classes as one class and compute it. The linear function should be used as the final layer activation function for linear SVM.

B. Optimizers and batch sizes

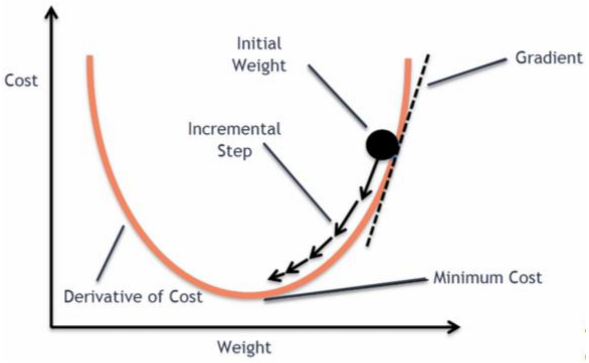


Figure 2-13. example illustration of gradient descent algorithm[56]

$$w = w - \alpha \frac{\partial loss}{\partial w} \qquad \text{(Equation 2.16)}$$

Gradient descent is the most used optimisation method in machine learning. Figure 2-13 and equation 2.16 illustrate how gradient descent minimises the loss (also called cost), where w is model weights, α is learning rate, and $\frac{\partial loss}{\partial w}$ is the gradient of the loss with respect to weights. In other words, the loss is minimised by updating model weights in the opposite direction of the loss function's gradient [57]. More specifically, the loss is minimised when the loss function's gradient is almost 0 [58]. Gradient descent optimisation methods can be split into two categories, classic gradient descent optimizers and extensions of gradient descent optimizers.

There are three classic gradient descent optimizers , batch gradient descent (BGD), stochastic gradient descent (SGD), and mini-batch gradient descent (MBGD). The major difference between them is the gradient of loss calculation. The loss function is evaluated on the entire training set in BGD, on each training sample in SGD, and on n training samples in MBGD [59]. In comparison, SGD converges the fastest, and BGD provides the most stable convergence. MBGD is an intermediate optimizer which provides a trade-off between the rate of convergence and convergence stability [59]. Batch size is set to use different classic gradient descent optimizers. To be specific, the batch size is set to 1 to use SGD, set to the number of total training samples to use BGD, and to any number between them to use MBGD.

$$m_n = E[X^n] \qquad \text{(Equation 2.17)}$$

Adaptive moment estimation (Adam) and Adadelta are two commonly used extensions of gradient descent optimizers. Adam optimizer updates model weights by computing the first and the second moments of the loss gradient [60]. The n-th moment m_n can be calculated using equation 2.17, where n is the n-th moment, E is the expected value, and X is the random variable [60]. More specifically, the first moment is the mean of the loss gradient, and the second moment is the uncentered variance of the loss gradient [60]. On the other hand, Adadelta optimizer updates model weights based on a moving window of the loss gradient [59].

C. Learning rates

Learning rate is one of the most important hyper-parameters since it decides the step size of weights update in gradient descent. As shown in equation 2.16, weights are reduced in the opposite direction of loss gradient with proportional to the learning rate α. Learning rate that is too small will result in a slow training process and failure in escaping the local optima [61]. On the other hand, using a learning rate that is too large will result in a suboptimal convergence [61].

The most used technique of choosing an appropriate learning rate is learning rate finder [62]. This technique trains the model for a few epochs with exponentially increased learning rate, and validation loss is evaluated and recorded every batch [62].

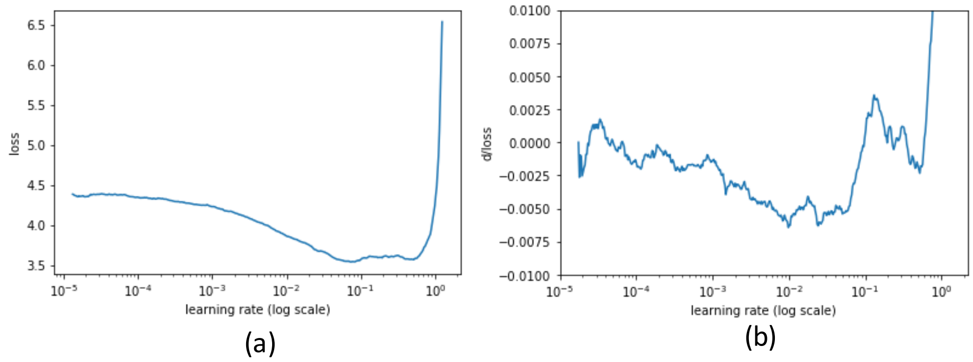
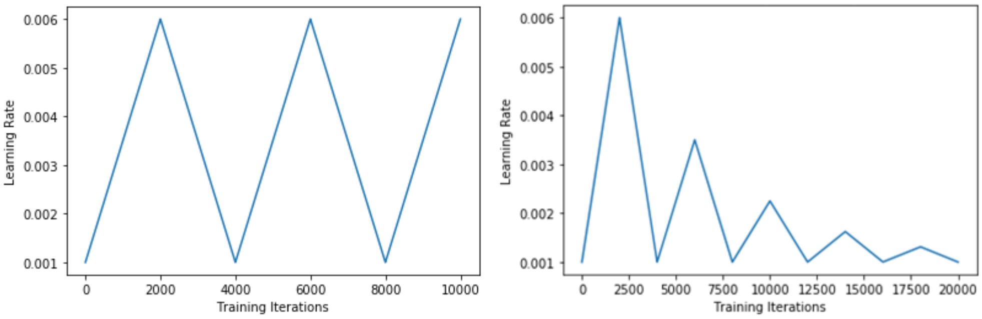


Figure 2-14. plots of, (a): validation loss against learning rate (log scale), (b): rate of change of validation loss against learning rate [63]

Figure 2-14 (a) shows an example of how validation loss changes with learning rate. From Figure 2-14 (a), the learning rate between 1e-5 and 1e-3 conducts a minimal reduction in the validation loss, and learning rate between 5e-1 and 1 increases the validation loss significantly. An appropriate learning rate can be any number between 1e-3 and 5e-2, where the validation loss is reduced significantly between this range.

From Figure 2-14 (b), the learning rate of the largest validation loss reduction can be identified by taking derivatives of validation loss with respect to the iteration number [63]. In this case, the learning rate of 1e-2 is a good choice since the validation loss is reduced the most when applying this learning rate.

Various learning rate schedulers are used to achieve better convergence, such as constant learning rate, decay learning rate and cyclical learning rate. Constant learning rate scheduler is the default learning rate scheduler where no additional changes are applied to the gradient descent optimizer except defined starting learning rate. On the other hand, decay learning rate scheduler applies a percentage decrease over time or steps. Lastly, cyclical learning rate scheduler allows the learning rate to vary within a selected range [64].



(a) (b)

Figure 2-15. cyclical learning rate policies, (a): *triangular1*, (b): *triangular2* [65]

Figure 2-15 shows two mostly used cyclical learning rate policies, *triangular1* and *triangular2*. From Figure 2-15 (a), learning rate in *triangular1* policy is varied cyclically (with a cycle of 4000 iterations) between the maximum learning rate (0.006) and the minimum learning rate (0.001). On the other hand, the maximum learning rates of *triangular2* policy, shown in Figure 2-15 (b), are halved each cycle. The author in [64] suggests that cyclical learning rate schedulers can help achieve the same or even better training results with a higher rate of convergence. This is because the increase in learning rates can help models escape the local minimum, and the decrease in learning rates can help achieve stable convergence.

D. Layers freezing

Layers freezing is often used with transfer learning technique to fine-tune the model. To be specific, weights in a frozen layer are not updated when training the model. Layers freezing can be used to protect the low-level features extracted from pre-trained models from destroyed. In a transfer learning model, all layers except the final classifier are frozen at the beginning. Next, layers from the output to the input are unfrozen gradually to fine-tune the model.

E. Stopping criterions

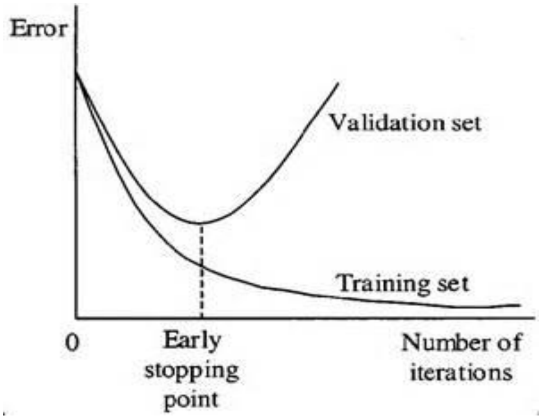


Figure 2-16. example illustration of early stopping [66]

The stopping criterion is rules applied to stop the model from training. Early stopping is a commonly used training technique to help model stop training at the correct point. Figure 2-16 shows an example of using early stopping to avoid the overfitting problem, where validation set error increases after the early stopping point. In practice, the number of patience epochs is defined to use the early stopping technique. For example, a model will be stopped from training if validation errors obtained in the next 20 epochs are not lower than the lowest validation error obtained when setting the patience epochs to be 20.

2.2.6 Data augmentation techniques

Data augmentation techniques are often used when the dataset is limited and lack of diversity. For image datasets, geometric and colour space transformations are two basic classes of data augmentation techniques [67]. Augmented images are added to enhance the diversity of training set, and models trained using the augmented dataset can generalize better [67].

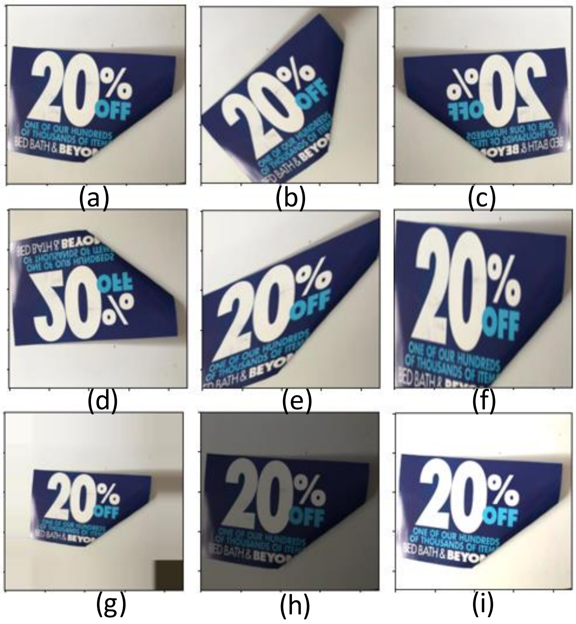


Figure 2-17. Image data augmentation examples, (a): original image, (b): rotation, (c): horizontal flipping, (d): vertical flipping, (e): shearing, (f): zoom in, (g) zoom out, (h): brightness decrease, (i): brightness increase

Figure 2-17 shows examples of geometric and colour space transformation techniques, including rotation, flipping, shearing, zoom, and brightness control. Figure 2-17 (a) shows the original image, which is an image of paper waste with numbers on it. Figure 2-17 (b) gives an example of rotation, where rotation can be clockwise and counter-clockwise within 360 degrees.

Figure 2-17 (c) and figure 2-17 (d) show horizontal flipping and vertical flipping, where horizontal flipping mirrors an image across the horizontal axis, and vertical flipping mirrors an image across the vertical axis. Figure 2-17 (e) gives an example of shearing, where shearing can happen in the horizontal direction and vertical direction with a defined shearing angle.

Figure 2-17 (f) and Figure 2-17 (g) give examples of zoom in and zoom out, where zoom in focuses on the object and zoom out focuses away from the object. Lastly, Figure 2-17 (h) and Figure 2-17 (i) give examples of brightness decrease and brightness increase, where brightness decrease makes an image darker, and brightness increase makes an image lighter.

2.3 Recycling waste image classification projects

2.3.1 Literature overview

In section 2.3, seven recycling waste image classification projects are critically reviewed. Differences between these projects are analysed and discussed, and general approaches are identified. These seven projects have different focuses, including datasets, model architectures, machine learning algorithms, and final classifiers.

Table 2-1. Introduction of recycling waste image classification projects

Year	Paper	Paper title	Author
2017	[8]	Classification of Trash for Recyclability Status	M. Yang et al.
2018	[9]	RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks	C. Bircanoglu et al.
2018	[68]	Image Based Trash Classification using Machine Learning Algorithms for Recyclability Status	M. Satvilkar et al.
2018	[10]	Fine-Tuning Models Comparisons on Garbage Classification for Recyclability	U. Özkaya et al.
2019	[69]	A Novel Framework for Trash Classification Using Deep Transfer Learning	A. Vo et al.
2019	[39]	Automatic Image-Based Waste Classification	V. Ruiz et al.
2020	[70]	WasteNet: Waste Classification at the Edge for Smart Bins	G. White et al.

