DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology

WASTE CLASSIFICATION AND OPTIMIZED ROUTING FOR WASTE COLLECTION

A FINAL PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

Rohan Prasai

5-2-1175-24-2018

4/17/2023

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by ROHAN PRASAI entitled **"waste classification and optimized routing for waste collection"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

Mr. Ritu Raj Lamsal

Project Supervisor

Assistant Director, Research Program

DWIT College

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

STUDENT’S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than that listed here have been used in this work.

…………………………………………

Rohan Prasai

4/17/2023

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

LETTER OF APPROVAL

This is to certify that this project prepared by ROHAN PRASAI entitled **"WASTE CLASSIFICATION AND OPTIMIZED ROUTING FOR WASTE COLLECTION"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied.

In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…………………………………… Mr. Ritu Raj Lamsal Project Supervisor Assistant Director, Research Program DWIT College	…………………………………… Mr. Hitesh Karki Campus Chief DWIT College
…………………………………… Mr. Niresh Dhakal Project Coordinator Associate Director of Undergrad Program DWIT College	…………………………………… Mr. Nawaraj Poudel Associate Professor Central Department of Computer Science and Information Technology, Kirtipur

ACKNOWLEDGEMENT

I would like to thank Mr. Ritu Raj Lamsal, Assistant Director, Research Program, Project Supervisor, DWIT College for supervising this project and giving assistance when faced with problems. Without his guidance and persistent help, this project would not have been possible.

I would also like to thank Mr. Bijay Babu Regmi, Senior Lecturer and Mr. Hitesh Karki, Chief Academic Officer, Niresh Dhakal, project coordinator for providing necessary feedback and guidance in the development of this project. They have been a great motivator and helper, encouraging me to keep moving ahead on the project.

Rohan Prasai

TU Reg. No: 5-2-1175-24-2018

4/17/2023

ABSTRACT

This project aims to develop a waste classification system using Convolutional Neural Networks (CNNs) and to optimize waste collection based on the classification results. The proposed system is designed to automatically identify and classify different types of waste materials, such as paper, plastic, glass, and metal, using image recognition technology. The CNN model is trained on a large dataset of waste images to ensure accurate classification. To obtain the data, the system utilizes flask API to access the classified data about factors such as waste category, waste filled percentage, latitude and longitude which is stored in the database using MySQL. This information is fetched by the system and integrated with the optimization algorithm to calculate the most efficient routes for waste collection. Finally, LeafletJs is used to display the optimized routes and waste collection points on a map. The system utilizes markers to represent the collection points and polylines to illustrate the optimized collection routes.

Keywords: *machine learning; classification; cnn algorithm; mysql; python; Leafletjs ;haversine formula*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

API	Application Programming Interface
CSS	Cascading Style Sheet
CV	Computer Vision Library
DB	Database
HTML	HyperText Markup Language
JS	Javascript
PY	Python
RAD	Rapid Application Development
TF	Tensorflow
UI	User Interface
VS	Visual Studio

CHAPTER 1: INTRODUCTION

1.1. Overview

With the increase in the number of industries in the urban area, the disposal of the solid waste is really becoming a big problem in Nepal. One of the major challenges in waste management is the effective collection and classification of waste. Traditional methods of waste collection and classification can be inefficient and time-consuming, leading to increased costs and environmental damage. Solid waste management center Nepal conducted survey in 2019, more than 1 million tons of garbage is produced every year in Nepal. Although 80% of the waste are recyclable waste but it is not being recycled properly [1].

Hence it is necessary to recycle the waste to protect the environment and human beings' health, and we need to separate the waste into the different components which can be recycled using different ways. Waste classification and optimized routing for waste collection are two promising approaches that can address the issue of inefficient waste management. The classification of waste materials can improve the efficiency and accuracy of waste sorting and disposal. Additionally, optimized routing can streamline the waste collection process, resulting in lower transportation costs, fewer carbon emissions, and reduced traffic congestion. This system can classify waste according to the input provided by the user and also show the recycle process and optimized routing of collecting waste from different location.

1.2. Background and Motivation

Waste management is a significant environmental issue, with a significant impact on public health, air and water quality, and climate change. Inefficient waste collection and management can lead to increased waste disposal costs and negative impacts on public health and the environment. In this context, the need for sustainable waste management solutions is evident.

Waste classification and optimized routing for waste collection are two promising approaches that can address the issue of inefficient waste management. The classification of waste materials can improve the efficiency and accuracy of waste sorting and disposal. Additionally, optimized routing can streamline the waste collection process, resulting in lower transportation costs, fewer carbon emissions, and reduced traffic congestion

1.3. Problem Statement

The important steps of waste management are the separation of the waste and manually doing this process is time consuming and is fairly inaccurate. If we could make this sorting process automatic, a huge number of times could be saved making the recycling process even more accurate.

Improper waste separation and disposal near public spaces such as parks and rivers have led to a significant challenge in waste collection systems. However, this waste classification and optimized routing application offer a solution by providing an efficient route for waste collectors to gather waste from the designated areas. This technology can help manage waste separation and disposal more effectively, reducing environmental impacts and improving the overall efficiency of waste management systems.

1.4. Objectives

The major objectives of this project are:

- Using machine learning algorithm (CNN), train the machine to classify the waste in to different categories of waste classes.
- To recycle the waste to protect the environment and human beings' health.
- To provide the route for collecting waste by calculating the shortest path and approximate fuel consumption.

1.5. Scope

- This application can be used in managing waste in different fields such as industry, metropolitan city etc.
- This routing application can be useful for waste collectors/drivers for collecting the waste from different locations.
- The classification algorithm using CNN can help recycling centers efficiently sort waste materials, increasing the amount of waste that can be recycled.
- The integration of waste management systems with other smart city technologies, such as traffic management systems, can create more efficient and sustainable cities.

1.6. Development Methodology

For the development of this project, rapid prototyping development methodology was followed which falls under agile methodology.

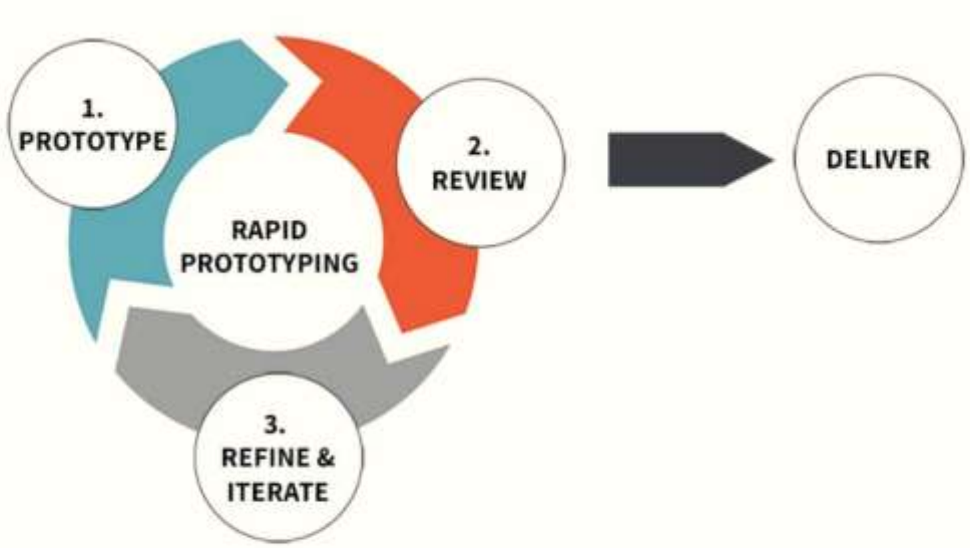


Figure 2.6.1: Project methodology using RAD model

1.7. Outline

1. The outline is divided into various sections with different information presented to the viewer about the report. Various diagrams are also shown as a clear demonstration about the plan, blueprint as well as the output of the project. The outline of the project report is shown below
2. **Preliminary Section**: Preliminary Section includes various outlines like title page, abstract, acknowledgement, table of contents, list of figures as well as list of tables.
3. **Introduction Section**: Introduction section consists of outlines which includes the overview of the project, the background and motivation of the project, problem statement with the objectives and scope.
4. **Requirement and Feasibility Analysis Section**: Requirement and Feasibility Analysis section includes functional and non-functional requirement as well as the research done about the project.
5. **System Design Section**: This section includes various diagram for understanding the overall design of the system.

