WASTE CLASSIFICATION USING INSTANCE-BASED ALGORITHM

AMIERAH HANIS BINTI ROSMEN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS LAPORAN

JUDUL: [*WASTE CLASSIFICATION USING INSTANCE-BASED ALGORITHM*]

SESI PENGAJIAN: [2021 / 2022]

Saya: ____AMIERAH HANIS BINTI ROSMEN__________________

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Universiti Teknikal Malaysia Melaka dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan unituk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. * Sila tandakan (✓)

__________	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
__________	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)
__________	TIDAK TERHAD	

________________________	________________________
(TANDATANGAN PELAJAR)	(TANDATANGAN PENYELIA)
Alamat tetap: No. 31 Jalan Cempaka Hutan, Maktab Teknik PDRM Bakri, 84200, Muar, Johor	DR ABDUL SYUKOR MOHAMAD JAYA _______________
	Nama Penyelia
Tarikh: __________________________	Tarikh: __________________________

CATATAN: * Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa

WASTE CLASSIFICATION USING INSTANCE-BASED ALGORITHM

AMIERAH HANIS BINTI ROSMEN

This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science (Artificial Intelligence)] with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

WASTE CLASSIFICATION USING INSTANCE-BASED ALGORITHM

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : ______________________________________ Date : ________
(AMIERAH HANIS BINTI ROSMEN)

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of [Computer Science (Software Development)] with Honours.

SUPERVISOR : ______________________________________ Date : ________
([DR ABDUL SYUKOR MOHAMAD JAYA])

DECLARATION

DEDICATION

I dedicate this final year project to my supervisor Dr Abdul Syukor Mohamad Jaya for his guidance and support. Without his knowledge and insight on the subject, surely the project will not become what it is today. Also, I thank my family and friends who have supported me during the time I feel like giving up and felt lost and helped me in my studies during the semester.

ACKNOWLEDGEMENTS

First and foremost, I am grateful to my supervisor Dr Abdul Syukor Mohamad Jaya for providing me his tips and recommendations during the progression of this project. His motivation for us to try our best has made it possible for me to move forward with this project.

Next, I would like to thank my friends for giving their honest thoughts on my project and continued to help me in times of need. During my studies, it is not easy to become level-headed as too many assignments and projects could cause us to forget important dates and also the need to relay our feelings. I am thankful to my friends for not leaving me alone and always remind me of the deadline of assignments.

Lastly, I thank my family for their full support to me to finish this degree without regrets and always motivate me to strive for the best

ABSTRACT

The aim of this project is to develop instance-based machine learning models for the use of classifying waste images in garbage dumps or waste sorting centers. We will choose between two instance-based models through their performance evaluation by performing training and testing using the images of organic and recyclable waste and perform prediction on the test set. A Naïve Bayes Classifier will also be included to be compared with the instance-based classifiers. The performance evaluation vector for this project includes accuracy, precision, recall and f1-measure. At the end of the project, the comparison between the performance evaluation vectors will determine the best instance-based model for waste image classification.

ABSTRAK

Objektif projek ini adalah untuk mengembangkan model pembelajaran mesin berasaskan contoh untuk penggunaan pengkelasan gambar sampah di tempat pembuangan sampah atau pusat pengumpulan sampah. Kami akan memilih antara dua model berdasarkan contoh melalui penilaian prestasi mereka dengan melakukan latihan dan ujian menggunakan gambar sampah organik dan kitar semula dan melakukan ramalan pada set ujian. Pengklasifikasi Naïve Bayes juga termasuk dalam membandingkan penilainan prestasi dengan model berdasarkan contoh. Vektor penilaian prestasi untuk projek ini merangkumi akurasi, ketepatan, mengingat semula dan ukuran f1. Pada akhir projek, perbandingan antara vektor penilaian prestasi akan menentukan model berasaskan contoh terbaik untuk klasifikasi imej sisa.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

LIST OF ABBREVIATIONS

FYP	-	Final Year Project
SVM	-	Support Vector Machine
RBF	-	Radial Basis Function
KNN	-	K-Nearest Neighbors
NB	-	Naïve Bayes

LIST OF ATTACHMENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction

Waste can be defined as a by-product made from human activities that are of no use to the producer but has the same substance available in the useful product (Ebikapade and Jim, 2016). Waste comes in many forms and can be classified into three physical states, which are liquid, solid and gaseous states. Municipal solid waste (MSW) on the other hand, is usually thought of as solid wastes generated by commercial, industrial and domestic sectors of the community which are then handled by the municipality (Asad et al, 2019).

2.01 billion tons of municipal solid waste are being generated globally every year and about 33% of the total waste is not being managed properly (Kaza et al., 2018). The problem of the environment being polluted by waste has become a major crisis in all parts of the world with the ongoing process of industrialization and urbanization, mostly due to solid waste pollution (Min Tian et al., 2019). Especially during this pandemic, managing these wastes has become vital for the community as increased buying of non-perishable items increases due to concern of food shortage (Hari Bhakta et al., 2020).

To reduce the impact of this problem, artificial intelligence can be used to separate the never-ending waste growth correctly in order to maintain the balance at which waste is being managed due to the increasing population growth (Bernardo et al., 2018). Deep learning and instance-based algorithms are examples of artificial intelligence techniques that are commonly used when classifying and predicting data.

For this paper, we will be focusing on TWO (2) instance-based algorithms to classify the waste images - recyclable and perishable - which are KNN and SVM to observe which instance-based algorithm is more reliable to perform this task. Naïve

Bayes Classifier will also be included to have the instance-based algorithms compare their performances with.

1.2 Problem statement(s)

To ease the task of classifying waste correctly. Classifying waste requires a lot of time and money to be performed in order to ensure proper conduct of waste management. Since humans are usually engaged in work and daily activities to the point where they cannot properly manage their waste, it is up to the machines available in landfills or garbage dumps to recognize and classify the waste. This project can help to increase the efficiency of classifying waste images into organic or recyclable waste.

1.3 Objective

The objectives of this project are as follows :-

i. To further ease the task of distinguishing organic and recycled waste.

ii. To develop a machine learning model for waste classification.

iii. To make recommendations based on the accuracy of different types of machine learning models in terms of accuracy, precision, recall and f1-measure.

1.4 Scope

The scope of the project includes the result of the classification of waste using instance-based algorithms. The waste classification is limited to organic and recycled waste. The instance-based algorithms that will be used are k-Nearest Neighbor (KNN) and Support Vector Machines (SVM). These algorithms will be used and compared in order to choose the algorithms with the highest accuracy and precision.

1.5 Project Significance

This project will help increase the performance of machines in landfill and garbage dumps to recognize waste and sort them into the correct types.

1.6 Expected Output

This project will analyze the performance of each learning model by ranking the accuracy, recall, f1-measure and precision from its classification of organic and recycled waste.