Table 2-1 shows basic information about seven waste image classification projects, including paper titles, publication year, and authors. Table 2-1 is sorted in ascending order of publication year, from 2017 to 2020. These projects are very recent because there was not a recycling waste image dataset available before 2017. In 2017, M. Yang and others hand-collected a recycling waste image dataset, called TrashNet, and made it public. Unlike other waste-related projects, these seven projects target on recycling waste image classification. For example, SpotGarbage project in [71] is not discussed here because it focuses on detecting garbage in images rather than classifying recycling waste images.

2.3.2 Dataset review

Table 2-2 gives dataset information of seven introduced projects in Table 2-1, including dataset, cross-validation method, dataset split ratios, and sampling strategies. Moreover, Table 2-2 is sorted in the same order as Table 2-1, with paper reference numbers noted at the front of each row.

Starting with datasets, all waste classification projects used TrashNet dataset, and one of these projects used both VN-Trash and TrashNet dataset. The general use of TrashNet dataset is because it is still the only publicly available recycling waste image dataset. In comparison, VN-Trash is less preferred since it is a private and non-recycling waste image dataset. Before the publishing of TrashNet dataset (in 2017), lack of an appropriate dataset was the biggest obstacle that stops recycling waste image classification research.

Furthermore, two of these projects omitted “trash” class images for two reasons. Firstly, the “trash” class contains only 137 images, which is about 1/5 of other class sizes. Secondly, the “trash” class contains mixed types of general waste and does not provide a precise classification boundary for algorithms to learn. In comparison, TrashNet with “trash” class images omitted is more suggested to use than the whole TrashNet dataset and VN-Trash.

Table 2-2 . Dataset split information of recycling waste image classification projects

Paper	Dataset	CV method	Split ratio (in %)	Sampling strategy
[8]	TrashNet (omit “trash” class)	Hold-out CV	70/ 13/ 17	Simple RS
[9]	TrashNet	Hold-out CV	Not given	Simple RS
[68]	TrashNet (omit “trash” class)	10-fold CV	56/ 19/ 25	Stratified RS
[10]	TrashNet	Hold-out CV	Not given	Simple RS
[69]	VN-Trash, TrashNet	Hold-out CV	60/ 20/ 20	Simple RS
[39]	TrashNet	5-fold CV	80/ 10/ 10	Simple RS
[70]	TrashNet	Hold-out CV	50/ 25/ 25	Simple RS

Secondly, Hold-out CV and K-fold CV are used to split training set further into training and validation set in these projects. As can be seen from Table 2, only two projects use K-fold CV to reduce bias by averaging K models’ performances, and their K choices are 5 and 10. On the other hand, the other projects do not mention their dataset split methods. However, it is reasonable to assume they use Hold-out CV since Hold-out CV is to split dataset directly with ratios specified. One possible reason behind the lack of use of K-fold CV in these projects is computational constraints. The authors in [8], [68] reduce model complexity and image quality respectively to save computational power. Therefore, the selection between Hold-out CV and K-fold CV is a trade-off between computational power and model bias.

Thirdly, five project papers mention what ratios they use to split the dataset into training, validation, and test sets. These split ratios are all different and lack of explanations. However, two trends can still be observed from these split ratios. Firstly, the training set size is made the biggest to give the model more data to learn. Secondly, the validation set size is made the same or similar to the test set, so the validation set tuned model can perform better on the test set. Different split ratios need to be experimented to confirm the best split ratio of TrashNet dataset.

Lastly, simple RS and stratified RS are used in these projects. Only one project uses stratified RS to split dataset. The other project papers do not mention their dataset sampling strategy, so the most used sampling strategy, simple RS, is assumed to use in their project. Again, experiments of sampling strategies are needed to know which sampling strategy works better since these papers do not provide explanations on their sampling strategy selection.

2.3.3 Learning algorithm and model structure review

Table 2-3. learning algorithm and model used in recycling waste image classification projects

Paper	Transfer Learning	Model transferred	Weights transferred	Learning Algorithm	Classification Result
[8]	Applied	AlexNet	No	SVM	63%
				CNN	75%
[9]	Applied	DenseNet	Yes	CNN	95%
[68]	Not applied	None	No	SVM	65.67%
				RF	62.61%
				XGBoost	70.1%
				KNN	52.5%
				CNN	89.81%
[10]	Applied	GoogleNet	Yes	CNN	97.86%
[69]	Applied	DNN-TC	Yes	CNN	94%
[39]	Applied	ResNet	No	CNN	88.66%
[70]	Applied	DenseNet	Yes	CNN	97%

Table 2-3 gives information about model and algorithm used in recycling waste image classification projects, including model and weights transfer, learning algorithms, and classification results. Furthermore, Table 2-3 is also sorted in the same order as Table 2-1, with paper reference numbers noted at the front of each row.

Among these projects, only project in [68] does not use any transfer learning techniques because it focuses on testing different machine learning algorithms. Transfer learning techniques, including model transfer and weights, are often used in these projects to improve model performances. Transferred models that achieve the highest classification result in each project are presented here, where DenseNet model achieves the highest classification accuracy in two projects (97% and 95%), and GoogleNet achieves overall the highest classification accuracy (97.86%). Next, weights transferred models generally gives higher classification accuracy (96% in average) results than none-weights transferred models (70% accuracy in average).

Moreover, different learning algorithms are used in these projects, including SVM, CNN, RF, XGBoost, and KNN. Among these learning algorithms, CNN is the most used machine learning algorithm and achieves the highest classification accuracy (91% in average) and mostly used in these projects. In comparison, CNN learning algorithm works better than other learning algorithms for two reasons. Firstly, CNN is easy to use since it can extract features from images without the help of additional feature extraction techniques. Secondly CNN learning algorithm works well on image classification tasks because it can extract important features by passing images through pooling layers and convolutional layers. Overall, pre-trained CNN model with weights transferred works the best for recycling waste image classification projects.

2.3.4 Training parameter review

Table 2-4. training parameter of different recycling waste image classification projects

Paper	Loss function	optimizer	Batch size	Learning rate	Training epoch
[8]	Not given	Not given	32	5e-8	60
[9]	Cross-Entropy	Adam, Adadelta	32	1e-3	200
[68]	Cross-Entropy	SGD	32	1e-2	60
[10]	Not given	Not given	Not given	Not given	200
[69]	Not given	Adam	8	1e-3	100
[39]	Not given	SGD	16	2e-4	Earlystopping(25)
[70]	Not given	Not given	Not given	Not given	1000

Table 2-4 shows the training parameters used in different recycling waste image classification projects, including loss functions, optimizers, batch size, learning rates, and training epochs. Table 2-4 is also sorted in the same order as Table 2-1, with paper reference numbers noted at the front of each row. Furthermore, many papers do not give their training parameters selections and reasons behind their choice. Therefore, not given is noted when this piece of information cannot be found in the paper.

Starting with loss function choices, only two of these projects state their loss functions used, which is Cross-Entropy loss. The other projects do not provide this piece of information because Cross-Entropy is the default loss function in classification neural network training. Hinge loss is an alternative loss function that often used to train the maximum-margin classification model, such as SVM. Overall, Cross-Entropy is a good loss function that can be used to train probabilistic classification models.

Secondly, Adam, Adadelta, and SGD optimizers are used in these projects to update weights and minimise loss. The authors in [9] compare two optimizing approaches, Adam optimizer and Adadelta optimizer. In their experiments, the best-performed transfer learning model, DenseNet, achieves 95% test accuracy, which is trained using Adam optimizer. However, further experiments on these three potential optimizers are needed to choose optimizer for this project.

Thirdly, batch sizes, including 8, 16, 32, are used in these projects are shown in Table 2-4. These batch size selections tend to be small because models in reviewed projects are deep and complex, so limited computational power cannot process images in large batch sizes. Furthermore, these batch sizes are set to the power of 2 to map onto physical processors and speed up the training process. Therefore, small numbers in the power of 2 are potential batch sizes selections in this project.

Fourthly, different learning rates are used in these projects to train models, from 5e-8 to 1e-2. The author in [8] used 5e-8 as the learning rate, which is much smaller than in other projects. The small learning rate selection can be one of the reasons that they achieve the lowest CNN model classification accuracy (75%). Also, learning rates need to be selected according to optimizer selections. Learning rate finder is a good approach to choose well-performed learning rates for a model.

Lastly, model training epochs of these projects are shown in Table 2-4. Fixed number of training epochs, such as 60, 100, 200, are often used to train a TrashNet classification model. In comparison, project [39] suggests a more sensible approach, early stopping with 25 patience epochs. In other words, model training is stopped if no lower validation obtained in the next 25 epochs. In comparison, early stopping training provides better flexibility than setting training epochs to a fixed number.

2.3.5 Data augmentation review

Table 2-5. data augmentation techniques of different recycling waste classification projects

Paper	Data augmentation techniques
[8]	Rotation, brightness control, translation, scaling, shearing
[9]	Horizontal flipping, vertical flipping, 15 degrees rotation
[68]	None
[10]	None
[69]	Horizontal flipping, cropping
[39]	40 degrees rotation, 20% width changes, 20% height changes, 20% shearing, 20% zooming, horizontal flipping, brightness control
[70]	Translation, zooming, shearing, rotation, horizontal flipping

Table 2-5 gives information about data augmentation techniques used in different recycling waste classification projects. These techniques are rotation, brightness control, translation, scaling, shearing, horizontal and vertical flipping, width and height changes, and cropping. Table 2-5 is also sorted in the same order as Table 2-1, with paper reference numbers noted at the front of each row.

From Table 2-5, projects [10], [68] do not use any data augmentation techniques. In other projects, the most used techniques are rotation (in 4 projects), flipping (in 4 projects), shearing (in 3 projects), zooming (in 2 projects), and brightness control (in 2 projects). These projects do not give explanations on their data augmentation choices, and the effects of each data augmentation technique on classification models are not clear. Experiments on different data augmentation techniques are needed to determine which techniques to use in this project.

2.3.6 Literature review summary

Seven recycling waste classification projects are reviewed from five aspects, including datasets, learning algorithms, model structures, training parameters and data augmentation techniques.

TrashNet dataset is by far the only available recycling waste image dataset, and “Trash” class image should be omitted for good interpretability. Hold-out CV and K-fold CV are two CV methods that often used together with sampling strategies (simple RS and stratified RS) to split datasets into sets in different ratios. However, there is not an agreed answer or a systematic approach to determine what CV method, sampling strategy, and split ratios to use .

Transfer learning techniques are used in most of these projects to transfer knowledge of other models. Pre-trained model structures, such as AlexNet, DenseNet, GoogleNet, are good waste classification model choices. Furthermore, CNN achieves the highest classification accuracy compared with other learning algorithms, such as SVM, RF, KNN.

Cross-Entropy is the most used loss function in these projects, and different optimizers (such as Adam, Adadelta, SGD) are used to minimise the loss. Small batch sizes, such as 8, 16, 32, are used because of model complexity, and these numbers are set to the power of 2 to speed up the training process. Also, small learning rates are preferred to protect low-level features learnt from pre-trained transfer learning models. To save computational resources, early stopping is preferred than setting training epochs to a fixed number.

Next, different data augmentation techniques are used in these projects to improve the size and quality of datasets. Commonly used techniques are rotation, flipping, shearing, zooming, and brightness control. Experiments on these techniques are needed to determine which techniques to use.

Overall, a lot of research has been done in applying transfer learning techniques to recycling waste image classification tasks, and these research projects achieve great classification results. However, limited research has been done in developing a systematic approach to build recycling waste image classification models. To be specific, some of these waste classification papers do not provide all implementation details, such as dataset split ratios, loss function, and optimizer. The lack of these implementation details makes it hard to reproduce their classification models.

Moreover, most papers do not provide a clear explanation and support evidence on implementation details. For example, none of these papers explains why they split datasets as they described. Therefore, limited useful information on model developing can be summarised from their projects since their work is not well-justified. Lastly, the large research gap can be further differentiated into three small gaps: dataset splitting (how to split datasets), training parameters selection (how to train models), and data augmentation techniques (how to improve quality and size of datasets).

3. Research Proposal

3.1 Motivation

As introduced in the literature review section, the UK's household recycling rate was 45.5% in 2017. In other words, more than half of the household waste was not disposed of environmentally friendly. Some of the other disposal methods, such as landfill and incineration, have negative impacts on the natural environment. Therefore, it is important to improve waste recycling rate by automating the waste classification process.

Recycling waste classification can happen at both the household side and the industry side. To be specific, households should segregate waste into different recycling waste types before putting into bins. After that, the waste industry confirms further segregates waste. The IBM Wastenet project focuses on waste recycling of household side by helping people throw rubbish into the correct bin.