6. **Implementation and Evaluation Section:** This section explains about the creation of the project as well as the evaluation results.
7. **Conclusion**: This section includes the overall theme and conclusion of the project.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1. Background Study

Waste management is a growing concern worldwide due to the increasing amount of waste generated daily. Proper waste management has become essential for environmental sustainability and public health. The traditional waste management systems are often ineffective due to the lack of an efficient waste collection system and unmanaged waste separation. These challenges have led to environmental pollution and increased health risks for communities.

To address these challenges, various technologies and strategies have been developed for effective waste management. One such technology is the use of Convolutional Neural Networks (CNNs) for waste classification. CNNs are a type of deep learning algorithm that has been widely used in image classification. In waste management, CNNs can be used to classify waste materials into various categories such as plastic, glass, organic, and metal, making it easier to sort and recycle waste materials.

Another challenge in waste management is the inefficient routing of waste collection vehicles. Inefficiencies in routing can lead to higher costs and increased environmental impacts due to increased vehicle emissions. However, advanced routing algorithms such as Haversine algorithm can be used to optimize waste collection routes and reduce environmental impacts.

To implement these technologies in waste management, various software tools and APIs can be used, such as OpenCV, TensorFlow, Leaflet Routing JS, and flask API. OpenCV and TensorFlow are open-source libraries used for image classification, while Leaflet Routing JS is a JavaScript library used for route optimization. Flask API can be used to generate synthetic waste data for testing and validation.

Overall, the use of advanced technologies such as CNNs for waste classification and routing optimization has the potential to revolutionize waste management systems. By providing efficient waste collection and management, these technologies can contribute to the development of more sustainable and environmentally friendly waste management practices.

Managing solid waste is one of the major challenges in urbanization. A survey conducted in all 58 municipalities of Nepal in 2012 found that the average municipal solid waste generation was 317 grams per capita per day. This translates into 1,435 tons per day or 524,000 tons per year of municipal solid waste generation in Nepal. Many of these technically and financially constrained municipalities are still practicing roadside waste pickup from open piles and open dumping, creating major health risks [1].

Solid waste refers to any type of garbage, trash, refuse or discarded material. It can be categorized according to where the waste is generated, for example as municipal solid waste, health care waste and e-waste. Over 2 billion tons of municipal solid waste are produced annually.

Improper disposal can lead to adverse health outcomes, for example through water, soil and air contamination. Hazardous waste or unsafe waste treatment such as open burning can directly harm waste workers or other people involved in waste burning and neighboring communities. Vulnerable groups such as children are at increased risk of adverse health outcomes. Poor waste collection leads to environmental and marine pollution and can block water drains. Resulting flooding and other standing waters in waste items favor cholera and vector-borne diseases such as malaria and dengue.

About 54 million tons of e-waste, such as TVs, computers and phones, are created annually (2019 data) with an expected increase to 75 million tons by 2030. In 2019 only 17% of e-waste was documented as being properly collected and recycled. Exposure to improperly managed e-waste and its components can cause multiple adverse health and developmental impacts especially in young children [2].

This paper aims to explore the application of CNNs for waste classification and the use of Leaflet Routing Machine for optimized waste collection. The study includes a software implementation that utilizes flask API to generate synthetic waste data and OpenCV and TensorFlow for image classification. The software also employs LeafletJS for data visualization, including the display of waste bin locations and optimized collection routes. Overall, the study highlights the potential benefits of using advanced technologies for waste management and provides insights into the development of more efficient waste management systems.

The CNN architecture's fundamental operation is the Convolution, hence the name Convolutional Network. In a nutshell, the Convolution operation takes as input the pixels of an image and outputs a feature map, which is no more than a matrix of values that can perfectly represent an image. A matrix colored in dark grey called kernel goes through all the input image pixels, multiplying each pixel value of the kernel by the corresponding pixel values of the input image. After that, it places the sum of all the resulting values in the feature map matrix sequentially. To preserve the dimensionality of the feature map with respect to the input image, the padding can be added, which is a line of null values around the image. Also, parameters are changeable like stride, which determines the 'steps' that the kernel takes to the left while traversing the image. A machine learning model can modify the values of a kernel to detect complex features like human faces, car wheels, etc.

In the context of waste classification, CNNs can be trained to classify images of waste materials into different categories, such as plastic, glass, organic, metal, and paper. This can be useful for recycling and waste management, as it allows waste materials to be sorted more efficiently and accurately.

CNNs consist of multiple layers, each with a specific function. The first layer is typically a convolutional layer, which applies a set of filters to the input image. Each filter is a small matrix of weights that is convolved with the input image, resulting in a set of feature maps that highlight different aspects of the image, such as edges, corners, or textures. The output from the convolutional layer is passed through a non-linear activation function, such as ReLU (Rectified Linear Unit), which introduces non-linearity into the model and helps to capture complex relationships between the input and output [6].

After the convolutional layers, the output is typically passed through one or more pooling layers, which down-sample the feature maps by taking the maximum or average value within a certain region of the map. This helps to reduce the spatial size of the feature maps and makes the model more efficient.

The output from the pooling layers is then passed through one or more fully connected layers, which are used to classify the input image into the desired categories. The final layer is typically a softmax layer, which converts the output of the previous layer into a probability distribution over the different categories. During training, the CNN model is fed with a large dataset of labeled images, and the weights of the filters are adjusted to minimize the difference between the predicted and actual labels. This is done using an optimization algorithm, such as stochastic gradient descent, which updates the weights based on the gradient of the loss function. Once the CNN model is trained, it can be used to classify new images of waste materials into their respective categories. This can be done by feeding the image through the CNN model and selecting the category with the highest probability value [3].

In summary, CNNs offer an efficient and accurate way to classify waste materials based on their images, which can contribute to more sustainable waste management practices.

The classified waste image is first stored in a MySQL database along with its corresponding latitude and longitude coordinates, which are obtained from the location where the image was captured. The database is set up to allow for efficient storage and retrieval of the image data. To retrieve the data from the MySQL database, an API is created in Flask, a Python web framework. The API allows for easy and secure communication between the database and the front-end application. It provides an endpoint that can be queried to obtain the image data and its associated location information.

Once the data is retrieved from the database using the API, it is displayed on a map using LeafletJS, a JavaScript library for creating interactive maps. The LeafletJS library allows for easy integration of maps into web applications and provides a range of features, such as zooming and panning, marker placement, and map layers. The LeafletJS map is customized to display the waste images with markers at their corresponding latitude and longitude coordinates. The markers are displayed with icons that represent the type of waste, such as plastic, paper, or glass. The map also provides an interface for waste collectors to view the waste collection routes optimized using Dijkstra's shortest path algorithm. Overall, this process allows for the efficient storage, retrieval, and visualization of waste image data, while also providing optimized routes for waste collection [7].

The use of modern technologies such as Flask API and LeafletJS for data management and visualization can also significantly improve the overall waste management process. The visual representation of waste data on maps can aid in identifying waste hotspots and identifying the best waste collection routes.

Incorporating waste management practices in our daily lives is crucial for a sustainable future. By utilizing these technologies and implementing efficient waste management practices, we can reduce the environmental impact of waste disposal and move towards a more sustainable and healthier planet.

2.2. Literature Review

It is crucial to develop human society based on natural conservation. Concepts of sustainable development between economy and environment can help to identify the balance point. Managing waste effectively can contribute towards conservation development. It is necessary incorporate economic concepts in the environmental project's evaluation and management. According to Brundtland Commission (WCED, 1987), one of the well-known definitions of sustainable development is sustainable development that meets the needs of the present without compromising the ability of future generations to meet their own needs [4]. It consists of two key concepts:

- The concept of needs, in particular the essential needs of the world's poor, to which overriding priority should be given.
- The idea of limitations imposed by the state of technology and social organization on the environment's ability to meet present and future needs.