1.7 Conclusion

In a nutshell, this project will perform image processing on a waste image dataset and perform waste classification using TWO (2) instance-based machine learning models. Two SVM models of different kernels, namely polynomial and RBF is used, along with KNN and NB models. The models are then evaluated to choose which are most reliable in providing the most reliable performance in order to be of use to the waste management sector.

CHAPTER 2. LITERATURE REVIEW

2.1 Introduction

Since the prehistoric period, waste production is still becoming a major concern as more and more waste is being generated through most human activities (Ebikapade Amasuomo et al., 2016). In some areas such as the Middle East and northern Africa, more than 50% of waste is being dumped openly while areas such as East Asia and the Pacific produce the most municipal waste (Min Tian et al., 2019). According to the World Bank report back in September 2018, 3.4 billion tons of waste a year will be accumulated around the world by 2050 (Min Tian et al., 2019).

During this covid-19 pandemic, waste management are being severely affected by the limitations on commercial activities, mobility and manufacturing sector, and it is important to highlight that waste management plays a very important role in human development and health outcomes, especially in these times of need (Samuel and Phebe, 2020). Many countries encourage their residents to wear facemasks in public in order to reduce the probability of infection. This causes the increase of facemasks waste as an estimated 129 billion facemasks are in demand each month, also including the global demand for gloves, as they reach 65 billion demands monthly (Dwi Hantoko et al., 2020). These waste are labeled recyclable and can later be disinfected and recycled for the use of the medical sector if they are managed properly using waste classification.

Since waste management processes requires a number of technical, climatic, environmental, demo-graphic, socio-economic and legislative parameters, artificial intelligence (AI) techniques are adapted in tackling these ill-defined problems as they learn from experience, and handling uncertainty and incomplete data (Mohamed Abdallah et al., 2020). The advancement of AI technologies and conventional computational techniques' limitations has caused many AI-based models to be implemented in almost all fields of study, including waste management (Mohamed Abdallah et al., 2020). Examples of AI-based models are Artificial neural networks (ANN), Support vector machines (SVM), Linear regression analysis (LR), Decision tree (DT), and Genetic algorithm (GA).

In this paper, we will be focusing on two instance-based algorithms to sort waste images which are Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). The reason is to compare the reliability of the algorithms when classifying waste images. This paper also aims to build a classification system based on these approaches.

In a study conducted by Nursalim et al (2014), the KNN method gives an accuracy of 83.33% when compared to other methods such as Naive Bayes (NB), Decision Tree (DT), Neural Network (NN) and Support Vector Machine (SVM) when classifying graduate job fields (Y Resti et al., 2019) whereas in another experiment which involves classifying waste images, KNN gives a percentage of 88.0% (Bernardo S. Costa et al., 2018). SVM, on the other hand, gives a percentage of 80.0%. In another study, conducted by Widhiasih et al (2013), using KNN to classify star fruit based on RGB variables gives an accuracy of 91% (Y Resti et al., 2019).

2.2 Facts and Findings

In this section, details of the essential elements that are to be included in the project are explained in order to give a deeper understanding of the problem at hand.

2.2.1 Waste Classification

Wastes come in many forms, thus, needing to be grouped and classified to ease the process of proper disposal. The most commonly used classifications are illustrated below.

A. **Physical state**

a. Solid waste

b. Liquid waste

c. Gaseous waste

B. **Source**

a. Household/Domestic waste

b. Industrial waste

c. Agricultural waste

d. Commercial waste

e. Demolition and construction waste

f. Mining waste

C. **Environmental impact**

a. Hazardous waste

b. Non-hazardous waste

In this project, however, only solid waste will be discussed in detail, excluding hazardous solid waste.

2.2.1.1 Solid Waste

According to Rick Leblanc (2020), solid waste is created from human and animal activities and are disposed of as they do not serve any purpose related to the current or future tasks in the particular society, household or environment. Solid waste is processed and handled in various ways to ensure they are properly disposed of and would not cause harm to the environment and citizens. Some categories of solid waste, with relation to this project are plastic, cloth, glass, paper and organic waste.

Some images used for training and testing are as follows:

Recyclable:

Figure 2.1 Plastic Waste

Figure 2.2 Cans

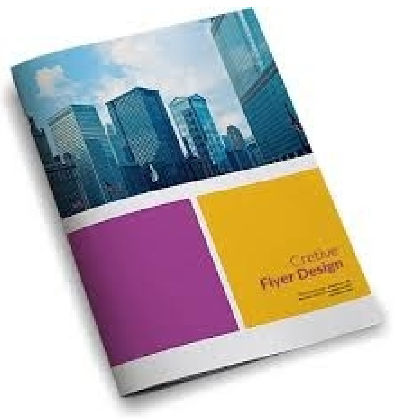


Figure 2.3 Magazine

Figure 2.4 Metal spoon

Figure 2.5 Curtain

Organic:

Figure 2.6 Fruits

Figure 2.7 Plant

Figure 2.8 A dish

Figure 2.9 Rotten strawberries

Figure 2.10 Carrots

2.2.2 Classifying Waste Using Image Recognition

In this project, we will perform image recognition to identify the objects and associate them with the correct category of waste. Usually, waste sorting devices use sorting techniques such as laser infrared and infrared sensors to differentiate between paper and plastic while monochromatic source laser and a detector can measure the intensity of the light passing through objects to differentiate between materials (Zol, 2012).

2.2.3 Instance-Based Algorithm

Instance-based learning can be defined as a type of machine learning technique that learns by generalizing or predicting the outcome from the data provided to it. This technique also finds the similarity between features in the training set that occurs in the data to help it predict the outcome. Instance-based learning algorithms store all the data instead of creating weights or building a model like other types of algorithms.

Examples of instance-based learning algorithms are as follows:-

1. k-nearest neighbours (KNN)
2. Kernel machines
3. RBF networks

2.2.4 K-NN

The K-Nearest Neighbor method (KNN) has been listed as one of the top 10 machine learning algorithms as it is simple and efficient. Not only that, this method has managed to perform machine learning tasks such as classification, regression and missing value imputation. This method predicts the label of a test data based on the major class of its k most similar training data points in the feature space (Shicai Zhang et al, 2017).

Okfalisa et al (2017) simply defined KNN as a technique to classify objects according to the closests distance of the new object to the training data based on the euclidean equation formula.