The smart bin introduced by IBM Wastenet team utilizes IBM Watson visual recognition service. In other words, the recognition process is carried out in the cloud. Moving the recognition process to the edge can have a shorter prediction time. Moreover, this project also works on improving waste classification accuracy and adding functionalities, such as classification probabilities and customizable outputs.

3.2 Aims and objectives

This MSc project aims to develop an edge side recycling waste classification model. To be specific, the waste classification model is designed to run on resource-constrained devices, such as microcontrollers, and perform accurate real-time predictions with additional features. Moreover, this mode is designed to solve various image issues, such as lighting levels, object distances, object orientations. Overall, this MSc project works on improve the original IBM Wastenet classification model from three aspects, accuracy level, prediction time, and functionalities.

Objectives of this MSc project are:

- Achieving model classification accuracy above at least 90%
- Reducing model prediction time below typical human visual reaction time (250 milliseconds) [72].
- Adding recycling waste sub-class classification, to be specific, classifying recycling waste into plastic, glass, paper, cardboard, and metal
- Adding prediction probability output, to be specific, a probability scale between 0% to 100%
- Applying image classification model on to videos using cameras

3.3 Research methodology

Normally, the original IBM Wastenet model should be treated as the benchmark model and evaluated at the beginning of this MSc project. However, the IBM Wastenet team does not provide the IBM Wastenet model or its performance statistics. It becomes impossible to improve the IBM Wastenet model accordingly and to quantify these improvements because of the lack of the IBM Wastenet model. As a result, this MSc project is carried out by developing a benchmark model to play the role of the IBM Wastenet model.

The best-performed model in [10] is reproduced to be the benchmark model for two reasons. Firstly, this model achieves the highest classification accuracy (97.86%). Secondly, the authors in [10] do not apply any data augmentation techniques to this model, which makes it a good choice to test various data augmentation techniques. However, this paper [10] does not provide implementation details other than the transfer learning model's name (GoogleNet) and final classifier used (Softmax and SVM), which makes it difficult to reproduce. To solve this problem, an approach of training benchmark model is developed in this project.

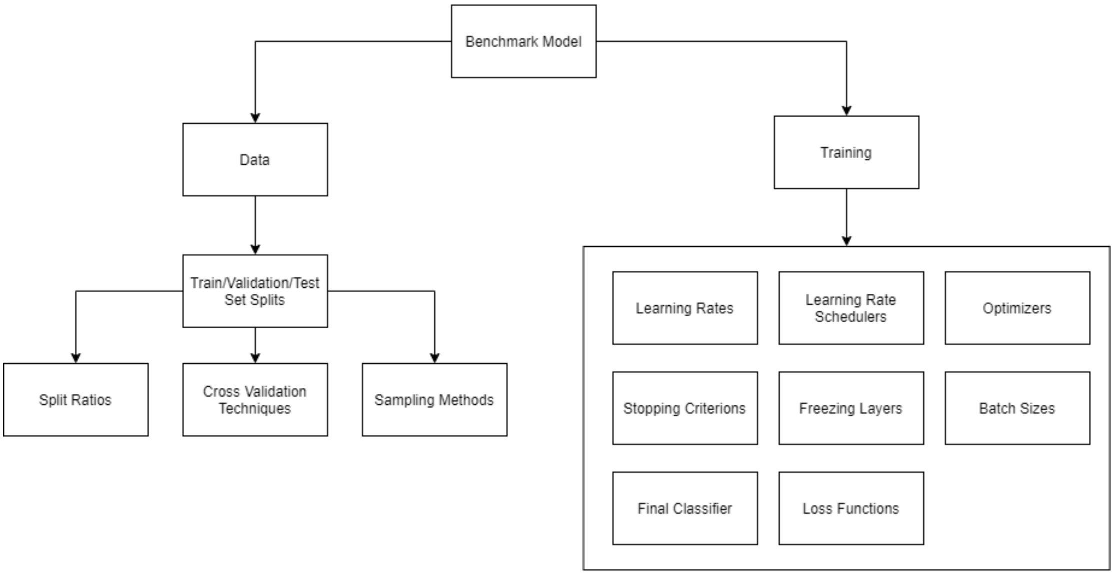


Figure 3-1. Flowchart of developing a benchmark model

Figure 3-1 illustrates how a benchmark model should be developed by splitting datasets and tuning various training parameters. Starting with datasets, split ratios, CV techniques, sampling strategies need to be determined to split datasets into training, validation, and test sets.

As discussed in the literature review section, these reviewed projects do not give an agreed split ratio or an approach to determine it. Instead of randomly choosing one dataset split ratio, split ratios used in these reviewed waste classification projects are adjusted and tested, then the one gives the highest classification accuracy is used to split the TrashNet dataset. These adjusted dataset split ratios (in percentage) are 80/10/10 (training/ validation/ test), 70/15/15, 60/20/20, and 50/25/25.

Next, K-fold CV is used to split the training set further into training and validation sets in this project since it gives lower model bias than using Hold-out CV. However, these reviewed projects still do not give an agreed K selection or an approach to determine it. Instead of randomly choosing a K number, K used in the reviewed projects (K=5 and K=10) are tested, and the one gives lower variance is selected. Lastly, Simple RS and Stratified RS as two sampling methods used in the reviewed waste classification projects are tested, and the one gives higher classification accuracy is selected.

On the other hand, experiments on training parameters (including learning rates, learning rate schedulers, optimizers, stopping criteria, freezing layers, batch sizes, final classifiers, and loss functions) are carried out to make a trade-off between classification accuracy, training time, and prediction time.

Constant learning rate, decay learning rate, and cyclical learning rate are the three most used learning rate schedulers. Experiments on learning rate schedulers are carried out to determine which one gives the highest classification accuracy. Moreover, learning rates used in each learning rate scheduler are determined using the learning rate finder. Next, three optimizers (including SGD, Adam, and Adadelta) are tested, and the one gives the highest classification accuracy is selected.

The number of patience epochs in the early stopping technique is important when making the stopping criteria. Common patience epoch number choices are 10 and 20. In this case, an appropriate patience epoch number can be determined by setting it to a large number, such as 200. On the other hand, layers freezing can be determined by gradually unfreezing layers from top classification layers to bottom input layers, and the one gives the highest classification accuracy is selected.

Different batch sizes are used in the reviewed waste classification projects, including 8, 16, and 32. These batch sizes are tested, and the one gives the highest classification accuracy is selected. Lastly, paper [10] compares SVM classifier with Softmax classifier, and the SVM classifier gives higher classification accuracy. Experiments on classifiers can be carried out to validate the results obtained in paper [10]. In addition, different loss functions are used with respect to the classifier. To be specific, Cross-Entropy loss function is used together with Softmax classifier, and Hinge loss function is used with SVM classifier.

3.4 Work plan

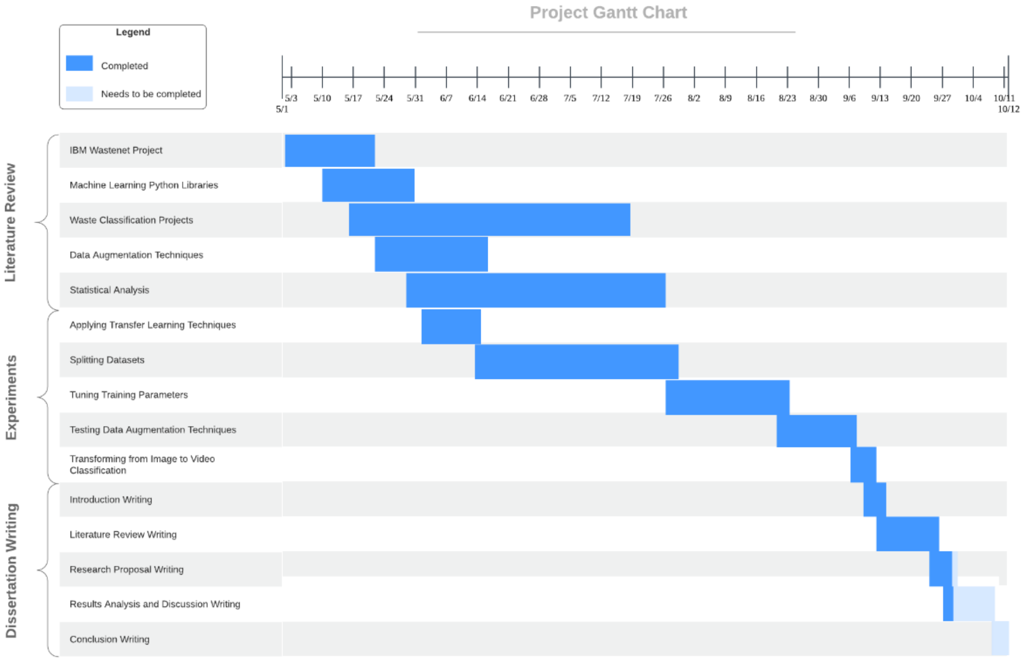


Figure 3-2. Gantt Chart of this MSc Project

A Gantt chart, shown in Figure 3-2, describes schedules of this MSc project. There are three main tasks in this project, and these main tasks can further be split into many sub-tasks. This project starts from May 1st and ends on October 12th. The subtasks need to be completed until now are research proposal writing, results analysis and discussion writing, and conclusion writing.

3.5 Expected outcome and impact

The expected outcome of this project is to help improve the IBM Wastenet model and achieve the objectives described in section 3.2. To be specific, these objectives are 90% above classification accuracy, prediction time less than 250 milliseconds, five-class recycling waste classification, prediction probability display, and video frame classification.

Moreover, an approach of training the recycling waste benchmark model is developed, including datasets splitting, training parameters tuning, and data augmentation techniques applying. Statistical analysis is applied to the dataset splitting to achieve higher classification accuracy.

Furthermore, this MSc project is an open source project. Programming code using TensorFlow and Keras machine learning libraries written in Python is available on GitHub webpage. Lastly, the model developed in this MSc project can be used on embedded devices (such as, a Raspberry Pi) to perform edge side machine learning.

The major impact of this MSc project is to provide future recycling waste image classification projects a benchmark model development approach. In this MSc project, issues of data splitting, training parameters tuning, and data augmentation techniques applying are addressed. An example of these issues is how to split the dataset to give higher classification accuracy, and this is addressed by comparing different split ratios using statistical analysis tools.

Moreover, none of the waste classification projects introduced in the literature review section shares the programming source code. With open source of this MSc project, the future recycling waste classification projects can obtain more implementation details on Python using TensorFlow and Keras Libraries.

4. Results Analysis and Discussion

In this chapter, the experiment results about data splitting (in section 4.1), training parameters tuning (in section 4.2), and data augmentation techniques applying (in section 4.3) are presented, analysed, and discussed.

4.1 Dataset split

Split ratios, sampling strategies, and CV techniques of datasets are selected to achieve higher CNN model classification accuracies. Statistical analysis is performed on the sampled classification accuracies to estimate the population classification accuracies. Normality and homogeneity of variances tests are carried out first to choose between parametric and non-parametric tests.

The research question answered in this section is how to split dataset to train a well-performed recycling waste image CNN classification model. In section 4.1.1, experimental results of the test set's split ratio are analysed and discussed. In section 4.1.2, training and validation sets are split by choosing K numbers of the K-fold CV. In section 4.1.3, the sampling strategy of the entire dataset is determined through experiments. Full statistical analysis proof of dataset split is provided in Appendix A.

Table 4-1. Dataset split results

	CV techniques	Split Ratio	Sampling Strategy
Training Set	10-fold CV	81%	Simple Random Sampling
Validation Set		9%	
Test Set	Hold-out CV	10%	

Table 4-1 presents the final data splitting decisions made in this project, including CV techniques, split ratios, and sampling strategies. Firstly, 10% of the entire dataset is separated into the test set using simple random sampling strategy. Secondly, 10-fold CV is applied and split the rest dataset (90% of the entire dataset) into training and validation sets. More specifically, nine folds are combined and used as the training set each time, and the remained fold is used as the validation set. Lastly, simple random sampling strategy is applied to both Hold-out CV and 10-fold CV.

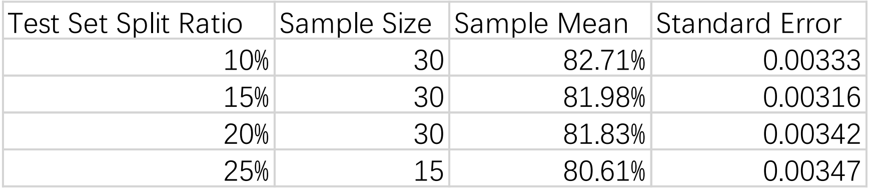
This set of data splitting options achieves the highest classification accuracy for two reasons. Firstly, the model has more images to learn since over 80% of the dataset is split into the training set. Secondly, the 10-fold CV technique can help tune hyper-parameters since the entire dataset except the test set is used as the validation set to evaluate the model's performance.