Waste management has become a critical issue in modern times due to the increasing amount of waste generated by human activities. Effective waste management involves the proper segregation, collection, and disposal of waste to ensure a sustainable environment. In recent years, there has been a growing interest in applying machine learning and optimization techniques to waste management systems to improve efficiency and reduce the negative impact on the environment.

One of the primary challenges in waste management is the proper segregation of waste. Traditional waste segregation methods rely on human expertise, which can be time-consuming and error-prone. To address this issue, researchers have explored the use of computer vision techniques, particularly convolutional neural networks (CNNs), for automated waste classification.

CNNs have been shown to be effective in image recognition tasks, including waste classification. For example, Arroyo et al. (2019) developed a waste classification system using a CNN that achieved an accuracy of 92.5%. Similarly, Bonett et al. (2020) used a CNN to classify waste images into five classes (paper, plastic, glass, metal, and organic), achieving an accuracy of 97%. In addition, advancements in web-based technologies, such as APIs and mapping libraries, have enabled the development of more sophisticated waste management systems. For example, Elsawah et al. (2020) developed a web-based waste management system that integrates waste classification using CNNs and waste collection routing using Haversine algorithm. The system was designed to provide real-time information on waste collection routes and to enable efficient waste management [5].

Leaflet.js is a JavaScript library used for building interactive maps and for performing spatial analysis. It provides various built-in functions and plugins for optimizing routing and distance calculations on maps. One of the popular plugins used for routing optimization is the Leaflet Routing Machine plugin.

To calculate the shortest distance between two points on a map, the Haversine formula is used. It takes into account the curvature of the Earth and provides an accurate distance calculation between two points on the surface of the Earth. This formula is implemented in the Leaflet Routing Machine plugin to calculate the shortest route for the waste collector to collect the waste from different locations.

Approximate fuel calculation is another feature that can be added to the routing optimization process. This feature calculates the approximate fuel consumption based on the distance of the route and the type of vehicle used for waste collection. This helps to optimize the collection route by minimizing the fuel consumption and reducing the carbon footprint.

In summary, Leaflet.js provides a robust platform for waste management by integrating optimized routing and distance calculation features using the Haversine formula and approximate fuel consumption calculations. This helps to optimize the waste collection process, minimize fuel consumption, and reduce the carbon footprint.

2.3. Current System

The city of Amsterdam has implemented a waste collection system that uses an algorithm to optimize collection routes based on real-time data on waste volumes, traffic congestion, and weather conditions. The system also includes a mobile application for residents to report waste collection issues and receive notifications about collection schedules.

Their automated waste classifier takes in a 768×1024 image containing 2 or more pieces (objects) of waste on a white background. These pieces can be overlapping or non-overlaping and of different size on the white background. They fine-tuned a Faster R-CNN model pre-trained on PASCAL VOC dataset. Thier fine-tuned model will produce anchors (region proposals) and classify objects into three classes: landfill, recycling, and paper. Thier image dataset is generated by composing (stitching together) images in TrashNet dataset [8].

Class	Learning Rate(0.0012)	Batch RPN Size (128)	Batch Size (128)
Landfill	0.514	0.725	0.722
Paper	0.433	0.596	0.596
Recycling	0.585	0.767	0.767

Table 2.3.1: Validation AP scores from best hyperparameters tested

After running Faster R-CNN on the dataset using proposed data split, they achieved a mean Average Precision (mAP) of 0.683 overall on classification of the trash images.

CHAPTER 3: SYSTEM ANALYSIS

3.1. Requirement Analysis

The requirements analysis is broken down into two sections namely function requirements and non-functional requirements and they are discussed below.

3.1.1. Functional Requirement

- The user shall be able to upload a test image in a web interface.
- The user shall be able to know the recycle process of the classified data.
- The system shall be able to train the data and classify the test data.
- The system shall be able to store the classified data in the database.
- The admin shall be able to find the route of different bins from different locations.

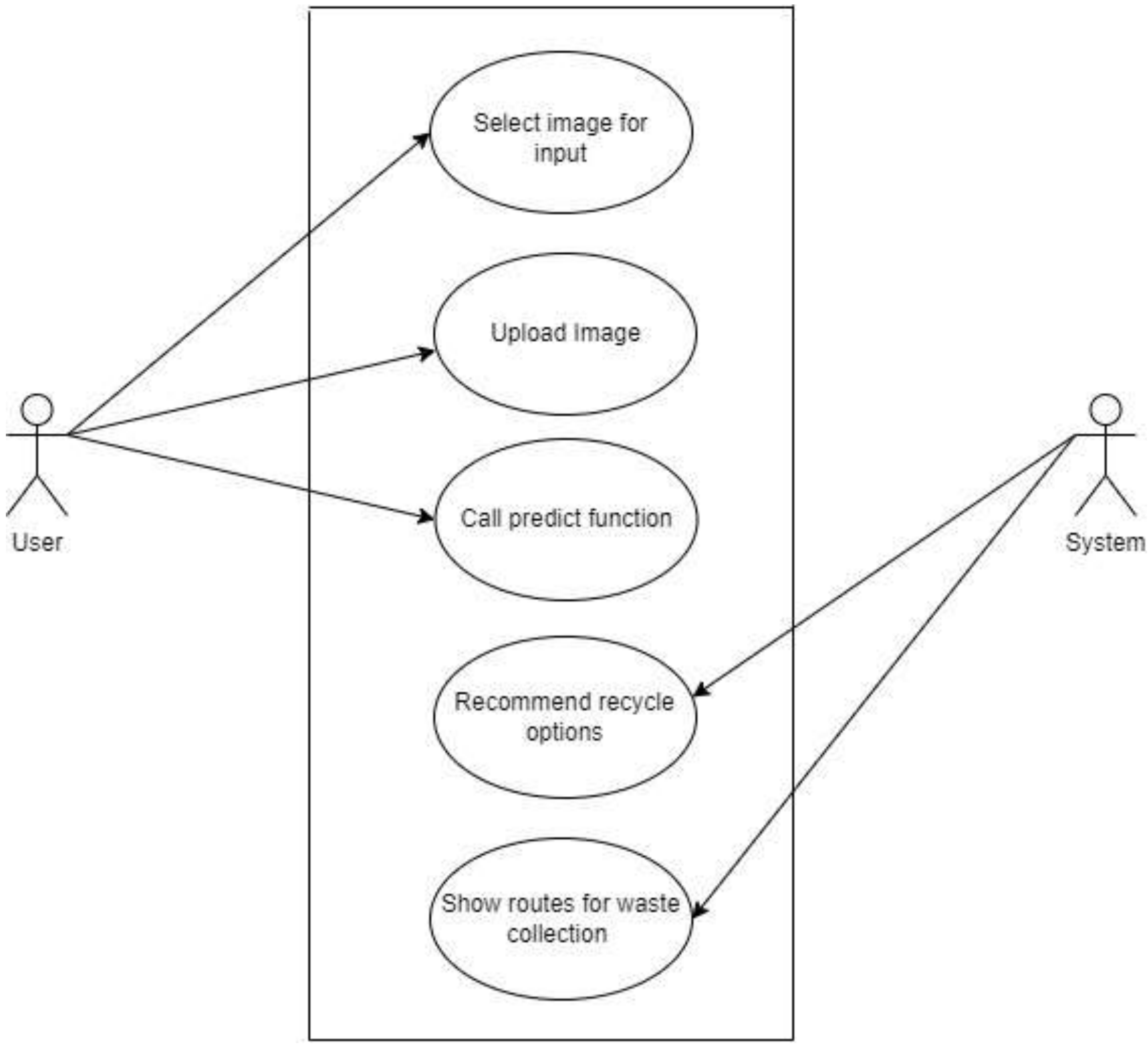


Figure 3.1.1.1: Use case diagram

In the above use case diagram, there are two actors on the scene, user and system. The user can select the image and provide it as an input to the system. The user can upload image of [jpeg, jpg, png] extension. After uploading an image user can click the classify button and call the predict function. On the other hand, system recommend user the recycle option according to classified waste image and display the result on the map of location of different bins and their waste fill percentage.