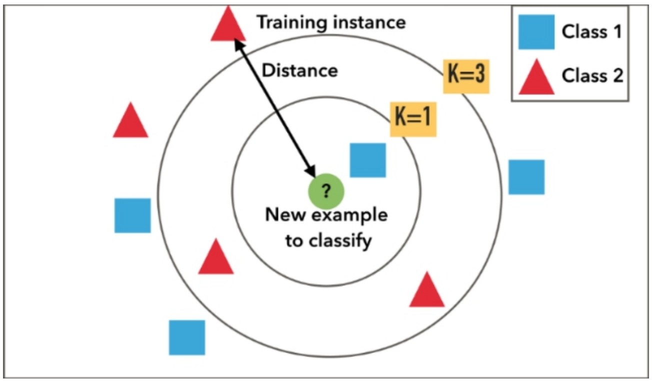


Figure 2.11 KNN

https://miro.medium.com/max/650/1*OyYyr9qY-w8RkaRh2TKo0w.png

2.2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM), also known as kernel machine, consists of well-developed theory, convex formulations, and flexibility in implementing pre-knowledge of the dependencies in the input. Because of that, they are regarded as one of the successful methods used in machine learning (Huan Song, 2017).

SVM basically constructs a hyperplane or a set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers' detection. The best hyperplane is one that almost perfectly separates the labels with the largest distance to the nearest training data of any of the labels.

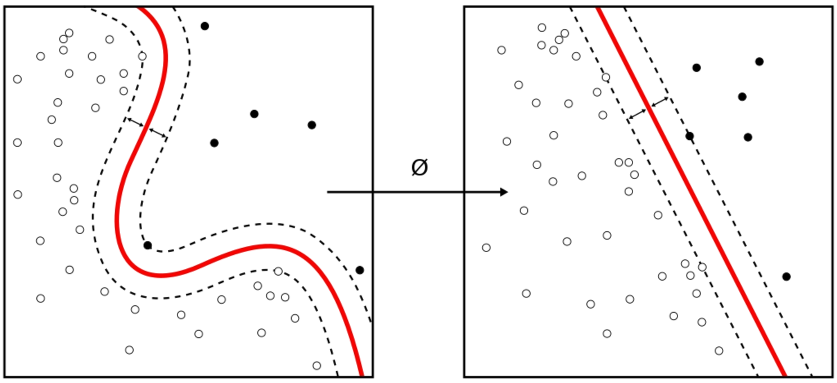


Figure 2.12 SVM

https://upload.wikimedia.org/wikipedia/commons/thumb/f/fe/Kernel_Machine.svg/1024px-Kernel_Machine.svg

2.2.5.1 Kernel Functions

Kernel functions basically gives us the plane that separates the existing data according to their respective labels or classes with a maximum distance or margin between the data and the plane. Kernel functions may separate the data linearly, non-linearly or overlapping (Deepika Kancherla et al, 2019). Linearly separable means the data of two classes is able to be separated using a linear boundary while non-linearly separable means that the data is able to be separated using a non-linear boundary. Other than that, if the data cannot be classified using linear or non-linear boundary, it is known as overlapping. Overlapping data more often than not needs to be transform into a higher dimensional space in order to create a hyperplane to separate the data.

Different kernel types such as Linear, Polynomial, Radial Basis Function (RBF) and Sigmodial can be used for classification using SVM. In this project, we will be focusing on Polynomial and RBF kernel types for classification. The polynomial kernel SVM model will be named "SVM" while the RBF kernel SVM model will be named "SVM-RBF".

2.2.6 Naïve Bayes (NB)

Using Bayes theorem, the Naïve Bayes classifier is a machine learning algorithm which implements probability calculation (Anisha et al, 2020). This classifier is considered popular due to its simplicity and short computational costs, thus making it deployable for various fields (Aji Prasetya Wibawa et al, 2019). The Bayes Theorem formula is as follows:

$$P(Q|X) = \frac{P(X|Q).P(Q)}{P(X)}$$

Where

X	Data with unknown class
Q	The hypothesis X is a specific class
P(Q|X)	The probability of the Q hypothesis refers to X

P(Q) Probability of the hypothesis Q (prior probability)

P(X|Q) Probability X in the hypothesis Q

P(X) Probability X

2.2.7 Existing System

An automatic plastic waste system created by Janusz Bobulski and Marius Kubanek (2019) used a Convolutional Neural-Network (CNN) technique to classify waste images that are commonly found in households.

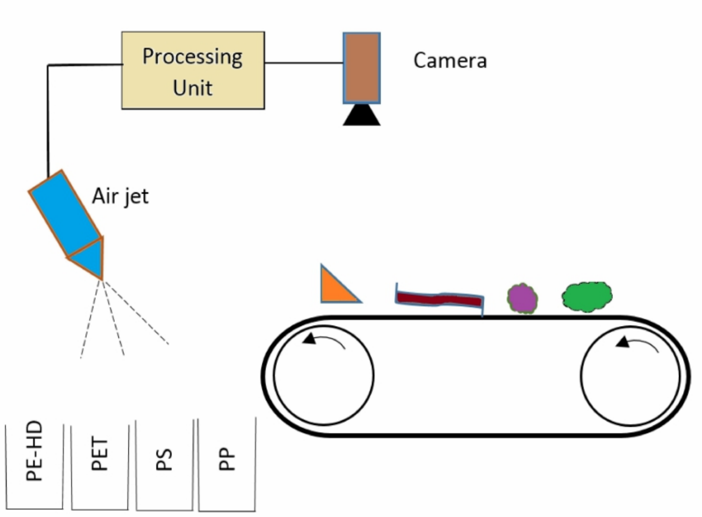


Figure 2.13 Image Classification Experiment

In their experiment, they used CNN and compared it with AlexNet, a classifying algorithm, and the outcome is that CNN performs better than AlexNet with shorter learning time. They classified plastic waste based on their type with the sole goal of performing AI classification specifically on plastic waste.

NETWORK	No. of Layers	Size of images	Accuracy [%]	Learning time [s]
AlexNet	23	120x120	96,41	364
Our Network	**15**	**120x120**	**99,92**	**217**
AlexNet	23	227x227	99,23	725
Our Network	15	227x227	91,72	540

Table 2.1 Accuracy and learning time of these algorithms when classifying the plastics

Their experiment proves to be more complicated as they are able to recognize the type of plastic by just classifying recyclables from perishables. Our project will be to perform the same means of image recognition but with instance-based algorithms.

2.2.8 Performance Evaluation

For this project, we will compare the average accuracy, precision, recall and f1-measure of the models. Firstly, we will explain the common terms when using these evaluation techniques.

True positive (TP) - value of correctly predicted positive class.

True negative (TN) - value of correctly predicted negative class.

False positive (FP) - value of incorrectly predicted positive class.

False negative (FN) - value of incorrectly predicted negative class.