Next, this set of data splitting options is very similar to how the project in [39] splits the dataset. The only significant difference is that the 5-fold CV is used instead of 10-fold CV. The significance of Table 4-1 is to have an overview of data splitting experiment results and how the dataset is split in this project. In comparison, other waste image classification projects either not state their data splitting methods clearly or not provide justifications. However, data splitting is one of the most important steps of training a classification model since models are developed based on data. Overall, the data splitting framework of this project can be further improved and used in the future recycling waste image classification projects.

4.1.1 Test set split ratio

Experiments of different test set split ratios, including 10%, 15%, 20%, and 25%, are conducted. From statistical analysis, 10% and 15% split ratios both give higher population mean accuracy than the other two split ratios. Among these two split ratios, 10% split ratio is selected since it gives the model more training data to learn.

Table 4-2. Descriptive statistics on test set accuracy of different test set split ratios



Test Set Split Ratio	Sample Size	Sample Mean	Standard Error
10%	30	82.71%	0.00333
15%	30	81.98%	0.00316
20%	30	81.83%	0.00342
25%	15	80.61%	0.00347

Table 4-2 shows descriptive statistics of the test set accuracy samples using different split ratios. These descriptive statistics include the total number of samples drawn (sample size), the average of accuracy samples (sample mean), and standard error of the sample mean (standard error). Although the sample size of 25% split is only half of the other three splits, the standard errors of these sample means are at the same level (slight above 0.003). Furthermore, the sample mean of test set accuracies increases as the test set split ratio decreases.

The sample mean accuracy goes up as the test set split ratio decreases because the model has more training and validation data. Unexpectedly, the sample size of 25% split is only half of the other splits to reach the same level of standard error. A possible explanation is that the sample mean accuracy converges much quicker to the population mean accuracy as more data is given to the test set. The significance of Table 4-2 is to show sample mean accuracy of different test set split ratios and to show the sample size selections are big enough to bring standard errors of different splits to the same level (slight above 0.003).

A. Sample Size and Standard Error

The standard error of the sample mean is used as an indicator to choose sample sizes for different test set splits. To be specific, the standard error reflects how far the sample mean deviates from the population mean. Stabilizing and reducing standard errors of these splits to the same level can help experiment results become statistically significant.

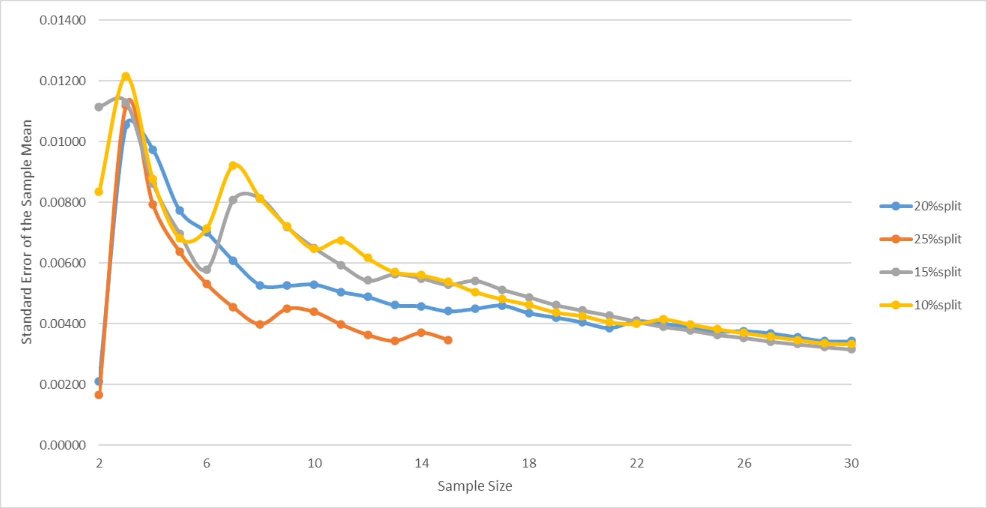


Figure 4-1. Standard error of the sample means against sample sizes

The standard error of all four splits, in Figure 4-1, shows a decreasing trend with increasing sample size. Firstly, all four split curves start (sample size=2) with low standard errors. Then, standard errors increase rapidly to the maximum at the next point (sample size=3). Furthermore, steep decreases can be observed in the next few increases of sample size in all splits. To be specific, 64.3% and 50% standard error decreases can be observed in 20% split and 50% split respectively when the sample size is increased from 3 to 8. On the other hand, steep standard error decreases can only be observed before sample size of 5 in 10% split and sample size of 6 in 15% split. After that, standard errors in all four splits decrease slowly and become stable.

To begin with, standard error of the sample mean is not defined at sample size=1 because of Bessel's correction [46], where sample standard deviation's denominator is corrected from the sample size to sample size -1. This makes the sample standard deviation become valid only with sample size bigger than one. Furthermore, the authors in [73] pointed out that the standard error from a small sample size is likely to underestimate the standard error of the population, and this explains the low standard errors when the sample size is 2.

Next, the standard error gradually decreases as more samples are collected, and it converges to zero when the sample mean approximates to the population mean. The significance of Figure 1 is to identify the relationship between the standard error of the sample mean and sample size, and to help choose the correct sample size. More specifically, a correct sample size selection should not be too small that the standard error is large, also not too large that the reduction of standard error becomes too small. These experimental results suggest that the standard error of the sample mean is a good indicator of selecting the sample size.

B. *Sample mean*

Statistical analysis, comparison of means, is performed on the sample mean accuracy to compare the population mean accuracy. ANOVA, a parametric test, is often used to compare means between multiple groups. Furthermore, the Shapiro-Wilk test and Levene's test are used to testing ANOVA's two assumptions, normal distribution and homogeneity of variances, respectively. Full statistical analysis result of the test set split ratio is provided in appendix A-1.

- The p-value of Shapiro-Wilk test (confidence level of 0.95) is 0.53, 0.70, 0.44, and 0.99 in 10%, 15%, 20%, and 25% split, respectively.
- The p-value of Levene's test (confidence level of 0.95) is 0.27.
- The p-value of ANOVA test (confidence level of 0.95) is 0.0036.
- The p-value of pairwise one-tailed t-test (confidence level of 0.95) is 0.059 between 10% and 15%, 0.037 between 10% and 20%, and <0.001 between 10% and 25%.

The Shapiro-Wilk tests' results suggest that these four splits data are normally distributed since their p-values are much larger than 0.05. Next, the p-value of Levene's test is larger than 0.05, which proves that there is not a significant difference in variance between these four splits. With two assumption hold, the ANOVA test is valid. The ANOVA test's result suggests that at least one group's mean is different from others since the p-value is smaller than 0.05. Additionally, one-tailed t-tests are performed pairwise between the highest sample mean group 10% and other split groups to identify mean differences. There is not a significant difference between 10% and 15% split since their t-test p-value is larger than 0.05. On the contrary, 10% split has larger mean compared with 20% split and 25% split since their p-values are smaller than 0.05.

Statistical analysis is useful to determine which data split ratio gives the highest classification mean accuracy. In comparison, the sample mean accuracy can be biased and misleading since the sample size is not large enough. The ANOVA test takes the sample size into account and compares mean differences, which conduct a more accurate analysis. Unexpectedly, although there is a 0.73% sample mean difference, from Table 4-2, between 10% and 15%, the ANOVA test does not consider this difference is significant. A possible explanation is that the confidence level of the ANOVA test is small. As a result, 10% and 15% split can be considered to produce the same level of classification accuracy.

In this project, the split ratio of the test set is selected by comparing mean classification accuracy using statistical tests, which makes the experiment result has statistical significance. This approach of selecting the test set split ratio can also be applied to other recycling waste classification projects to split the dataset and improve classification accuracy.

4.1.2 Training and validation set split ratio

The training and validation set split ratio is determined using the K-fold CV. To be specific, the remaining dataset (90%), excluding the test set, is further split into training and validation set. This split ratio cannot be selected freely since K-fold CV splits the remaining dataset into K folds, and the validation set size is the size of one fold. For example, if a 10-fold CV is applied to the remaining dataset, only 9% of the entire dataset is split as the validation set. Therefore, K selections in the K-fold CV is important to generate a well-performed classification model.

In this project, experiments of two most used K values, 5 and 10, are carried out, and the one gives higher validation set accuracy is selected. Fifty samples are collected, and statistical analysis is applied to them to compare different K values. From statistical analysis, K=10 gives higher classification mean accuracy, and 10-fold CV is applied to split the training and validation set in this project.

Table 4-3. Descriptive statistics on the validation set accuracy of different K selections

	Sample Size	Sample Mean	Standard Error
K=5	50	84.14%	0.0022
K=10	50	84.97%	0.0028

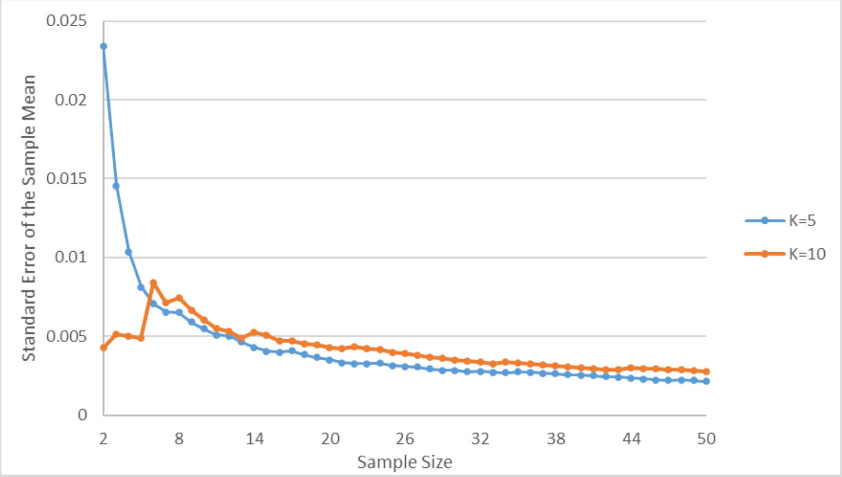


Figure 4-2. Standard error of the sample means against sample sizes of different K selections

Table 4-3 and Figure 4-2 are put together to analyse and discuss. From Table 4-3, K=10 gives slightly higher (0.85%) sample mean accuracy than K=5. Also, there is a 21.4% difference in standard errors between two K selections. On the other hand, Figure 4-2 shows that the standard error in both K selections have reached the threshold point, where the reduction rate of the standard error is rather low. In the K=5 curve, a percentage decrease of 91% is conducted between sample size=2 and sample size=15. In the K=10 curve, most reduction (62.6% decrease) happens between sample size=6 and sample size=13.

10-fold CV gives higher validation accuracy than 5-fold CV because the model has more training data to learn. The difference between standard error is acceptable since the K number is not selected based on the comparison of sample means. From Figure 4-2, the sample size is large enough to be used to perform the comparison of means test since the most reduction in standard error is conducted. A major limitation of this experiment is that the sample size is selected too large. As can be seen from Figure 4-2, only little reduction (6.3% decrease in K=5 and 26.8% decrease in K=10) of the standard error is conducted as the sample increases beyond 20. Therefore, too much computational resources are spent to collect K-fold CV comparison samples. The significance of Table 4-3 is to have an overview of sample statistics of different K selections. On the other hand, Figure 4-2 helps identify a limitation in the experiment set up, which is the sample size is not carefully selected for both K selections.

Statistical comparison of means is again applied to the k=5 and k=10 data split, with normality and homogeneity of variances test performed before the one-tailed t-test. The analysis and discussion of this experiment are general because the statistical analysis detail has been discussed in section 4.1.1, and the approach is very similar. Full statistical analysis result of the training and validation set split ratio is provided in appendix A-2.

- The p-value of Shapiro-Wilk test (confidence level of 0.95) is 0.169 in k=5 data split, and 0.403 in k=10 data split.
- The p-value of Levene's test (confidence level of 0.95) is 0.36.
- The p-value of pairwise one-tailed t-test (confidence level of 0.95) is 0.017.

The Shapiro-Wilk and Levene's test results suggest that normality and homogeneity of variances assumption are still held for K=5 and K=10 data split since these p values are larger than 0.05. Furthermore, the one-tailed t-test is performed to compare means between these two K selections, and the result (<0.05) suggests that there is a significant difference between these two groups' means. As a result, 10-fold CV is preferred in this project since it produces a higher classification mean accuracy experiment result.