3.1.2. Non-functional requirement:

- The system must be able to augment data of small size images because of different orientation of the waste materials.
- The trained model must be saved on the local storage so that user can apply prediction method in it.
- The system must be able to provide high accuracy as possible.

3.1.3. Sequence Diagram

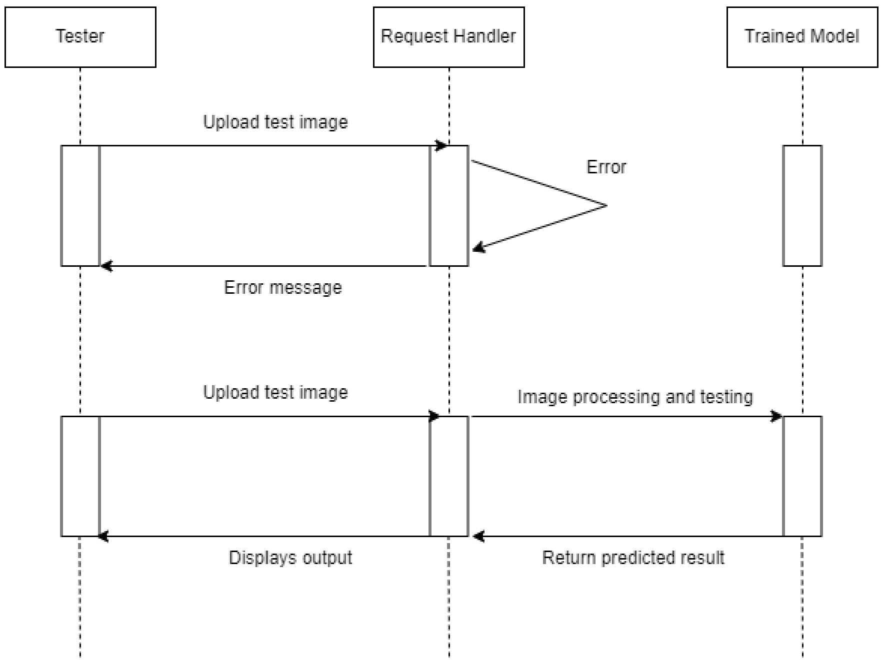


Figure 3.1.3.1: Sequence diagram

The process modeling of the system is described with the help of a sequence diagram. Here, a tester or and end user uploads a test image to the server. The request handler or the middle tier, acts as a middleware between the user input and the trained model. The request handler returns errors id any wrong input is fed to the system. If the upload is successful, then the image is processed and is tested against the trained model. Once the testing process is completed, the model returns the predicted result back and the result is displayed to the tester via web interface.

3.1.4. Activity Diagram

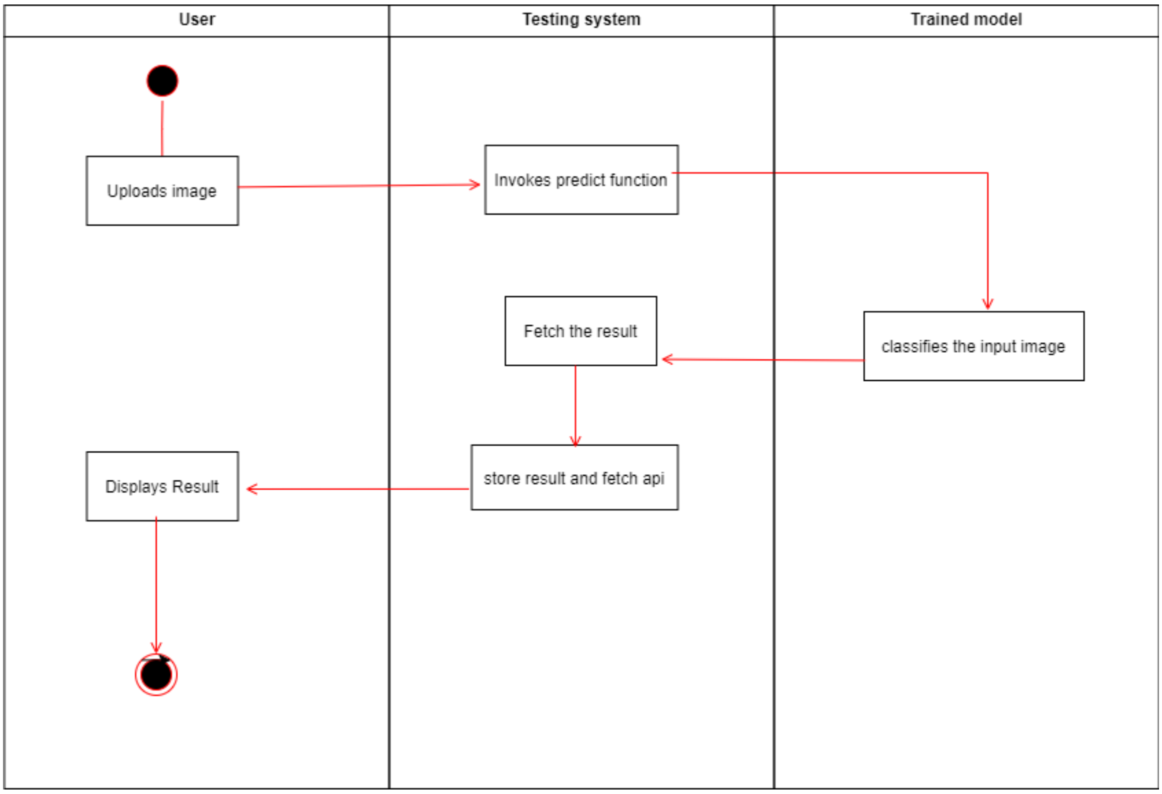


Figure:3.1.3.1.1: Activity diagram

The activity diagram explains all the activities that get carried out from the start till the end of the system. First, the user uploads a test image and class the predict function. Once the prediction function is called, the trained model is loaded and it classifies the input image and then stored those image along with user location in the database and then the api of stored is fetched and result is displayed on the map to the end user.

3.2. Feasibility study:

3.2.1. Technical feasibility:

Project technologies are open source. The application can be tested in any computing platforms. The software used for this project are:

- Python
- OpenCV
- Tensorflow

The hardware used is personal computer. Hence, we can say that this project is technically feasible.

3.2.2. perational feasibility:

The user is supposed to have a basic computing knowledge and experience in working with browsers and internet. The knowledge of machine learning could be a plus point.

3.2.1. Economic feasibility:

This project makes the use of software and web applications like OpenCV, Tensorflow, Jupyter network which are all open source or some free version are available. So, overall completion of this project does not require any cost which is the reason this project can be concluded as economically feasible.

3.2.2. Schedule feasibility:

A Gantt chart is used to do the timetable feasibility study. A list was made of all the critical tasks for this project and a timeline was prepared indicating that all of them could be accomplished by the deadline