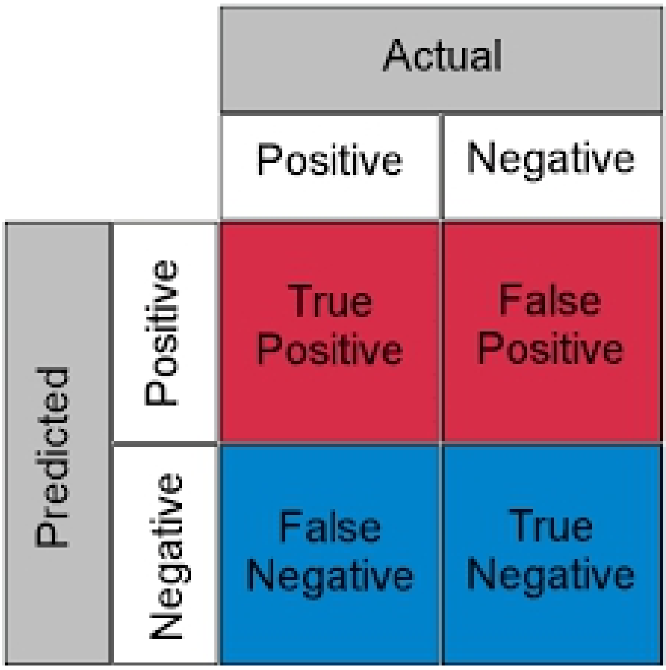


Figure 2.14 The confusion matrix of the terms

According to Wikipedia, accuracy can be defined as the closeness of measurement to a true value. Accuracy can also be defined as the proportion of correct predictions (both true positives and true negatives) among the total number of data involved.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision can be defined as the closeness of the measurement to other values.

Precision = TP / (TP + FP)

Recall can be defined as the fraction of the correct predictions

Recall = TP / (TP + FN)

F-measure is the combination of precision and recall. Its function is to solve the balance of precision and recall in one value.

*F-Measure = (2 * Precision * Recall) / (Precision + Recall)*

2.3 Summary

In conclusion, this chapter reveals a significant review of various literature which strongly relates to the project in hand and gives a better and clearer understanding of how the project will be done.

CHAPTER 3 PROJECT METHODOLOGY

3.1 Introduction

This chapter will present the project methodology and explain how the project will be completed. Methodology provides clarity to the progress of the project and the general steps on how to achieve the goal of the project.

3.2 Methodology

The methodology used in this project is the Top Down design approach. The reason we use this methodology is because it is easy to implement as the project simply aims in solving a data analytic problem which need not require system testing.

3.2.1 Literature Review

In this part of the methodology, journals, articles, trusted websites and portions of magazines related to waste management, image classification, SVM and KNN algorithms, Python coding and machine learning models' performance evaluation are searched, compiled and referred to in the making of the report and also to understand the underlying problem regarding classification using instance-based machine learning models.

This part of the methodology has also given a deeper insight on the possibility of classifying complicated extensions of the new types of waste that continues to be produced due to the increase of the evolvement of technology, and to what degree of accuracy can image classification using instance-based algorithm be used to separate the different types of wastes.

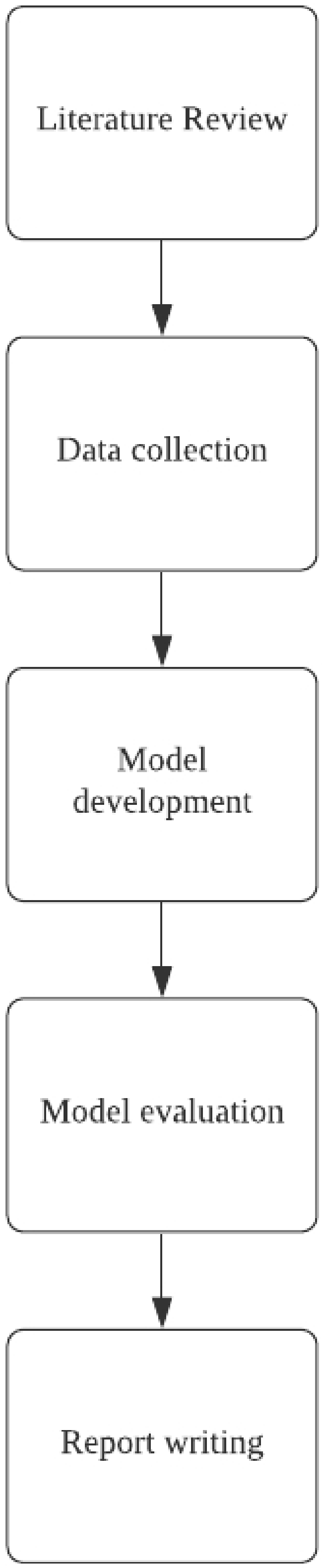


Figure 3.1 Top Down design methodology

3.2.2 Data Collection

In this part of the methodology, we collect scraped data from https://www.kaggle.com/, a famous website which stores authentic datasets and encourages data science experiments by publicizing the datasets to the whole Internet community. The waste images for the project are obtained from this website and are used during both training and testing of the machine learning model.

3.2.3 Model Development

In this part of the methodology, we produce the machine learning models using the Python language, with help from tutorials and numerous sources on the Internet. The model trains and tests the dataset many times until the accuracy of the model's prediction is at a satisfying percentage. During this phase, we built two SVM models with different kernels – polynomial and Radial Basis Function (RBF). The polynomial kernel SVM model well be named “SVM” while the RBF kernel SVM model will be named as “SVM-RBF”.

3.2.4 Model Evaluation

In this part of the methodology, we evaluate the models by reusing the dataset while changing the training and testing data within the same percentage of distribution. We then obtain the model and proceed to use the model to compare with other models to prove which models gives the best performance.

3.2.5 Report Writing

In this part of the methodology, the development and outcome of the project will be recorded in the form of a report. With the help of the Final Year Project guide, we follow its report format and include all the diagrams, tables, and references used in the project.

3.3 Project Milestones

WEEK	ACTIVITY	NOTE/ACTION
1	Chapter 4 Chapter 5	Deliverable – Chapter 4 Action – Student
2 (26/7 -> 1/8)	Chapter 5 Project Progress	Action - Student, Supervisor Progress Presentation 1 (PK 1)
3 (2/8 -> 8/8)	Chapter 5 Chapter 6	Deliverable – Chapter 5 Action - Student
4 (9/8 -> 15/8)	Chapter 6 Project Progress	Action - Student, Supervisor Progress Presentation 1 (PK 2)
5 (16/8 -> 22/8)	Chapter 6 Chapter 7 Presentation Schedule	Deliverable – Chapter 6 Action – Student Action – PSM/PD Committee
6 (23/8 -> 29/8)	Chapter 7 PSM2 Draft Report	Deliverable – Chapter 7 Action - Student, Supervisor
7 (30/8 -> 5/9)	Final Presentation & Project Demonstration Submission of PSM Draft Report into Ulearn	Action - Student, Supervisor, Evaluator Deliverable – Complete PSM Draft Report
8 (6/9 -> 12/9)	Final Examination Weeks PSM2 Draft Report correction Ulearn Submission	Action – Student, Supervisor Deliverable – PSM2 Logbook and EoS

	Submit EoS Survey Form	Survey Form
9 (13/9 -> 19/9)	Inter-Semester Break Submit corrected PSM2 Report	Deliverable – PSM2 Report Action – Student, Supervisor

Table 3.1 Project Milestones

Task	1	2	3	4	5	6	7	8	9
Chapter 4									
Chapter 5									
Chapter 6									
Chapter 7									
Final Presentation and Evaluation									
Submission of PSM 1 Final Report									

Table 3.2 Gantt Chart

3.4 Conclusion

In conclusion, this chapter gives a formal presentation of how the project is planned according to the chosen methodology and the estimated milestones. The next chapter will focus on the project's design and analysis phase.