4.1.3 Dataset sampling strategies

Sampling strategies of the dataset is determined by testing two sampling strategies, simple random sampling and stratified random sampling, and the one gives higher classification result is selected. The experiment conducted to test different sampling strategies is very similar to the split ratio experiment, as well as the results analysis part. Therefore, only key differences between them are presented, and full statistical analysis results of dataset sampling strategies are provided in appendix A-3.

The most significant difference in statistical analysis is that the homogeneity of variances assumption is violated in the test set sampling strategy experiment. Therefore, the unequal variances one-tailed t-test is applied to compare two group means. The p-values of t-test results in both the training and validation set and the test set are larger than 0.05, which suggests there is not a significant difference between the mean classification accuracies produced by two these two different sampling strategies. This experiment result can be explained by the balance subset size, where each subset of the TrashNet dataset contains around 500 images. Therefore, stratified sampling strategy does not conduct a higher classification accuracy in this project, but it can be useful when the dataset is highly imbalanced in terms of subset size. As a result, simple random sampling strategy is applied to the dataset because it consumes less computational resources compared with applying the stratified sampling strategy.

4.2 Training parameter

Various training parameters are tuned based on the 10-fold CV's average validation set classification accuracy. These training parameters are learning rate schedulers, optimisers, patience epochs, layers freezing, loss function, final classifiers, batch sizes, and learning rates. The research question answered in this section is how to tune these training parameters to achieve higher classification accuracy. An overview of experimental results is given in section 4.2.1, experimental results of layers freezing, learning rates, and stopping criterion are analysed and discussed in detail in section 4.2.2, 4.2.3, and 4.2.4, respectively.

Table 4-4. Training parameters results

Learning rate scheduler	Constant learning rate scheduler
Optimizer	Adam optimizer
Stopping criteria (patience epochs)	100 epochs
Layers freezing	Only freeze base convolutional layers
Loss functions + Final classifiers	Cross Entropy + Softmax
Batch size	16
Learning rate	2.00E-05

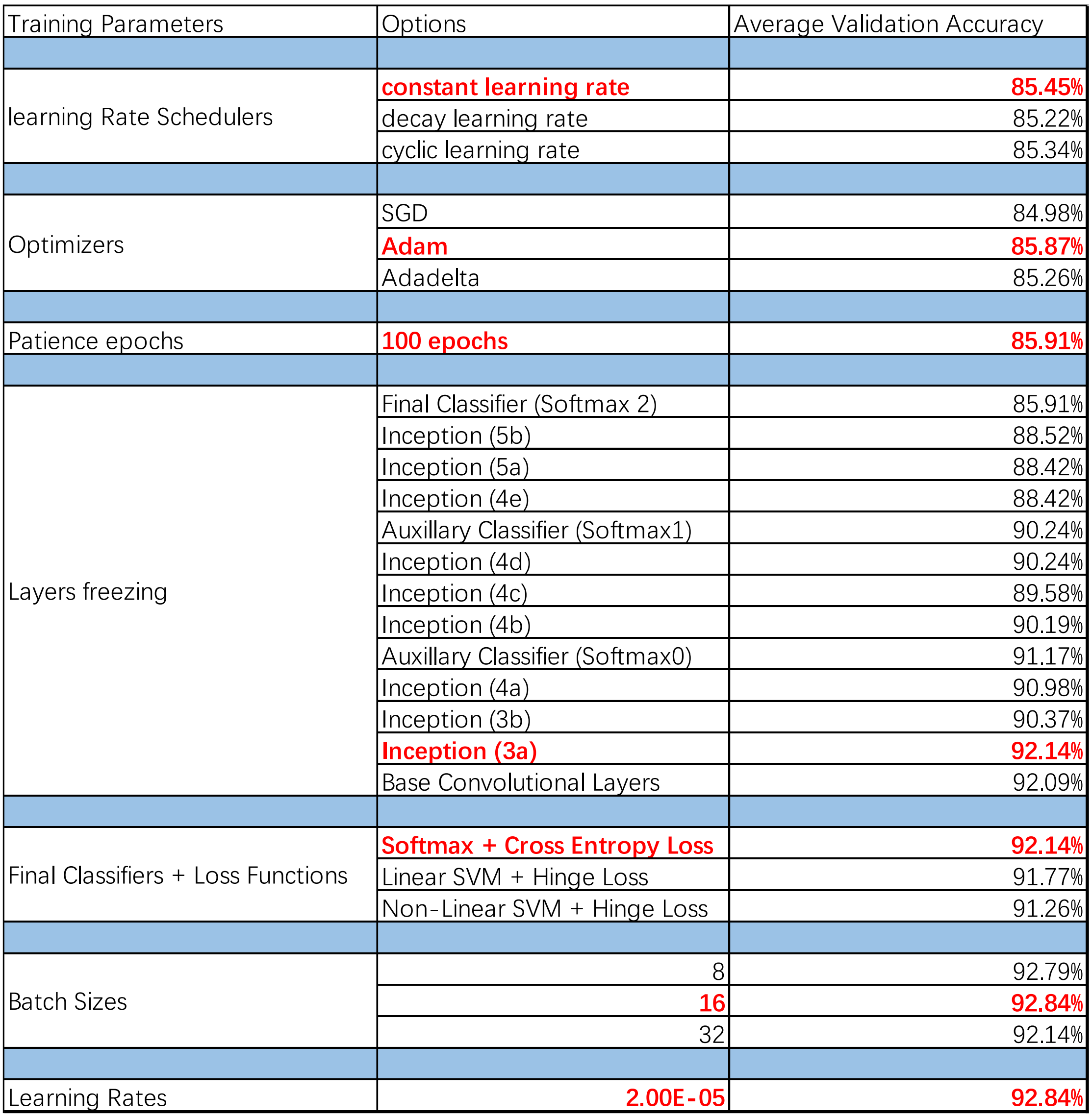
From Table 4-4, all base convolutional layers of the GoogleNet model are unfrozen. Accordingly, a small learning and a small batch size are selected to protect the low-level features extracted from the pre-trained weights. Classic CNN training parameters, such as constant learning rate scheduler, Adam optimizer, Cross-Entropy, and Softmax final classification layer, are used to train the recycling waste image classification model in this project.

Next, these training parameters chosen are similar to the parameters used in project [39]. To be specific, a small learning rate (2e-4), a small batch size (16), and the early stopping (patience epochs=25) technique are used in project [39]. On the other hand, the most significant difference is that SGD optimizer is used instead of the Adam optimizer in project [39]. The significance of Table 4-4 is to give an overview of the tuned training parameters which are used to train the CNN model. In comparison, some of the other waste classification projects do not provide information about training parameters, and most of these projects do not justify their selections. In this project, training parameters are tuned using the 10-fold CV, and the one gives the highest average validation set accuracy is selected.

4.2.1 Experiment results overview

The 10-fold CV technique helps split the dataset into ten different splits, and ten models are trained using these splits. Various training parameters are tested, and the validation set accuracies of ten models is averaged and compared.

Table 4-5. Average validation accuracy of 10-fold CV of different training parameters options



Training Parameters	Options	Average Validation Accuracy
learning Rate Schedulers	**constant learning rate**	**85.45%**
	decay learning rate	85.22%
	cyclic learning rate	85.34%
Optimizers	SGD	84.98%
	Adam	**85.87%**
	Adadelta	85.26%
Patience epochs	**100 epochs**	**85.91%**
Layers freezing	Final Classifier (Softmax 2)	85.91%
	Inception (5b)	88.52%
	Inception (5a)	88.42%
	Inception (4e)	88.42%
	Auxillary Classifier (Softmax1)	90.24%
	Inception (4d)	90.24%
	Inception (4c)	89.58%
	Inception (4b)	90.19%
	Auxillary Classifier (Softmax0)	91.17%
	Inception (4a)	90.98%
	Inception (3b)	90.37%
	Inception (3a)	**92.14%**
	Base Convolutional Layers	92.09%
Final Classifiers + Loss Functions	**Softmax + Cross Entropy Loss**	**92.14%**
	Linear SVM + Hinge Loss	91.77%
	Non-Linear SVM + Hinge Loss	91.26%
Batch Sizes	8	92.79%
	16	**92.84%**
	32	92.14%
Learning Rates	**2.00E-05**	**92.84%**

Table 4-5 shows the experiment results of various training parameters, where the selected training parameter and its accuracy are Highlighted using red colour. As can be seen from the column "Average Validation Accuracy", an increasing trend of accuracy can be observed from top to the bottom because these experiments are carried in the exact order, as shown in Table 4-5, and tuned training-parameters from previous experiments are applied to the model.

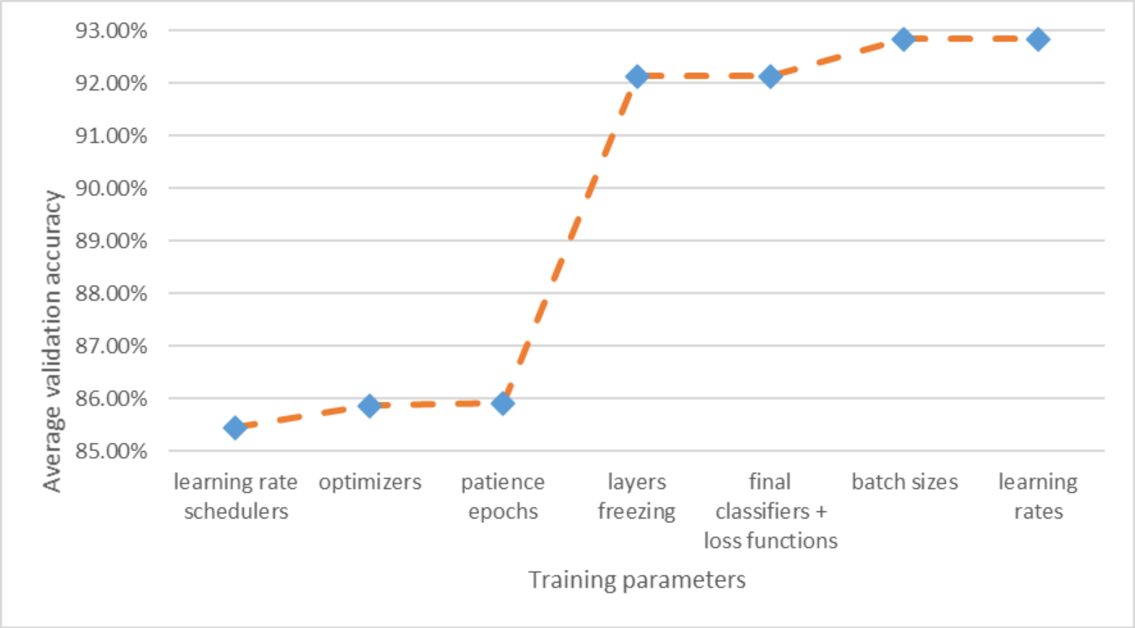


Figure 4-3. The increase of validation accuracies of different training parameters

The average validation accuracy increases are plotted as a curve in Figure 4-3 to identify the effects of different training parameters. From Figure 4-3, the most increase of the average validation accuracy, 7.25% increase, is conducted by unfreezing layers. The second most increase, 0.76% increase, is achieved by tuning the batch size. In comparison, there is a big gap between the improvements achieved by the first and second most effective training parameters. On the other hand, tuning of final classifiers, loss functions, and learning rates do not have any increase on the curve.

The core functions of transfer learning, using pre-trained weights to apply knowledge of another task to the target task, can explain this huge increase conducted by tuning the layers freezing. In this case, most layers can be retrained to improve classification accuracy, which suggests a great similarity can be found between the source domain (ImageNet dataset) and the target domain (TrashNet dataset). On the other hand, If the difference between these two domains is huge, then unfreezing all layers can conduct a negative impact on the model performance since low-level features extracted are destroyed.

Next, the benefits of tuning the final classifiers, loss function, and learning rates are not shown up on the curve. This is because the Softmax classifier and Cross-Entropy loss are used as the default selection, and the other classifiers or losses do not conduct the better performance. On the other hand, learning rates are selected using the learning rate finder throughout experiments. As a result, no other learning rates can be compared with to show up the benefits of tuning learning rates.

The effects of unfreezing layers can also be witnessed in reviewed waste classification projects. Among seven waste classification projects, six of them apply transfer learning techniques. Moreover, four of them initialise the model using pre-trained weights, and their average classification accuracy is 95.96%. However, these projects do not provide information about layers freezing, such as the number of frozen layers and position of frozen layers in the model. In this project, experiments of layers freezing are conducted, results are analysed and discussed, and the final decision on layers freezing is presented.