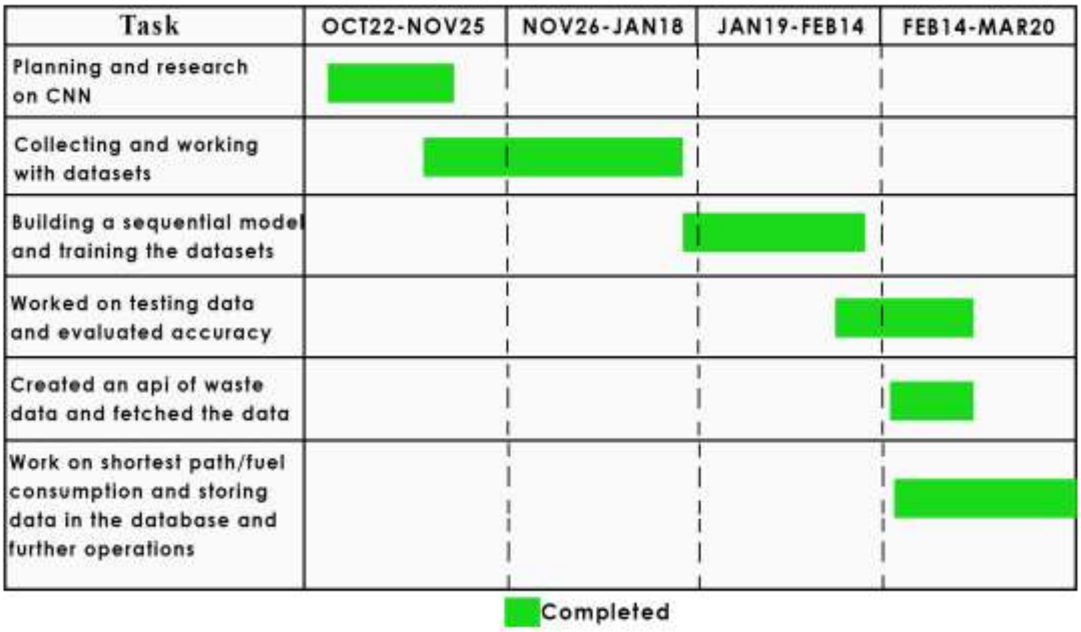
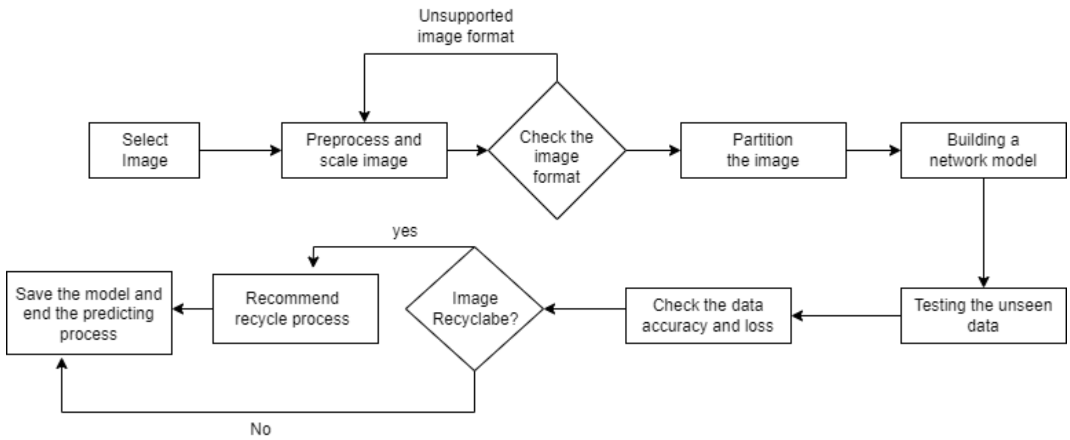


Figure 3.2.2.1: Gantt chart of the project

CHAPTER 4: SYSTEM DESIGN

4.1. Flow Diagram

- **Waste Classification**



- **Waste Collection**

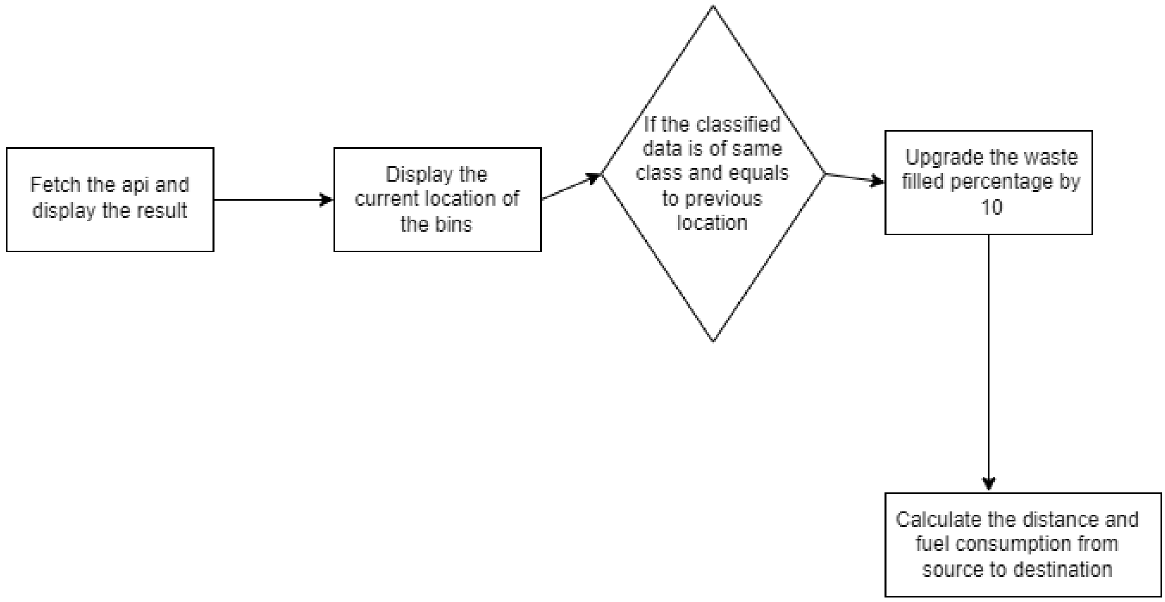


Figure 4.1.1: Flow Diagram for classification and optimization

The waste classification process using CNN involves collecting a large dataset of waste images and manually classifying them into different categories such as plastic, paper, glass, metal, and organic waste. The dataset is then divided into two parts: the training set and the testing set. The training set is used to train the CNN model to recognize the patterns and features of each category of waste.

Once the model is trained, it is evaluated using the testing set to measure its accuracy and performance. The user can then upload an image of waste to the system, and the model is used to classify which category the waste belongs to. If the uploaded image does not match the acceptable file extension, an error message is displayed.

After classification, the information about the classified waste, along with the latitude and longitude of the location, is stored in a database using MySQL. The data is then fetched using an API in Flask and displayed on a map using Leaflet JS.

The waste collection system in the application involves displaying the locations of different waste bins on the map, and the optimized routing algorithm using the Haversine formula is used to find the shortest path between the waste bins for efficient waste collection. The approximate fuel cost for the route is also calculated to minimize costs.

Overall, this system aims to provide an efficient and optimized waste management solution by using advanced technologies such as CNN for waste classification and Leaflet JS for waste collection route optimization. It has the potential to significantly improve waste management systems and promote a cleaner and more sustainable environment.