CHAPTER 4 ANALYSIS AND DESIGN

4.1 Introduction

This section of the report will discuss the analysis and design part of the project in more detail. Analysis is a crucial part of the project as the main objective of this project is to analyze the models used and compare them in order to obtain the best model with the higher accuracy and prediction rate. Analysis also helps us to recognize the requirements needed to complete the project and thus brings ease to solve the problem stated and hopefully for future problems with the same dilemma. We will discuss this phase in a few general subtopics which are problem analysis, requirement analysis, data requirement, functional and nonfunctional requirements, and other necessary requirements. The design of the project is also highlighted as it represents the project as a whole and gives more understanding on how this project has been done in a formal way. SVM and KNN are two famous instance-based algorithms for machine learning and can be used for image classification while Naïve Bayes is a probability classifier that can be used in various fields. Using the waste classification dataset from Kaggle (Waste Classification), we will use these models to classify the images of organic and recyclable wastes and retrieve their accuracy before comparing the usefulness of the models.

4.2 Problem Analysis

Waste is a global problem and the increasing need to solve it due to the rise in human activities and the need to lessen the use of natural resources has pushed the use of Artificial Intelligence in order to sort them in a quick and accurate manner. Around 7 to 10 billion tonnes per annum of urban waste is expected to accumulate worldwide and developing countries are doing their utmost to reduce the cost of managing waste as the number of the population increases as years go by (Victor, 2017). It is important to manage and sort waste efficiently so as to reduce waste accumulation. Not all waste can be recycled, thus proving the importance of sorting them. The use of Artificial Intelligence for classifying waste not only offers benefits to the environment, but also positively affects the growth of the economy (Bernardo et al, 2018).

Instance-based algorithms such as SVM and KNN can be used to classify images by labelling the data based on a set of boundaries that can be in one or more dimensions. SVM is an algorithm which helps to figure out convex optimization problems in an analytical way by discerning the label of a data point using a discriminant function. KNN is helpful to determine the label of a data point based on the distance to the nearest training point of a specific labelled cluster.

4.3 Requirement Analysis

The following section will introduce the requirements for the purpose of achieving the goal of this project which is to analyze the machine learning models.

4.3.1 Data Requirement

The waste classification dataset obtained for this project is from Kaggle Waste Classification. There are two folders which are train and test and each folder consists of two subfolders labelled ‘O’ which stands for organic and ‘R’ which stands for recyclable.

The dataset of the train and test folder are combined into only two folders which are ‘O’ and ‘R’. This is to ease the shuffling and splitting of the data using the Python API train_test_split. The total number of images is 17852.

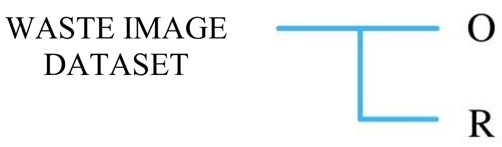


Figure 4.1 Folder Structure

4.3.2 Functional Requirement

Functional requirement basically defines the system's behavior when given an input. It includes calculations, data input and business processes. Functional requirements must be met in order to make the system work.

Calculations for distance measurement are done through a Python API (sklearn). Data inputs are the dataset as stated in the section above and will be split into training and testing sets. Testing set is 25% of the overall data.

4.3.3 Non-functional Requirement

Non-functional requirements specify the actions of the system, in other words, how exactly the system should behave. They are not compulsory as the system will continue to perform its basic functionality even if the non-functional requirements are not met.

Accuracy and prediction of the models are obtained. In the case of KNN, the best k will be used to obtain the highest possible accuracy before obtaining its average accuracy.

4.3.4 Others Requirement

Requirements for software and hardware will be discussed in this section of the report.

4.3.4.1 Software Requirement

i. Jupyter Notebook
ii. Windows 10 Home
iii. Docs Google

4.3.4.2 Hardware Requirement

Processor	Intel(R) Core(™) i5-8250U @ 1.60GHz 1.80GHz
Installed RAM	8.00 GB
System Type	64-bit operating system, x64-based processor

Table 4.1 Laptop

Edition	Windows 10 Home Single Language
Version	1909
OS build	18363.1556

Table 4.2 Operating System

4.4 High-Level Design

The aim of this project is to analyze the development of instance-based machine learning models to perform image classification and to evaluate the models. The two models which will be evaluated are SVM and KNN. NB will also be evaluated in order to compare the instance-based models with another type of machine learning classifier. The models will be evaluated by comparing the accuracy, precision, recall and F1 measure of the models.

4.4.1 System Architecture

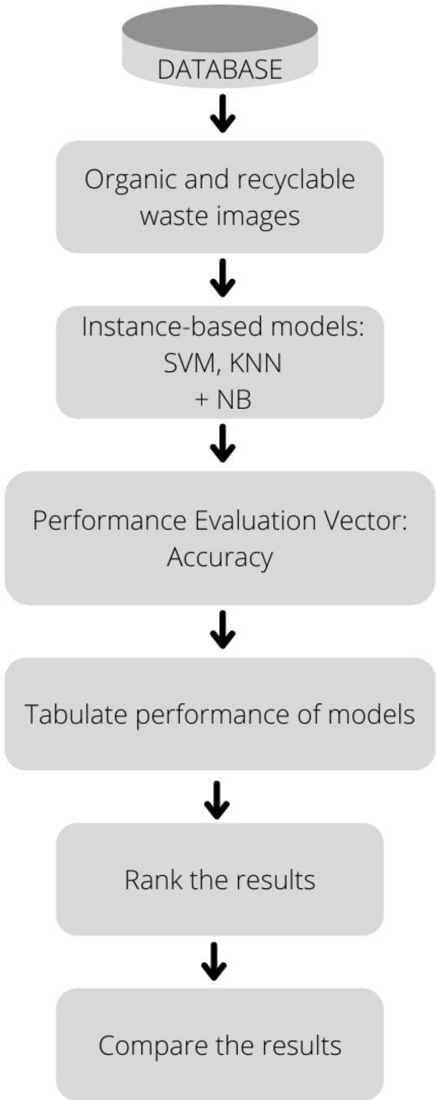


Figure 4.2 System architecture

Figure above shows the flowchart of the system architecture. Organic and recyclable images obtained from Kaggle will be split into train and test sets before fitted to a machine learning model for training. The model will then be used with the test set to obtain the accuracy and precision of the model before being compared.