4.2.2 Layers freezing

The GoogleNet model, introduced in the literature review section, is selected as the source model of transfer learning. The only change made to the GoogleNet model structure is the final dense layer. To be specific, the number of nodes in the final dense layer is reduced from 1000 to 5 since this MSc project is to classify five types of recycling waste images. Next, the layers freezing experiment is carried out by freezing all layers of the transferred GoogleNet model at first. Layers are gradually unfrozen from the output dense layer to the input convolution layer, and the one gives higher classification accuracy is selected.

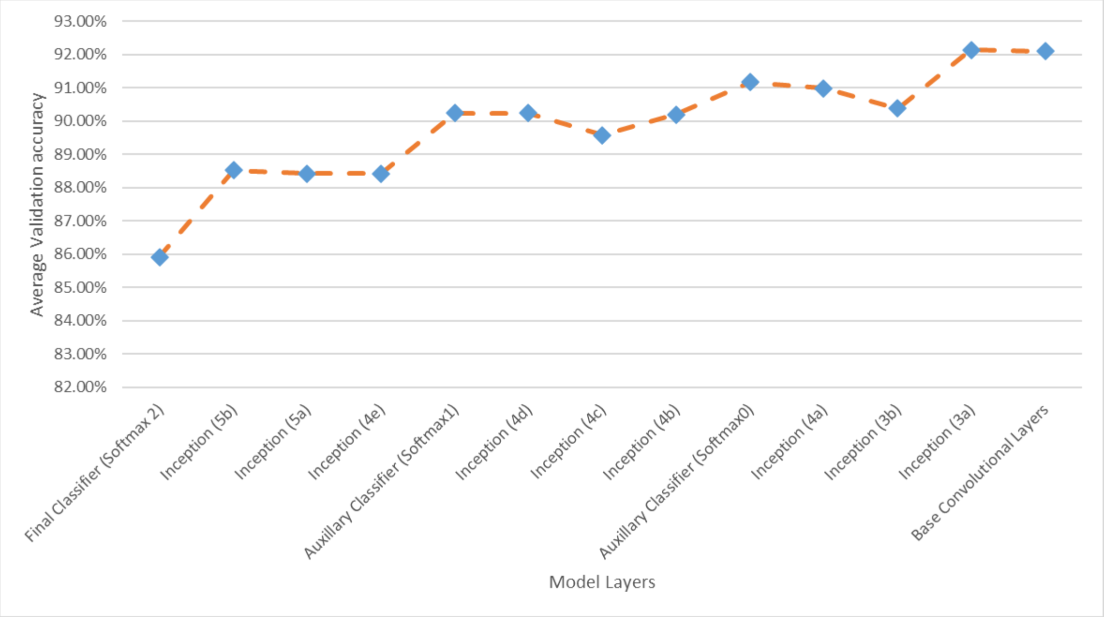


Figure 4-4. Validation accuracy of different layers freezing state

From Figure 4-4, an increasing trend in average validation accuracy can be observed as the layers are gradually unfrozen. There are five steep increases, which happens at the unfreezing of Inception module (5b, 4b, 3b) and Auxiliary classifier (Softmax1, Softmax0). Among these improvements, the unfreezing of Inception module 5b increases the accuracy the most, 2.61% in accuracy. The second most increase happens at the unfreezing of Auxiliary classifier softmax1, 1.82% in accuracy. On the other hand, two decreases of validation accuracy, about 0.6% in accuracy, can be observed when unfreeze Inception module (4c, 3b).

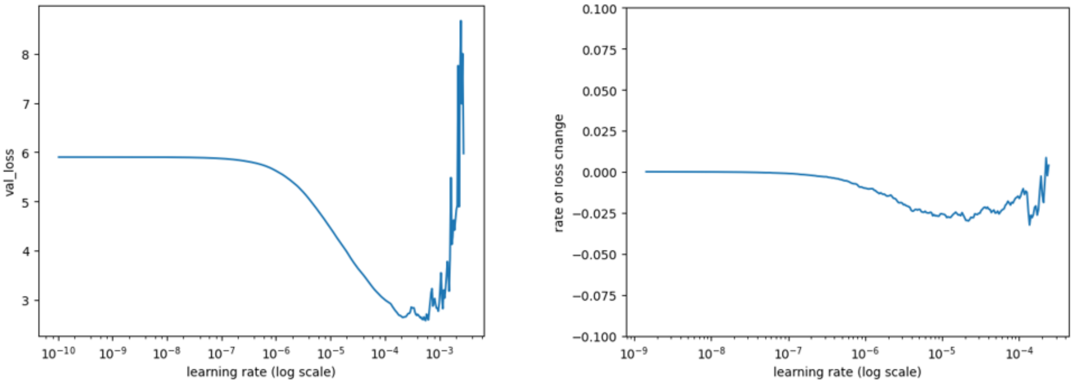
In general, CNN models can be considered as a feature extraction model if ignoring the final classification layer. Unfreezing of layer training is the same as retraining the feature extraction algorithm to extract different levels of features. To be specific, features extracted by the layers close to the final classification layer are high-level features, and low-level features if extracted by the layers close to the input convolutional layers. That being said, the first steep increase can be explained by the benefits of retraining a high-level feature extraction algorithm towards recycling waste image classification model.

Also, the accuracy is improved when two auxiliary classifiers are unfrozen because the model can better combat the vanishing gradient problem as introduced in the literature review section. Although Inception module 3a is very close to the input convolutional layers, the validation accuracy still increases as the layer is unfrozen. A possible explanation is that the low-level features of the ImageNet dataset are very similar to the TrashNet dataset. On the other hand, the small accuracy reduction of base convolutional layers can be explained by the disruption of low-level feature extraction algorithm.

Overall, the layers freezing experiment helps protect the low-level feature extraction algorithm and tune the high-level feature extraction algorithm towards the recycling waste classification task. In comparison, the other waste image classification projects do not provide information about layers freezing or related experimental results. For future waste image classification projects, similar layers freezing experiments are necessary when applying transfer learning techniques.

4.2.3 Learning rates

Learning rates are determined using the learning rate finder introduced in the literature review section. To be specific, the validation loss at different learning rate is obtained and plotted on a graph. Then, the most reduction range is zoomed out and differentiated with respect to iterations, and the lowest rate of loss change learning rate is used to train the model.



(a) (b)

Figure 4-5. plots of, (a): validation loss against learning rate, (b): rate of loss change against learning rate

The learning rate is dependent on various model parameters, such as batch sizes, model structures, and optimizers. Therefore, the learning rate experiment is carried out every time before the conduction of other experiments to achieve accurate experiment results. Figure 4-5 shows how the learning rate of the benchmark model is identified, where both validation losses and the rate of change of validation losses are plotted against learning rates. From Figure 4-5 (a), the most reduction of learning rate happens between learning rates of 1e-7 and 2e-4. Next, Figure 4-5 (b) further helps identify the learning rate of the lowest rate of change of loss, around 2e-5.

The selected learning rate, 2e-5, is very small because the pre-trained weights of the GoogleNet model should be fine-tuned to adapt to the recycling waste image classification task. A large learning rate choice can destroy features and knowledge learnt from the pre-trained weights. Five waste classification projects provide their learning rate choices, and the most used learning rate, 1e-3, is also a small learning rate. Unlike this MSc project, these reviewed projects do not provide justification for their learning rate selections. The learning rate finder is a useful technique, which can be applied to almost all neural network training process.

4.2.4 Stopping criterion

The patience epoch experiment is carried out by selecting a large patience epoch number first, and the largest epoch gap is identified. To be specific, the largest epoch gap is 87 when the patience epoch number is 200. To discover all potential validation accuracy increase, a larger patience epoch (100) is selected to train the benchmark model.

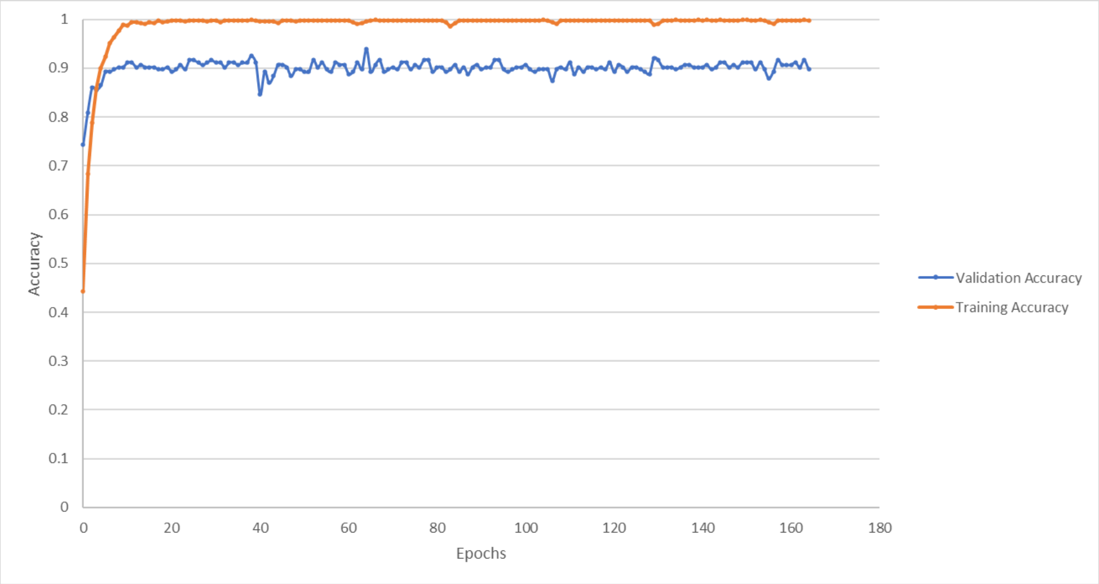


Figure 4-6. Validation and Training accuracy against epochs

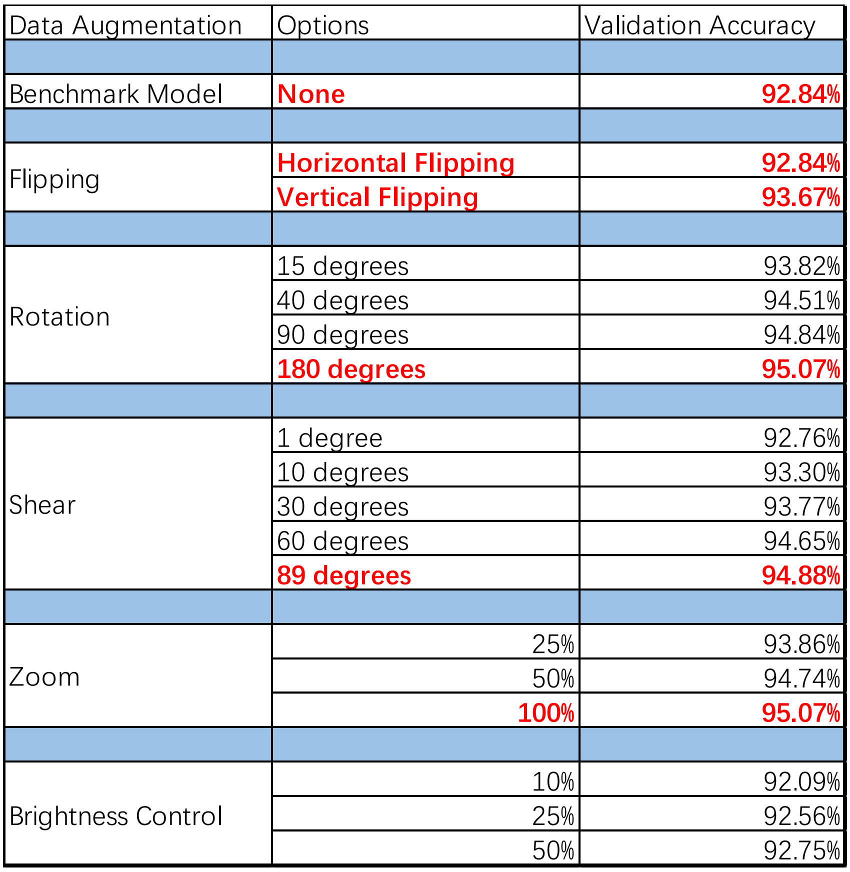
From Figure 4-6, the training accuracy increases to around 100% accuracy after 20 epochs. On the other hand, the validation accuracy increases above 90% after ten epochs, but fluctuations can be observed between 40 epochs and 60 epochs. The major limitation of this experiment is that the patience epoch number is selected too large, and small increases in validation accuracy do not have a huge impact on improving model performances. Therefore, the patience epoch experiment should be redesigned and combined with the validation accuracy plot. More specifically, most validation improvement should be obtained but without too much computational resources wasted on subtle validation accuracy improvements. In this case, a reasonable patience epoch section should be around 20 since most validation accuracy improvements happen within 20 epochs.