4.1.1. Component Diagram

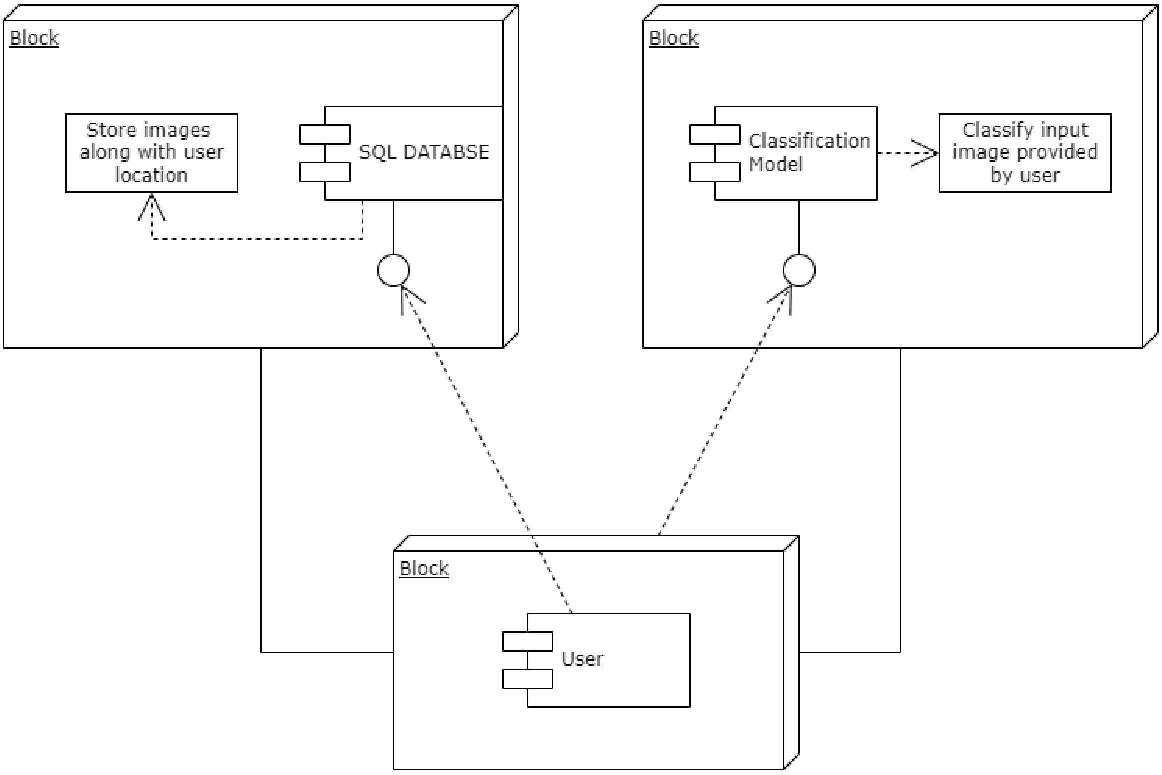


Figure 4.1.2.1: Component Diagram

The component diagram for the waste classification and optimized waste collection system can be divided into three main components:

- Image Classification Component: This component is responsible for classifying the images of waste into different categories such as plastic, paper, glass, metal, and organic using CNN. This component takes the input image, preprocesses it, and passes it through the CNN model. The output of this component is the waste category label.
- Routing and Collection Component: This component is responsible for optimizing the collection of waste from different locations. It takes the location of the waste bins and the location of the waste collection vehicles as inputs and uses the Dijkstra shortest path algorithm to calculate the optimal route for waste collection. This component also calculates the approximate fuel cost for the collection of waste using the Haversine formula.

- Database and Map Display Component: This component is responsible for storing the waste classification and location data in a MySQL database. It also fetches this data using an API in Flask and displays it on a map using Leaflet JS. This component enables the waste collection team to view the location of waste bins and the optimal route for waste collection on a map.

Overall, the component diagram illustrates the high-level architecture of the waste classification and optimized waste collection system, which consists of multiple components that work together to achieve the system's overall functionality.

4.2. Algorithm used:

i. CNN algorithm:

Convolutional Neural Network, also known as convnets or CNN, is a well-known method in computer vision applications. It is a class of deep neural networks that are used to analyze visual imagery. This type of architecture is dominant to recognize objects from a picture or video. It is used in applications like image or video recognition, neural language processing, etc.

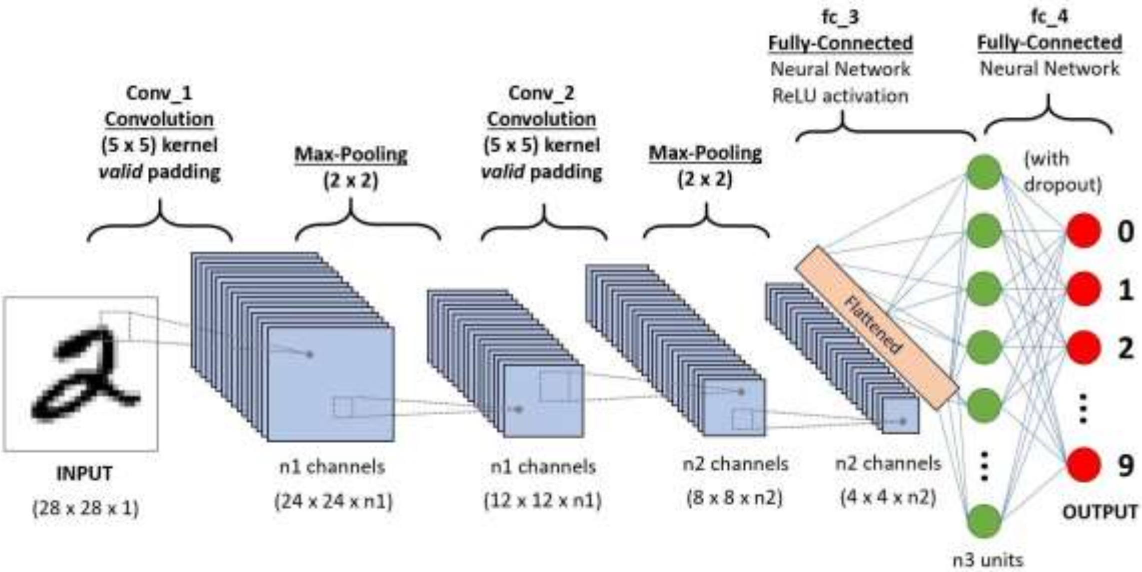


Figure 4.4.i.1: CNN Architecture

CNNs consist of multiple layers, each with a specific function. The first layer is typically a convolutional layer, which applies a set of filters to the input image. Each filter is a small matrix of weights that is convolved with the input image, resulting in a set of feature maps that highlight different aspects of the image, such as edges, corners, or textures. The output from the convolutional layer is passed through a non-linear activation function, such as ReLU (Rectified Linear Unit), which introduces non-linearity into the model and helps to capture complex relationships between the input and output.

After the convolutional layers, the output is typically passed through one or more pooling layers, which down-sample the feature maps by taking the maximum or average value within a certain region of the map. This helps to reduce the spatial size of the feature maps and makes the model more efficient.

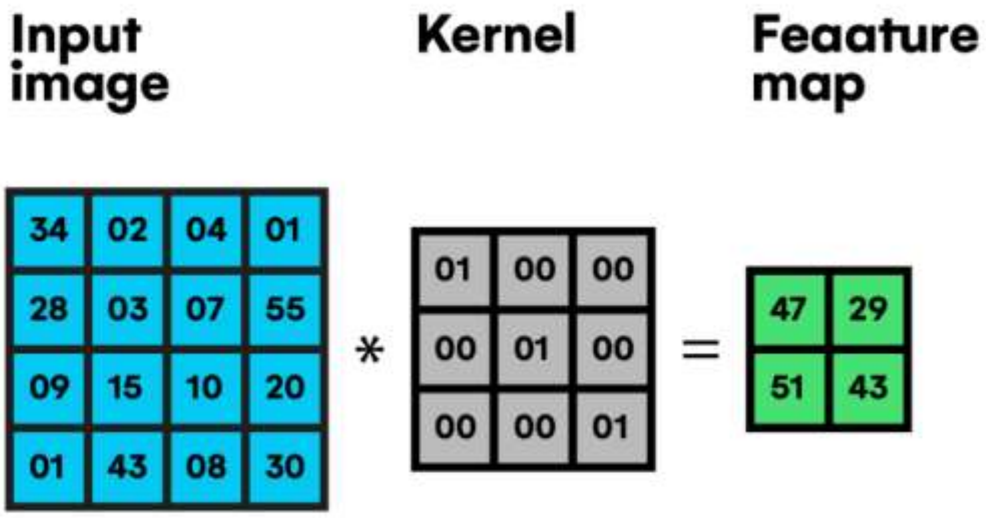


Figure 4.4.i.2: CNN feature extraction

The output from the pooling layers is then passed through one or more fully connected layers, which are used to classify the input image into the desired categories. The final layer is typically a softmax layer, which converts the output of the previous layer into a probability distribution over the different categories.

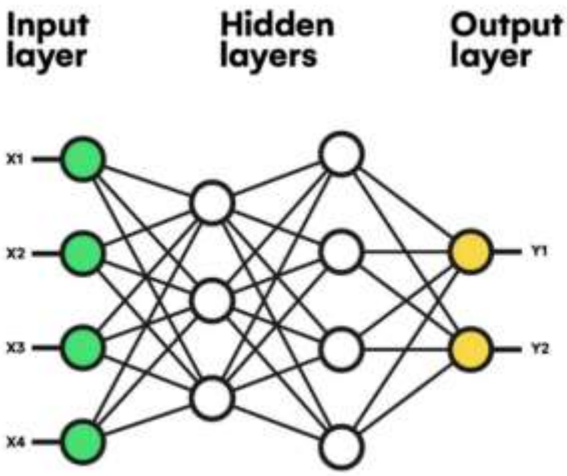


Figure 4.4.i.3: Hidden layer architecture

During training, the CNN model is fed with a large dataset of labeled images, and the weights of the filters are adjusted to minimize the difference between the predicted and actual labels. This is done using an optimization algorithm, such as stochastic gradient descent, which updates the weights based on the gradient of the loss function. Once the CNN model is trained, it can be used to classify new images of waste materials into their respective categories. This can be done by feeding the image through the CNN model and selecting the category with the highest probability value [3].