4.4.2 Database Design

The database design for this project consists of datasets of organic and recyclable images taken from Kaggle. The database design includes the two subfolders, 'O' for organic and 'R' for recyclable, are in the main folder DATASET2. The figure below shows the design of the database.

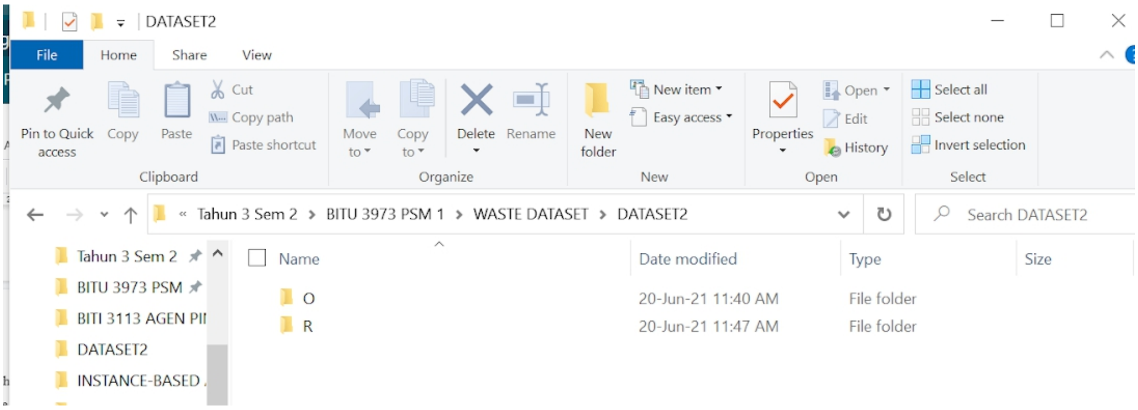


Figure 4.3 Database Design

4.5 Summary

This chapter has discussed the project flow, the requirements met and the design of the database. This chapter is important to give more understanding to the flow of the project as a whole. The next chapter will focus on the implementation phase of the project.

CHAPTER 5 IMPLEMENTATION

5.1 Introduction

In this chapter, we discuss the implementation phase of the project and the outputs of the models which includes the accuracy, precision, recall and F1 measure of the models. Several software such as Anaconda, which helps to code in Jupyter Notebook, were used to ensure that the models are built efficiently, and the results can be seen clearly.

5.2 Environment Setup

Jupyter Notebook was used to perform Python codings and build the models. We used Python 3.7.4 despite not being the most advanced version of Python. This is due to the 3.0 version of Python being compatible with most API, and thus eases the process of model building.

Anaconda allows the use of the Jupyter Notebook in the web browser and performs machine learning modelling using Python. The codes are able to run in an unordered manner and thus allow flexible coding without having to build the model again and again just to test the accuracy of the model using different data.

Figure 5.1 Anaconda

Figure 5.2 Jupyter Notebook

Figure 5.3 Python

5.3 Flow of the system

From building the model to the evaluation process, we will go through several processes which are import, specify, load and predict.

5.3.1 Import

Python libraries are imported into the Anaconda environment so that it can be used when coding Python.

5.3.2 Specify

The location of the dataset and the saving folder for the model is specified in the DATASET2 as mentioned in the prior chapter. For KNN, we firstly obtain the best k and specify the best k into the model.

5.3.3 Load

After creating the model using the training data, we export the model using 'pickle', a Python API. Then, we load once more into the Jupyter Notebook and perform image classification on the testing data.

The data will be shuffled using the random.shuffle() function to ensure there is no bias in choosing the training and testing data.

5.3.4 Predict

We then used the models to predict the labels of the testing data. The model will then give the value of the model's performances and the result of the prediction.

5.4 Summary

In conclusion, we have described how we implement the model using the required software and how we compared the accuracy of both models.

CHAPTER 6 TESTING

6.1 Introduction

In this chapter, we discuss the testing phase of the project and how well the performance of the models. The testing phase will determine which model is better in classifying images.

6.2 Performance Evaluation Vector

The performance evaluation vectors to evaluate the models are as follows:

Performance Evaluation Vector	**Description**
Accuracy	Ratio of correct prediction to total prediction
Precision	Ratio of correct prediction to total positive prediction
Recall	Ratio of correctly predicted observation to actual positive class
F1 Measure	Weighted average precision of precision and recall

Table 6.1 Performance Evaluation Vector

6.3 Test Implementation

This section will explain in detail the waste images dataset used in this project.

6.3.1 Test Data

In this project, we will use 6741 images of organic waste and 11 111 images of recyclable waste. Image size are set to (50,50).

6.5 Test Results and Analysis

This section will display the results of the analysis.

6.5.1 Result and Analysis of Performance of Models

Performance	SVM	SVM-RBF	KNN	NB
Accuracy	0.906	0.907	0.662	0.613
Precision	0.931	0.87	0.694	0.722
Recall	0.917	1.0	0.814	0.612
F1 Measure	0.924	0.931	0.749	0.662
Prediction	R	R	R	R

Table 6.2 Performance Evaluation for Models

Next, we rank the performance of the models and calculate their cumulative score.

Scoring : 1 – Poor, 4 – Excellent

Performance	**SVM**	**SVM-RBF**	**KNN**	**NB**
Accuracy	3	4	2	1
Precision	4	3	1	2
Recall	3	4	2	1
F1 Measure	3	4	2	1
Total Score	13	15	7	5

Table 6.3 Ranking Score for Performance Measures of Models

From table 6.3, we can conclude that the SVM-RBF model performs much better than SVM, KNN and NB by observing the ranking score. This proves SVM when using an RBF kernel is more suitable to perform waste classification in real life.

6.6 Summary

This chapter gives the conclusion to the analysis and reaches the goal which is to evaluate the models and choose the best for classifying waste images. The next chapter will give an overall conclusion of the project as a whole.

CHAPTER 7 PROJECT CONCLUSION

7.1 Introduction

This chapter concludes the project as a whole. Contribution of this project and possible improvements for this project will also be discussed.

7.2 Project Summarization

Every project has its ups and downs, including this project. We will recognize the strengths and weakness of the project so as to motivate for the betterment of the project and prepare for future improvements.

7.2.1 Strengths

The model is able to display a high accuracy which makes it reliable to be used in classifying wastes.

7.2.2 Weaknesses

The images used are not perfectly filtered, especially in the organic folder as some organic images are the patterns on a recyclable item. The dataset needs more plain images and without patterns as it could affect the training process of the models. Some waste images such as plastic or paper strips can also be misunderstood as peelings of fruits or vegetables.

7.2.3 Future Improvements

In the future, we will add more machine learning models to find a better solution for classifying these images. The dataset will also be revised and only use reliable images for building the models.