4.3 Data augmentation

After developing the benchmark model, various data augmentation techniques can be tested and determine which set of techniques to use. These data augmentation techniques are flipping, rotation, shear, zoom, and brightness control. The research question answered in this question is how to select and apply data augmentation techniques to improve the quality of a dataset. In this section, data augmentation techniques used in this project are selected based on validation accuracy, and confusion matrices of different data augmentation techniques are plotted and analysed.

4.3.1 Experiment results overview

Table 4-6. Validation accuracy of different data augmentation techniques



Data Augmentation	Options	Validation Accuracy
Benchmark Model	**None**	**92.84%**
Flipping	**Horizontal Flipping**	**92.84%**
	Vertical Flipping	**93.67%**
Rotation	15 degrees	93.82%
	40 degrees	94.51%
	90 degrees	94.84%
	180 degrees	**95.07%**
Shear	1 degree	92.76%
	10 degrees	93.30%
	30 degrees	93.77%
	60 degrees	94.65%
	89 degrees	**94.88%**
Zoom	25%	93.86%
	50%	94.74%
	100%	**95.07%**
Brightness Control	10%	92.09%
	25%	92.56%
	50%	92.75%

From Table 4-6, the application of four data augmentation techniques, including flipping, rotation, shear, and zoom, improve validation accuracy. Among these four techniques, the most increase, 2.4% increase, can be observed in both zoom and rotation techniques. On the other hand, all the brightness control options tested conduct a decrease in validation accuracy.

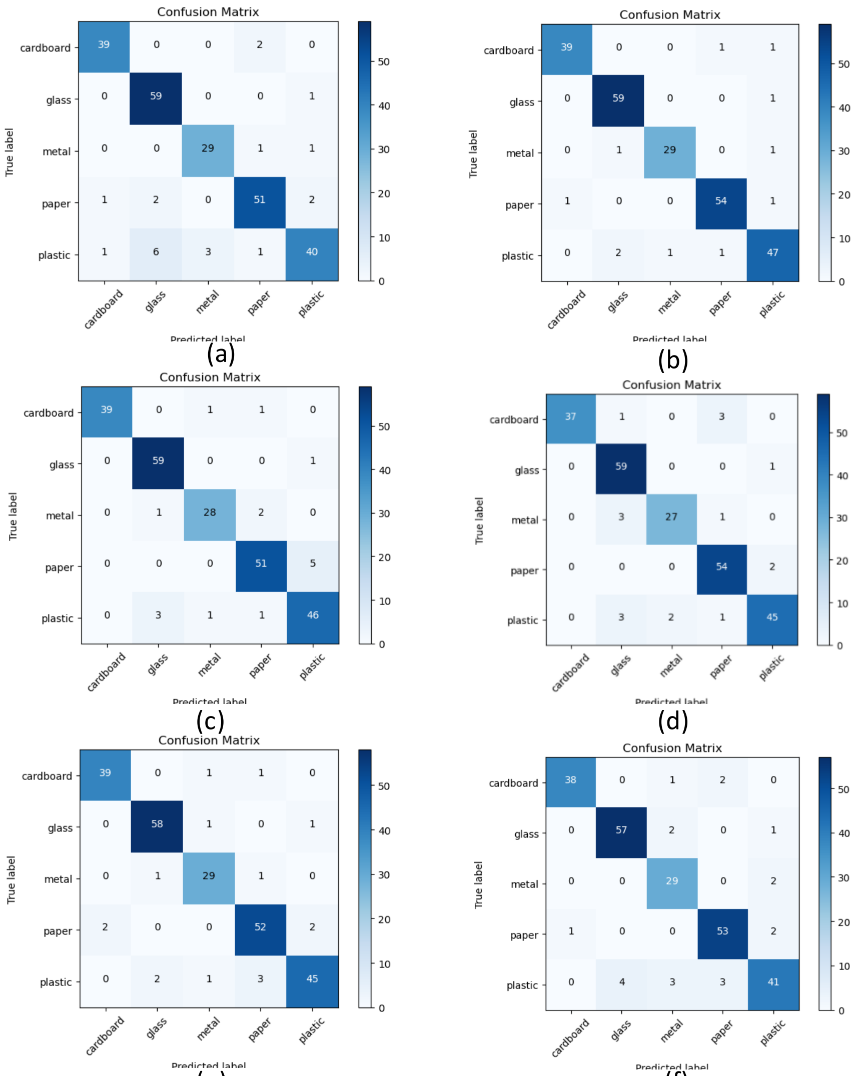
The validation increase is the result of improvement in both size and diversity of datasets through applying data augmentation techniques. To be specific, images are applied with a set of random data augmentation techniques before passing to the learning algorithm, for example, a combination of horizontal flipping, clockwise rotation of 36 degrees, zoom in 25%, and shear of 30 degrees. This process is carried out randomly before each iteration. As a result, the learning algorithm can access to a huge number of augmented training images. Also, the effects of the overfitting problem can be mitigated through data augmentation techniques.

Unexpectedly, applying the brightness control technique conducts a small percentage decrease in validation accuracy, and this may be explained by the brightness uniformity of the TrashNet dataset. To be specific, the TrashNet images are taken under similar lighting condition, sunlight or room light. Moreover, data augmentation techniques are only applied to the training set to test their effects, and this causes a significant brightness difference between training set and validation set images. As a result, the validation set accuracy is worsened when brightness control techniques are applied.

Overall, the data augmentation experiment conducted in this project help selects suitable techniques to be performed on the TrashNet dataset, and these techniques are used to provide more training samples to the classification model. In comparison. the other waste classification projects simply state their data augmentation techniques without justifications. For any image related classification tasks, the data augmentation experiment is useful since it helps choose the most suitable techniques for datasets.

4.3.2 Confusion matrix

Confusion matrix of the benchmark model, final model, and the benchmark model with different data augmentation techniques applied (including flipping, zoom, rotation, and shear) are plotted. These matrices visually present the classification results of different classes, where correct predictions are plotted on the diagonal line, and wrong predictions are plotted on the rest blocks of these matrices.



(a) (b) (c) (d) (e) (f)

Figure 4-7. Confusion matrix of different models

From Figure 4-7 (a), the benchmark model has the highest error rate, 21.57%, in plastic image predictions, and the second-highest error rate, 8.93%, in paper image predictions. Figure 4-7 (b) shows the final model's performance, which is applying selected data augmentation techniques to the benchmark model. To be specific, the error rate of the plastic class is reduced from 21.57% to 7.84%, and the paper class's error rate is reduced from 8.93% to 3.57%. Also, the classification accuracy of other classes remains the same.

Next, the confusion matrix between the benchmark model and different data augmentation techniques applied are compared. From Figure 4-7 (c), the benchmark model applied with random flipping has the same paper class error rate (8.93%), but lower plastic class error rate (9.80%). However, all five misclassified paper images are identified as plastic. From Figure 4-7 (d), the benchmark applied with random zooming has lower plastic class error rate, 11.76%, and lower paper class error rate, 3.57%. However, the metal class error rate is increased from 6.45% to 12.9%.

On the other hand, the benchmark model applied with random rotation, shown in Figure 4-7 (e), has lower plastic class error rate, 11.76%, and a slight lower paper class error rate, 7.14%, and there is not a significant change on the other classes' error rates. From Figure 4-7 (f), the benchmark model applied with random shearing has a slight lower plastic class error rate 19.6%, and lower paper class error rate, 5.36%.

Plastic images are often classified as metal and glass by the benchmark model, and one possible explanation is that these three are all reflective materials. Flipping, zooming, and rotation techniques help the model distinguish these three waste materials by discovering new features. For example, more weights can be assigned by the model to discover shape differences between plastic bottle and metal can by learning rotated images. As a result, the final model recognizes paper and plastic images by discovering new classification features from augmented images.

4.3.3 Test set accuracy

Lastly, the final model and benchmark model are evaluated on the test set. The effects of data augmentation techniques can be quantified using the test set accuracy improvement, also the final model's performance.

	Test Accuracy
Benchmark Model	91.21%
Final Model	95.40%

Table 4-7. Test set accuracy of benchmark model and final model

From Table 4-7, the final model achieves a satisfied classification result on the test set, and the data augmentation techniques lead to a 4.19% increase in accuracy. From the literature review, only two waste classification projects [10], [70] achieve higher accuracy. Overall, this classification model can classify most recycling waste images correctly with the help of transfer learning techniques, tuned training parameters, and data augmentation techniques.

5. Conclusion

This research aims to develop an approach to train recycling waste image classification models. To develop this approach, three research questions need to be answered, including how to split the dataset, how to select training parameters, and how to choose data augmentation techniques.

The dataset is split by choosing three parameters, including split ratios, sampling strategies, and CV techniques, which are determined through statistical tests of test set mean accuracy. In this project, 10% of the entire dataset is split using the Hold-out CV technique with simple random sampling strategy, and the rest dataset is then split into training and validation sets using the 10-fold CV technique with simple random sampling strategy.

Next, the training parameters are selected by comparing the average 10-fold validation accuracy, and these parameters are learning rate schedulers, patience epochs, layers freezing, loss function, final classifiers, batch sizes, and learning rates. In this project, the constant learning rate scheduler of 2e-5 learning rate is used together with Adam optimizer to optimise the Cross-Entropy loss. Next, the node number in the final classification layer, Softmax, of the GoogleNet model is reduced from 1000 to 5, and only base convolutional layers are frozen. Then, the pre-trained weights of the GoogleNet model are used to initialise the transferred GoogleNet model, and images are passed in a batch size of 16. Lastly, the training process is stopped if no higher validation accuracy is obtained in the next 100 epochs. The benchmark model is trained using the above-described training parameters and obtained 91.21% test set classification accuracy.

Various data augmentation techniques are tested using the average 10-fold validation accuracy, and these augmentation techniques are flipping, rotation, shear, zoom, and brightness control. In this project, all techniques except brightness control are applied to the benchmark model, and the test set classification accuracy is increased to 95.40%. Lastly, this model has been used to perform real-time classification on a computer using webcam and achieved adequate classification result.

Future research can focus on moving the classification process to the edge through embedded devices, and this requires the knowledge of computational resource and model complexity. Furthermore, bounding boxes can be added to the classification model as an additional feature, as it will help distinguish between foreign objects and waste objects, and further improve classification accuracy.

In conclusion, this project develops an approach of training recycling waste image classification models and achieves a great success in terms of classification accuracy, 95.40%. Also, the model can perform real-time recycling waste classification using the webcam on the computer.

References

[1] S. H. Schneider, “Municipal solid waste,” *Energy Conversion, Second Ed.*, pp. 73–79, 2017.

[2] S. Kaza, L. Yao, P. Bhada-Tata, and F. Van Woerden, *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050*. The World Bank, 2018.

[3] M. A. Hannan, M. Abdulla Al Mamun, A. Hussain, H. Basri, and R. A. Begum, “A review on technologies and their usage in solid waste monitoring and management systems: Issues and challenges,” *Waste Manag.*, 2015.

[4] T. Parker, “IBM testing AI solution to improve recycling at Marwell Zoo.” https://www.nspackaging.com/news/ibm-recycling-marwell-zoo-wastenet/ (accessed May 12, 2020).

[5] T. Parker, “Don’t know which waste goes where? IBM may have found a solution.” https://www.ns-businesshub.com/science/ibm-waste-solution/ (accessed May 12, 2020).

[6] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, Dec. 2015.

[7] “garythung/trashnet: Dataset of images of trash; Torch-based CNN for garbage image classification.” https://github.com/garythung/trashnet (accessed May 13, 2020).

[8] G. Thung and M. Yang, “Classification of Trash for Recyclability Status,” 2017.

[9] C. Bircanoglu, M. Atay, F. Beser, O. Genc, and M. A. Kizrak, “RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks,” *2018 IEEE Int. Conf. Innov. Intell. Syst. Appl. INISTA 2018*, 2018.

[10] U. Özkaya and L. Seyfi, “Fine-Tuning Models Comparisons on Garbage Classification for Recyclability,” *Proc. 2nd Int. Symp. Innov. Approaches Sci. Stud.*, 2018.

[11] Goverment UK, “Decide if a material is waste or not: general guide.” https://www.gov.uk/government/publications/legal-definition-of-waste-guidance/decide-if-a-material-is-waste-or-not#decide-if-your-material-is-waste (accessed Sep. 05, 2020).

[12] Goverment UK, “Classify different types of waste.” https://www.gov.uk/how-to-classify-different-types-of-waste (accessed Sep. 05, 2020).

[13] Statistics Explained, “Glossary:Recycling of waste.” https://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Recycling_of_waste (accessed Sep. 05, 2020).

[14] ISM Waste & Recycling, “Types of Recycling.” https://ismwaste.co.uk/recycling-services/types-of-recycling (accessed Sep. 05, 2020).