In summary, CNNs offer an efficient and accurate way to classify waste materials based on their images, which can contribute to more sustainable waste management practices.

ii. Haversine algorithm

The Haversine algorithm is a mathematical formula used to calculate the distance between two points on the surface of a sphere, such as the Earth. It takes into account the curvature of the sphere and provides a more accurate calculation of distance than the simple Pythagorean formula used for flat surfaces.

The formula is as follows:

d = 2r * asin(sqrt(sin^2((lat2-lat1)/2) + cos(lat1) * cos(lat2) * sin^2((lon2-lon1)/2)))

Where:

d: distance between the two points in units of the radius of the sphere (usually kilometers or miles).

r: radius of the sphere (usually the radius of the Earth).

lat1, lat2: latitude of the two points in radians.

lon1, lon2: longitude of the two points in radians.

The Haversine formula assumes a perfect, spherical Earth, which is not accurate. For more precise calculations, additional corrections may be necessary, such as the use of the Vincenty formula or an ellipsoidal model.

Once we have the latitude and longitude coordinates for the two points, we can use the Haversine formula to calculate the distance between them. The result will be the shortest distance between the two points on the surface of the sphere. The Haversine algorithm is widely used in geospatial applications, such as GPS navigation, mapping, and location-based services. It is particularly useful when calculating distances over large distances, such as between cities or even continents.

haversine(d/r) = haversine(Φ2-Φ1) + cos(Φ1)cos(Φ2)haversine(λ2-λ1)

where:

- d is the distance between the two points on the sphere
- r is the radius of the sphere
- Φ1 and Φ2 are the latitudes of the two points (in radians)
- λ1 and λ2 are the longitudes of the two points (in radians)
- haversine is a trigonometric function, defined as haversine(x) = $\sin^2(x/2)$

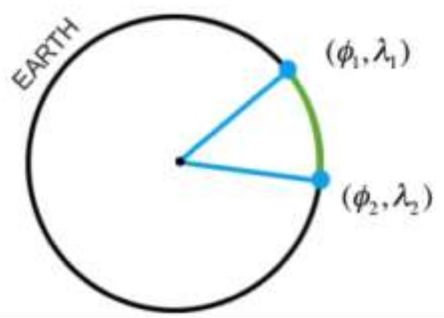


Figure 4.4.ii.1: Haversine algorithm

```
let markersList = [];
function distance(lat1, lon1, lat2, lon2) {

    const radlat1 = (Math.PI * lat1) / 180;
    const radlat2 = (Math.PI * lat2) / 180;
    const radlon1 = (Math.PI * lon1) / 180;
    const radlon2 = (Math.PI * lon2) / 180;


    const theta = lon1 - lon2;
    const radtheta = (Math.PI * theta) / 180;
    let dist =
        Math.sin(radlat1) * Math.sin(radlat2) +
        Math.cos(radlat1) * Math.cos(radlat2) * Math.cos(radtheta);
    dist = Math.acos(dist);
    dist = (dist * 180) / Math.PI;
    dist = dist * 60 * 1.1515 * 1.609344;
    return dist;
}
```

Figure 4.4.1: Code for calculating distnace using haversine

```
In [13]: num_classes = 5

model = Sequential([
  layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes,activation='softmax')
])

In [14]: model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

In [15]: epochs=10
history = model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=epochs
)

Epoch 1/10
```

Figure 4.4.2: Code for sequential building using cnn

CHAPTER 5: IMPLEMENTATION

5.1. Implementation

The software implementation of waste classification and optimized routing for waste collection typically involves several components and technologies, including machine learning algorithms, leafletjs, and used flask api of generating data of different types which includes waste categories, waste filled percentage, latitude and longitude.

Here are some of the key steps involved in the software implementation of waste classification and optimized routing for waste collection:

- Data collection and preprocessing: This involves collecting data on the waste generation patterns, waste types, and other relevant factors. The data may be obtained through various sources such as sensors, cameras, or manual reporting. The data is then preprocessed to remove noise and inconsistencies.
- Waste classification: Machine learning algorithms such as Convolutional Neural Networks (CNN) can be used to classify waste into different types such as organic waste, plastic waste, and paper waste. The algorithms can be trained on a large dataset of waste images and can learn to recognize the different types of waste.
  The model was implemented using keras, an open source python library for machine learning which uses TensorFlow as the backend. Keras version 2.3.8 was used and Tensorflow version 2.11.0 was used.

Convolutional Neural Network

A 1 layer CNN was implemented to build the model. Each layer work as:

- Layer 0: Input image of size 300*300
- Layer 1: Convolution with 96 filters, stride 4, padding 2, activation ‘relu’
- Layer 2: Max pooling with a size 3*3, pool size 2
- Layer 3: Convolution with 64 filters, stride 1, padding 2, activation ‘relu’
- Layer 4: Max pooling with a size 3*3, pool size 2
- Layer 5: Convolution with 32 filters, stride 1, padding 2, activation ‘relu’
- Layer 6: Max pooling with a size 3*3, pool size 2
- Layer 7: Convolution with 64 filters, stride 1, padding 2, activation ‘relu’

- Layer 8: Max pooling with a size 3*3, pool size 2
- Layer 9: Fully connected with 4096 neurons
- Layer 10: Fully connected with 5 nuerons

The classification model was trained with the following hyperparameters.

Hpyerparameters	Value
Learning Rate	0.7
Optimizer	Adam
Epochs	20
Image Shape	300*300*3
Activation	Relu
Loss	Categorical Cross entropy

Table 5.1.1. Hyperparameters

During the implementation process, all the data images were preprocessed before they were fed to the algorithm for training. To generate the required train and test images, following parameters were tuned.

To generate training and testing images:

Properties	Value
Horizontal flip	True
Vertical flip	True
Validation split	0.2
Rescale	1/255

Table 5.1.2. Image Augmentation

Out of 5143 images, 80% of the images were used for training and the remaining 20% were used to test the model.

- Optimized routing: Algorithms such as Dijkstra's algorithm can be used to find the shortest path between collection points, taking into account factors such as the location and type of waste, traffic congestion, and the availability of collection trucks. The optimized routing algorithm can be integrated with the GIS to provide real-time updates on the collection routes.

5.1.1 Tools Used

CASE tools:

- Draw.io to design the UML diagrams and other block diagrams.

Client side:

- HTML to design the user interface.
- CSS and Bootstrap to beautify the user interface.
- JavaScript(jquery) to make the user interface interactive and the machine learning model.

Server Side:

- Python is used as the server-side programming language.
- A python framework, Flask is used to implement the user interface. The framework starts the web application, takes the user request and displays the output to the webpage.
- CNN algorithm is used to classify and the input image and mysql is used to store the image.
- Leafletjs is used to display the results on the map.

Hardware:

- The classifier model was trained and tested on a personal computer having the following specs:
- GPU: Nvidia GTX 1050ti, 4GB DDR5 VRAM
- CPU: Intter core i7-9795H, 12 core CPU @ 2.4 GHz clock speed
- RAM: 16GB DDR5
- OS: Windows 11, professional edition

- Open cv2: OpenCV (Open Source Computer Vision Library) is a popular open-source computer vision library that can be used for image classification tasks. It provides a set of tools and algorithms for image processing, including feature extraction, image segmentation, and object detection.
- TensorFlow: It is an open-source machine learning framework that provides a set of tools and APIs for building and training deep neural networks. It can be used for a wide range of machine learning tasks, including image classification.
- Python: It is a high-level programming language that is widely used in various fields, including data science, web development, and artificial intelligence. It is known for its simplicity and ease of use, making it an ideal language for beginners and experienced programmers alike.