7.3 Project Contribution

The development of the SVM model can be used to further ease the task of distinguishing organic and recycled waste. This is as stated in the objective which is to develop and choose the best model to be used for classifying waste. The model can be used in garbage dump facilities and reduce the time taken for the workers to sort trash significantly.

7.4 Project Limitation

The limit of the project is the use of sensors to provide real life application of the model on wastes. The lack of reliable dataset is also a problem for training and testing the model.

7.5 Conclusion

In conclusion, this project requires more maintenance and solid performance evaluation in order to be able to judge the dependability of the model better. All the objectives are able to be fulfilled and the SVM model has proved to be useful in classifying the waste images.

REFERENCES

Costa, B. S., Bernardes, . C., Pereira, J. V., Zampa, V. H., Pereira, V. A., Matos, G. F., … Silva, F. (2018). Artificial Intelligence in Automated Sorting in Trash Recycling.

Prasanna M, A., Vikash Kaushal, S., & Mahalakshmi, P. (2018). Survey onidentification and classification of waste for efficient disposal and recycling. International Journal of Engineering & Technology, 7(2.8), 520. https://doi.org/10.14419/ijet.v7i2.8.10513

Amasuomo, E., & Baird, J. (2016). The Concept of Waste and Waste Management. Journal of Management and Sustainability, 6(4), 88. https://doi.org/10.5539/jms.v6n4p88

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 2(3). https://doi.org/10.1007/s42979-021-00592-x

Hantoko, D., Li, X., Pariatamby, A., Yoshikawa, K., Horttanainen, M., & Yan, M. (2021). Challenges and practices on waste management and disposal during COVID-19 pandemic. Journal of Environmental Management, 286, 112140. https://doi.org/10.1016/j.jenvman.2021.112140

Iqbal, A., Liu, X., & Chen, G. H. (2020). Municipal solid waste: Review of best practices in application of life cycle assessment and sustainable management techniques. Science of The Total Environment, 729, 138622. https://doi.org/10.1016/j.scitotenv.2020.138622

Bobulski, J., & Kubanek, M. (2021). Deep Learning for Plastic Waste Classification System. Applied Computational Intelligence and Soft Computing, 2021, 1–7. https://doi.org/10.1155/2021/6626948

Resti, Y., Mohruni, A., Rodiana, T., & Zayanti, D. (2019). Study in Development of Cans Waste Classification System Based on Statistical Approaches. Journal of Physics: Conference Series, 1198(9), 092004. https://doi.org/10.1088/1742-6596/1198/9/092004

Sharma, H. B., Vanapalli, K. R., Cheela, V. S., Ranjan, V. P., Jaglan, A. K., Dubey, B., Goel, S., & Bhattacharya, J. (2020). Challenges, opportunities, and innovations for effective solid waste management during and post COVID-19 pandemic. Resources, Conservation and Recycling, 162, 105052. https://doi.org/10.1016/j.resconrec.2020.105052

Song, H., Thiagarajan, J. J., Sattigeri, P., & Spanias, A. (2018). Optimizing Kernel Machines Using Deep Learning. IEEE Transactions on Neural Networks and Learning Systems, 29(11), 5528–5540. https://doi.org/10.1109/tnnls.2018.2804895

Rebinth, A., & Kumar, S. M. (2021). Wavelet Packet Transform-Based Image Classification for Computer-Aided Glaucoma Diagnosis Using Naïve Bayes Classifier. In Communication Software and Networks (pp. 597-605). Springer, Singapore.

Murti, R. P. A., Putra, S. M., Kurniawan, S. A., & Nugraha, Y. R. Naïve Bayes Classifier for Journal Quartile Classification.

Wikipedia contributors. (2021, June 10). Precision and recall. Wikipedia. https://en.wikipedia.org/wiki/Precision_and_recall#Recall

Wikipedia contributors. (2021a, May 24). Instance-based learning. Wikipedia. https://en.wikipedia.org/wiki/Instance-based_learning

Shanmukh, V. (2021, March 5). Image Classification Using Machine Learning-Support Vector Machine(SVM). Medium. https://medium.com/analytics-vidhya/image-classification-using-machine-learning-support-vector-machine-svm-dc7a0ec92e01

APPENDICES

IMPORT LIBRARIES

```
In [37]: %matplotlib inline
import os
import numpy as np
import cv2
import matplotlib.pyplot as plt
import pickle
import random
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import roc_curve
from sklearn.datasets import load_wine
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import precision_recall_curve
#from sklearn.metrics import plot_precision_recall_curve
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn. metrics import f1_score
from sklearn import metrics
from tabulate import tabulate
```

DECLARE DIRECTORIES

```
In [2]: dir = 'C:\\Users\\User10\\Desktop\\Tahun 3 Sem 2\\BITU 3973 PSM 1\\WASTE DATASET\\DATASET2'

categories = ['O','R']

data = []

for category in categories:
    path = os.path.join(dir,category)
    label = categories.index(category) # get zero and one (distinguish O and R)

    for img in os.listdir(path):
        imgpath = os.path.join(path,img)
        waste_image = cv2.imread(imgpath,0)
        try:
            waste_image = cv2.resize(waste_image,(50,50)) # make all images same size
            image = np.array(waste_image).flatten() # make full image single area
            data.append([image,label]) # append image and label
        except Exception as e:
            pass
```

APPEND FEATURES AND LABELS INTO PICKLE AND SHUFFLE AND SPLIT THE DATA

```
In [3]: pick_in = open('data.pickle','wb') # create pickle to save data, wb means write
pickle.dump(data,pick_in) # dump data in pick_in
pick_in.close() # close the pickle
```

```
In [4]: pick_in = open('data.pickle','rb') # read pickle file
data = pickle.load(pick_in) # load pick_in file, must in same directory with this python file
pick_in.close()
```

```
In [7]: ### SHUFFLE DATA
random.shuffle(data)
features = []
labels = []

for feature, label in data:
    features.append(feature)
    labels.append(label)

### SPLIT DATA INTO TRAIN AND TEST SET
xtrain, xtest, ytrain, ytest = train_test_split(features,labels,test_size=0.25) #test size is 25% from dataset
```

FIND THE BEST VALUE FOR K FOR KNN CLASSIFIER

In [8]:
```python
### FIND BEST K
## try K=1 through K=25 and record testing accuracy
k_range = range(1, 26)

# We can create Python dictionary using [] or dict()
scores = []

# We use a loop through the range 1 to 26
# We append the scores in the dictionary
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(xtrain, ytrain)
    ypred = knn.predict(xtest)
    scores.append(metrics.accuracy_score(ytest, ypred))

print(scores)

# plot the relationship between K and testing accuracy
# plt.plot(x_axis, y_axis)
plt.plot(k_range, scores)
plt.xlabel('Value of K for KNN')
plt.ylabel('Testing Accuracy')
```

```
[0.6066666666666667, 0.6296666666666667, 0.6223333333333333, 0.643, 0.6433333333333333, 0.6526666666666666, 0.652, 0.6533333333
333333, 0.6493333333333333, 0.6576666666666666, 0.6553333333333333, 0.6566666666666666, 0.6573333333333333, 0.6656666666666666,
0.6586666666666666, 0.6663333333333333, 0.6613333333333333, 0.665, 0.6666666666666666, 0.665, 0.6593333333333333, 0.66633333333
33333, 0.6613333333333333, 0.6663333333333333, 0.6653333333333333]
```