[15] WayBack Machine, “How is Paper Recycled?” https://web.archive.org/web/20111130061422/http://www.tappi.org/paperu/all_about_paper/earth_answers/EarthAnswers_Recycle.pdf (accessed Sep. 05, 2020).

[16] Conserve Energy Future, “Metal Recycling: How to Recycle Metal and its Importance.” https://www.conserve-energy-future.com/recyclingmetal.php (accessed Sep. 05, 2020).

[17] Manchester City Council, “See which recycling bin to use.” https://secure.manchester.gov.uk/info/200084/bins_rubbish_and_recycling/6026/see_which_recycling_bin_to_use (accessed Sep. 05, 2020).

[18] S. P. Gundupalli, S. Hait, and A. Thakur, “A review on automated sorting of source-separated municipal solid waste for recycling,” *Waste Manag.*, 2017, [Online]. Available: http://dx.doi.org/10.1016/j.wasman.2016.09.015.

[19] F. Daniea, “What is Machine Learning?” https://emerj.com/ai-glossary-terms/what-is-machine-learning/ (accessed Sep. 07, 2020).

[20] B. Jason, “Supervised and Unsupervised Machine Learning Algorithms.” https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (accessed May 17, 2020).

[21] K. Renu, “Convolutional Neural Network(CNN) Simplified.” https://medium.com/datadriveninvestor/convolutional-neural-network-cnn-simplified-ecafd4ee52c5 (accessed May 22, 2020).

[22] M. M. Candace, “Fully connected neural network.” https://radiopaedia.org/articles/fully-connected-neural-network (accessed Sep. 08, 2020).

[23] WayBack Machine, “The Machine Learning Dictionary.” https://web.archive.org/web/20180826151959/http://www.cse.unsw.edu.au/~billw/mldict.html#neuron (accessed Sep. 08, 2020).

[24] MissingLink.ai, “Convolutional Neural Network: How to Build One in Keras & PyTorch.” https://missinglink.ai/guides/neural-network-concepts/convolutional-neural-network-build-one-keras-pytorch/ (accessed Sep. 08, 2020).

[25] R. Abilash, “APPLYING RANDOM FOREST (CLASSIFICATION) — MACHINE LEARNING ALGORITHM FROM SCRATCH WITH REAL DATASETS.” https://medium.com/@ar.ingenious/applying-random-forest-classification-machine-learning-algorithm-from-scratch-with-real-24ff198a1c57 (accessed Sep. 08, 2020).

[26] Y. Andreopoulos, “Applied Machine Learning System.”

[27] Savyakhosla, “Using SVM to perform classification on a non-linear dataset.” https://www.geeksforgeeks.org/ml-using-svm-to-perform-classification-on-a-non-linear-dataset/ (accessed Sep. 09, 2020).

[28] S. Amarappa and S. V Sathyanarayana, “Data classification using Support vector Machine (SVM), a simplified approach.” Accessed: May 22, 2020. [Online]. Available: www.ijecse.org.

[29] R. Berwick, “An Idiot’s guide to Support vector machines (SVMs).”

[30] N. Avinash, “KNN Classification using Scikit-learn.” https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn (accessed Sep. 09, 2020).

[31] L. Torrey and J. Shavlik, “Transfer Learning.”

[32] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, 2010.

[33] B. Jason, “A Gentle Introduction to Transfer Learning for Deep Learning.” https://machinelearningmastery.com/transfer-learning-for-deep-learning/ (accessed Sep. 10, 2020).

[34] Stanford Vision Lab, “ImageNet.” http://www.image-net.org/about-stats (accessed Sep. 10, 2020).

[35] B. Jason, “A Gentle Introduction to the ImageNet Challenge (ILSVRC).” https://machinelearningmastery.com/introduction-to-the-imagenet-large-scale-visual-recognition-challenge-ilsvrc/ (accessed Sep. 10, 2020).

[36] C. Szegedy *et al.*, “Going Deeper with Convolutions.”

[37] B. Jason, “How to Fix the Vanishing Gradients Problem Using the ReLU.” https://machinelearningmastery.com/how-to-fix-vanishing-gradients-using-the-rectified-linear-activation-function/ (accessed Sep. 11, 2020).

[38] Cross Validated, “deep learning - Google Inception model:why there is multiple softmax?” https://stats.stackexchange.com/questions/274286/google-inception-modelwhy-there-is-multiple-softmax/274623 (accessed Sep. 11, 2020).

[39] V. Ruiz, Á. Sánchez, J. F. Vélez, and B. Raducanu, “Automatic Image-Based Waste Classification,” Jun. 2019.

[40] DewangNautiyal, “Underfitting and Overfitting in Machine Learning.” https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/ (accessed Sep. 13, 2020).

[41] B. Jason, “Overfitting and Underfitting With Machine Learning Algorithms.” https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/ (accessed Sep. 13, 2020).

[42] Z. Reitermanová, *Data Splitting*. .

[43] J. Chow, “Model Validation.” https://www.h2o.ai/community/glossary/model-validation-hold-out-cross-validation (accessed Sep. 13, 2020).

[44] QuestionPro, “Stratified Random Sampling: Definition, Method and Examples.” https://www.questionpro.com/blog/stratified-random-sampling/ (accessed Sep. 13, 2020).

[45] D. Roberts, “Variance and Standard Deviation.” https://mathbitsnotebook.com/Algebra1/StatisticsData/STSD.html (accessed Sep. 13, 2020).

[46] E. W. Weisstein, “Bessel’s Correction.”

[47] La Trobe University, “Hypothesis testing.” https://latrobe.libguides.com/maths/hypothesis-testing (accessed Sep. 14, 2020).

[48] B. Jason, “A Gentle Introduction to Statistical Hypothesis Testing.” https://machinelearningmastery.com/statistical-hypothesis-tests/ (accessed Sep. 14, 2020).

[49] P. A. Mackowiak, S. S. Wasserman, and M. . Levine, “Confidence Intervals.” http://www.stat.yale.edu/Courses/1997-98/101/confint.htm (accessed Sep. 14, 2020).

[50] B. BEERS, “P-Value Definition.” https://www.investopedia.com/terms/p/p-value.asp (accessed Sep. 14, 2020).

[51] Laerd Statsitics, “Testing for Normality using SPSS Statistics.” https://statistics.laerd.com/spss-tutorials/testing-for-normality-using-spss-statistics.php (accessed Sep. 14, 2020).

[52] SEMATECH e-Handbook of Statistical Methods, “Anderson-Darling and Shapiro-Wilk tests.” https://www.itl.nist.gov/div898/handbook/prc/section2/prc213.htm (accessed Sep. 14, 2020).

[53] C. Zaiontz, “Shapiro-Wilk Test.” http://www.real-statistics.com/tests-normality-and-symmetry/statistical-tests-normality-symmetry/shapiro-wilk-test/ (accessed Sep. 14, 2020).

[54] L. Statistics, “One-way ANOVA.” https://statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide-2.php (accessed Sep. 14, 2020).

[55] ML Glossary, “Loss Functions.” https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html (accessed Sep. 14, 2020).

[56] RekhaMolala, “The Ascent of Gradient Descent.” https://blog.clairvoyantsoft.com/the-ascent-of-gradient-descent-23356390836f (accessed Sep. 14, 2020).

[57] B. Jason, “Gradient Descent For Machine Learning.” https://machinelearningmastery.com/gradient-descent-for-machine-learning/ (accessed Sep. 14, 2020).

[58] A. V Srinivasan, “Stochastic Gradient Descent.” https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31 (accessed Sep. 16, 2020).

[59] S. RUDER, “An overview of gradient descent optimization algorithms.” https://ruder.io/optimizing-gradient-descent/ (accessed Sep. 17, 2020).

[60] V. Bushaev, “Adam — latest trends in deep learning optimization.” https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c (accessed Sep. 17, 2020).

[61] B. Jason, “Understand the Impact of Learning Rate on Neural Network Performance.” https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/ (accessed Sep. 17, 2020).

[62] P. Surmenok, “Estimating an Optimal Learning Rate For a Deep Neural Network.” https://towardsdatascience.com/estimating-optimal-learning-rate-for-a-deep-neural-network-ce32f2556ce0 (accessed Sep. 17, 2020).

[63] P. Surmenok, “keras_lr_finder.” https://github.com/surmenok/keras_lr_finder (accessed Sep. 17, 2020).

[64] L. N. Smith, “Cyclical Learning Rates for Training Neural Networks.” Accessed: Jul. 11, 2020. [Online]. Available: www.cs.toronto.edu/.

[65] B. Kenstler, “Cyclical Learning Rate.” https://github.com/bckenstler/CLR (accessed Sep. 18, 2020).

[66] Elite Data Science, “Overfitting in Machine Learning: What It Is and How to Prevent It.” https://elitedatascience.com/overfitting-in-machine-learning (accessed Sep. 18, 2020).

[67] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” Accessed: Jun. 14, 2020. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0.

[68] M. Satvilkar, “Image Based Trash Classification using Machine Learning Algorithms for Recyclability Status MSc Research Project Data Analytics.”

[69] A. H. Vo, L. Hoang Son, M. T. Vo, and T. Le, “A Novel Framework for Trash Classification Using Deep Transfer Learning,” *IEEE Access*, vol. 7, pp. 178631–178639, 2019.

[70] G. White, C. Cabrera, A. Palade, F. Li, and S. Clarke, “WasteNet: Waste Classification at the Edge for Smart Bins.”

[71] G. Mittal, K. B. Yagnik, M. Garg, and N. C. Krishnan, “SpotGarbage: Smartphone app to detect garbage using deep learning,” *UbiComp 2016 - Proc. 2016 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, 2016.

[72] A. Jain, R. Bansal, A. Kumar, and K. Singh, “A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year

students," *Int. J. Appl. Basic Med. Res.*, 2015, Accessed: Sep. 27, 2020. [Online]. Available: /pmc/articles/PMC4456887/?report=abstract.

[73] J. Gurland and R. C. Tripathi, "A Simple Approximation for Unbiased Estimation of the Standard Deviation," *Am. Stat.*, 1971.

Appendix A

A-1: Test set split ratio statistical analysis

Table 0-1. Shapiro-Wilk test for test set split ratios

	Statistic	df	Sig.
10split	0.969551	30	0.526975
15split	0.975593	30	0.700191
20split	0.966212	30	0.44137
25split	0.990648	15	0.999633

Table 0-2. Levene's test for test set split ratios

Tests of Homogeneity of Variances				
	Levene Statistic	df1	df2	Sig.
Based on Mean	1.326820292	3	101	0.27
Based on Median	1.084919417	3	101	0.36
Based on Median and with adjusted df	1.084919417	3	99.52	0.36
Based on trimmed mean	1.327635104	3	101	0.27

Table 0-3. ANOVA test for test set split ratios

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.0044	3	0.00148	4.81	0.0036
Within Groups	0.0311	101	0.00031		
Total	0.0355	104			

Table 0-4. One-tailed t test for test set split ratios

10 split and 15 split	P(T<=t) one-tail	0.059
	t Critical one-tail	1.672
10 split and 20 split	P(T<=t) one-tail	0.037
	t Critical one-tail	1.672
10 split and 25 split	P(T<=t) one-tail	0.000
	t Critical one-tail	1.681

A-2: Training and validation set split ratio statistical analysis

Table 0-5. Shapiro-Wilk test for training and validation set K-fold CV

	Statistic	df	Sig.
k=5	.967	50	.169
k=10	.976	50	.403

Table 0-6. Levene's test for training and validation set K-fold CV

Levene Statistic	df1	df2	Sig.
4.533	1	98	.036

Table 0-7. One tailed t-test for training and validation set K-fold CV

one-tailed t-test statistic	df1	sig.
-2.171	49	0.017

A-3: Sampling strategy statistical analysis

Table 0-8. Shapiro-Wilk test for the test set sampling strategy

	Statistic	df	Sig.
Simple RS	0.970	30.000	0.527
Stratified RS	0.944	30.000	0.116

Table 0-9. Levene's test for the test set sampling strategy

Levene Statistic	df1	df2	Sig.
9.515	1	58	.003

Table 0-10. Unequal variances assumed one-tailed t-test for the test set sampling strategy

one tailed t-test statistic	df	sig.
-0.803	50	0.213

Table 0-11. Shapiro-Wilk test for the training and validation set sampling strategy

	Statistic	df	Sig.
Simple RS	.971	50	.263
Stratified RS	.976	50	.403

Table 0-12. Levene's test for the training and validation set sampling strategy

Levene Statistic	df1	df2	Sig.
3.227	1	98	.075

Table 0-13. One-tailed t-test for the training and validation set sampling strategy

one-tailed t-test statistic	df	sig.
0.4161	49	0.340