Out[8]:
```
Text(0, 0.5, 'Testing Accuracy')
```

TRAIN THE MODELS

In [ ]:
```python
### CREATE Simple Naive Bayes Model
#Import Gaussian Naive Bayes model
from sklearn.naive_bayes import GaussianNB

#Create a Gaussian Classifier
gnb = GaussianNB()

#Train the model using the training sets
nb = gnb.fit(xtrain, ytrain)

#Predict the response for test dataset
ypred = nb.predict(xtest)

#Accuracy
print("Accuracy:",metrics.accuracy_score(ytest, ypred))
```

In [ ]:
```python
### CREATE KNN MODEL
# instantiate the model with the best known parameters
knn = KNeighborsClassifier(n_neighbors=21) ### input best k in the n_neighbors
knn.fit(xtrain, ytrain)
```

In [ ]:
```python
### CREATE SIMLPE SVM POLY MODEL
svm = SVC(C=1,kernel='poly',gamma='auto')
svm.fit(xtrain,ytrain) # give features and labels to train the model

### CREATE SVM RBF MODEL
svm_rbf = SVC(C=1,kernel='rbf',gamma='auto')
svm_rbf.fit(xtrain,ytrain) # give features and labels to train the model
```

SAVE THE MODELS

In [ ]:

```
### SVM MODEL
pick = open('svm_model.sav','wb')
pickle.dump(svm,pick)
pick.close()

### SVM RBF BASED KERNEL
pick = open('svm_rbf_model.sav','wb')
pickle.dump(svm_rbf,pick)
pick.close()

### KNN MODEL
pick = open('knn_model.sav','wb')
pickle.dump(knn,pick)
pick.close()

### Save Naive Bayes MODEL
pick = open('nb_model.sav','wb')
pickle.dump(nb,pick)
pick.close()
```

PERFORMANCE EVALUATION OF THE MODELS

In [9]:

```
### TESTING SVM MODEL
svm_prediction = svm.predict(xtest)
svm_accuracy = svm.score(xtest,ytest)
svm_precision = precision_score(ytest,svm_prediction)
svm_recall = recall_score(ytest,svm_prediction)
svm_f1_metric = f1_score(ytest,svm_prediction)

### TESTING SVM RBF BASED KERNEL
svm_rbf_prediction = svm_rbf.predict(xtest)
svm_rbf_accuracy = svm_rbf.score(xtest,ytest)
svm_rbf_precision = precision_score(ytest,svm_rbf_prediction)
svm_rbf_recall = recall_score(ytest,svm_rbf_prediction)
svm_rbf_f1_metric = f1_score(ytest,svm_rbf_prediction)

### TESTING KNN MODEL
knn_prediction = knn.predict(xtest)
knn_accuracy = knn.score(xtest,ytest)
knn_precision = precision_score(ytest, knn_prediction)
knn_recall = recall_score(ytest, knn_prediction)
knn_f1_metric = f1_score(ytest,knn_prediction)

### TESTING NB MODEL
nb_prediction = nb.predict(xtest)
nb_accuracy = nb.score(xtest,ytest)
nb_precision = precision_score(ytest,nb_prediction)
nb_recall = recall_score(ytest,nb_prediction)
nb_f1_metric = f1_score(ytest,nb_prediction)
```

TABULATE THE PERFORMANCE EVALUATION OF THE MODELS AND DISPLAY THE FIRST PREDICTED PICTURE

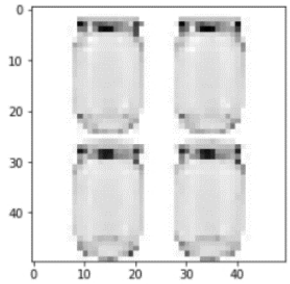
In [10]:

```
### PRINT PERFROMANCE EVALUATION
categories = ['O','R']
data = [['Performance','SVM','SVM-RBF','KNN','NB'],
        ["Accuracy",round(svm_accuracy,3),round(svm_rbf_accuracy,3),round(knn_accuracy,3),round(nb_accuracy,3)],
        ["Precision",round(svm_precision,3),round(svm_rbf_precision,3),round(knn_precision,3),round(nb_precision,3)],
        ["Recall",round(svm_recall,3),round(svm_rbf_recall,3),round(knn_recall,3),round(nb_recall,3)],
        ["F1 Measure",round(svm_f1_metric,3),round(svm_rbf_f1_metric,3),round(knn_f1_metric,3),round(nb_f1_metric,3)],
        ["Prediction",categories[svm_prediction[0]],categories[svm_rbf_prediction[0]],categories[knn_prediction[0]],categori

print(tabulate(data, headers='firstrow',tablefmt='fancy_grid'))

### SHOW PREDICTION
waste_classified = xtest[0].reshape(50,50)
plt.imshow(waste_classified,cmap='gray')
plt.show()
```

Performance	SVM	SVM-RBF	KNN	NB
Accuracy	0.906	0.907	0.662	0.613
Precision	0.931	0.87	0.694	0.722
Recall	0.917	1.0	0.814	0.612
F1 Measure	0.924	0.931	0.749	0.662
Prediction	R	R	R	R



PREDICT ANOTHER PICTURE

In [12]:

```
categories = ['O','R']

#Predict another picture
imgpath = "C:\\Users\\User10\\Desktop\\BITU 3973 PSM 1\\Waste Classification Using Instance-Based Algorithm\\dirty-shirt.jpg

waste_image = cv2.imread(imgpath,0) # value 0 is to turn image into grayscale

try:
    waste_image = cv2.resize(waste_image,(50,50)) # make all images same size
    test_image = np.array(waste_image).flatten() # make full image single area
except Exception as e:
    pass

svm_prediction = svm.predict(test_image.reshape(1,-1))
svm_rbf_prediction = svm_rbf.predict(test_image.reshape(1,-1))
knn_prediction = knn.predict(test_image.reshape(1,-1))
nb_prediction = nb.predict(test_image.reshape(1,-1))

### SHOW PREDICTION
plt.imshow(waste_image)

### Print prediction
print('SVM Prediction is ',categories[svm_prediction[0]])
print('SVM-RBF Prediction is ',categories[svm_rbf_prediction[0]])
print('KNN Prediction is ',categories[knn_prediction[0]])
print('NB Prediction is ',categories[nb_prediction[0]])
```

```
SVM Prediction is  R
SVM-RBF Prediction is  R
KNN Prediction is  R
NB Prediction is  O
```