- Leaflet: It is an open-source JavaScript library for creating interactive maps. It provides a simple, lightweight and flexible framework for mapping applications, with a range of features including markers, popups, and tile layers. Leaflet is widely used for building web-based mapping applications, and is compatible with most modern web browsers and mobile devices.

5.2. Testing

5.2.1. Test Cases for System Testing

Table 5.2.1.1: System testing test cases

Test Case	Description	Expected Result	Actual Result	Remarks
TU01	Data validation test whether the system is validating user input correctly.	The system should not accept invalid input data and should display appropriate error messages.	The system rejects invalid input data and displays error messages.	PASS
TU02	Performance test whether the system is performing optimally under high load.	The system should be able to handle a large number of users simultaneously without experiencing performance issues.	The system is able to handle a large number of users without experiencing any significant performance issues.	PASS
TU03	Compatibility testing to Test whether the system is compatible with different operating systems and web browsers.	The system should be accessible and function properly across different platforms and browsers.	The system does not work correctly across different operating systems and web browsers.	FAIL
TU04	Route calculation test whether the routing functionality of the Leaflet library is working correctly.	The library should calculate the optimal route between two or more locations and display it on the map.	The library correctly calculates the route and displays it on the map.	PASS

TU05	Route customization test whether the routing functionality of the Leaflet library can be customized to meet specific requirements.	The library should allow for customization of the route based on user preferences, such as avoiding toll roads or highways.	The last entered username appeared in main menu	PASS

5.2.2. Test Cases for Unit Testing

Table 5.2.2.1: Unit testing test cases

Test Case	Description	Expected Result	Actual Result	Remarks
TU01	When user clicks on the single waste classification button.	The user should redirect to single waste classify page.	The waste was redirected to the page.	PASS
TU02	When user clicks on the multiple waste classification button.	The user should redirect to multiple waste classify page.	The waste was redirected to the page.	PASS
TU03	When user clicks on the classify waste button.	The waste should be classified according to its class.	The waste was classified according to its class	PASS
TU04	When user tries to upload to image with the extension that is not [jpeg,jpg,png]	The system must show an image extension error.	The system showed an image extension error.	PASS
TU05	When the admin clicks on any marker on the map.	The map should show the shortest direction and	The map shows the direction, distance, time and	PASS

		approx. fuel consumption to collect the waste.	approximate fuel consumption.	
TU06	When the admin clicks on any marker on the map.	Map should show the time to travel and fuel according to distance, traffic condition, road condition.	Map should show the time to travel and fuel according to distance only.	FAIL
TU07	When the user clicks on the multiple image classification button	The system should display percentage of multiple waste in the image.	The system displayed the percentage of multiple waste	PASS
TU08	When the waste is collected from different location from the bin.	The bin should be empty and the waste percentage should be zero.	The bin percentage are unchanged.	FAIL

CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION

6.1. Conclusion

In conclusion, the waste classification and optimized routing system developed using Leaflet JS has the potential to significantly improve waste management practices. The system is capable of accurately classifying different types of waste and routing them to appropriate destinations, reducing the amount of waste that ends up in landfills and promoting more sustainable waste management practices. The use of Leaflet JS also provides a user-friendly and visually appealing interface, making it easy for waste management personnel to use and understand.

6.2. Limitations

There are may rooms for improvements for the model. The model still is not perfect in classifying the garbage with 100% accuracy. Some of the limitations are:

- The machine learning model required high computing power to train.
- The system is only good at the testing model it uses.

Apart from these limitations, there are plenty of further enhancements that could be done on the future. Since the prototype for garbage classification is set, rather it can be done for real time garbage classification with computer vision.

6.3. Future Recommendation

While the waste classification and optimized routing system developed in this project is a significant step towards better waste management practices, there is still room for improvement. Here are some recommendations for future research and development:

- Incorporating real-time data: One limitation of the current system is that it relies on static data. Incorporating real-time data, such as traffic conditions and current waste levels at different destinations, could improve routing accuracy and efficiency.

- Integration with other waste management systems: The current system operates independently of other waste management systems. Integrating it with other systems, such as waste collection or disposal systems, could create a more comprehensive waste management solution.
- Testing and validation: While the system was tested in simulated environments, further testing and validation in real-world scenarios could provide additional insights into its performance and effectiveness.

By incorporating these recommendations, the waste classification and optimized routing system developed in this project could become an even more effective tool for improving waste management practices and promoting sustainable waste management.

REFERENCES:

[1] News, N.*1 million tonnes of garbage produced per year in Nepal, nepalnews*. Available at: https://nepalnews.com/s/nation/1-million-tonnes-of-garbage-produced-per-year-in-nepal.

[2] Authors, A. and Kasian Myagila & Hassan Kilavo.*A comparative study on performance of SVM and CNN in Tanzania sign language translation using imagerecognition*,*Taylor&Francis*.Availableat:https://www.tandfonline.com/doi/full/10.1080/08839514.2021.2005297.

[3] Solla, D.G. (2022) *Advanced Waste Classification with machine learning*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/advanced-waste-classification-with-machinelearning-6445bff1304f

[4] *Guidance on solid waste and health World Health Organization*. World Health Organization. Available at: https://www.who.int/tools/compendium-on-health-and-environment/solid-waste.

[5] *World Health Organization*. World Health Organization. Available at: https://www.who.int/tools/compendium-on-health-and-environment/solid-waste.

[6] Bányai, T. *et al.* (2019) *Optimization of municipal waste collection routing:Impact of industry 4.0 technologies on environmental awareness and Sustainability*, *International journal of environmental research and public health*. U.S. National Library of Medicine. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6406842/.

[7] *ML Practicum: Image Classification | machine learning | google developers* (no date) *Google*. Google. Available at: https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks (Accessed: March 20, 2023).

[8] *Machine learning | course | stanford online*. Available at: https://online.stanford.edu/courses/cs229-machine-learning (Accessed: March 20, 2023).

APPENDIX

Figure i: Landing page

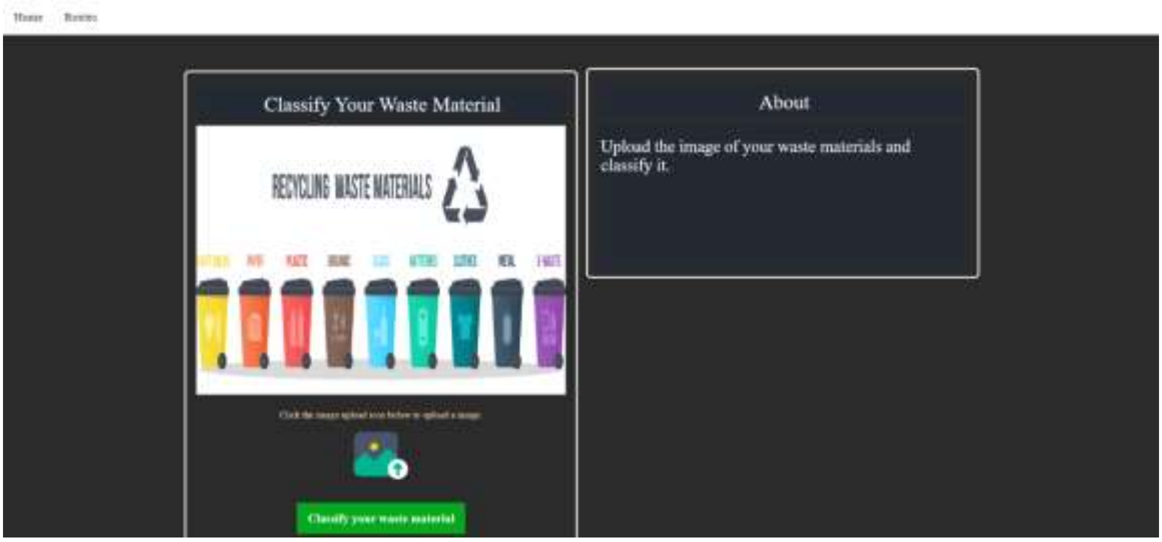


Figure ii: Classification page

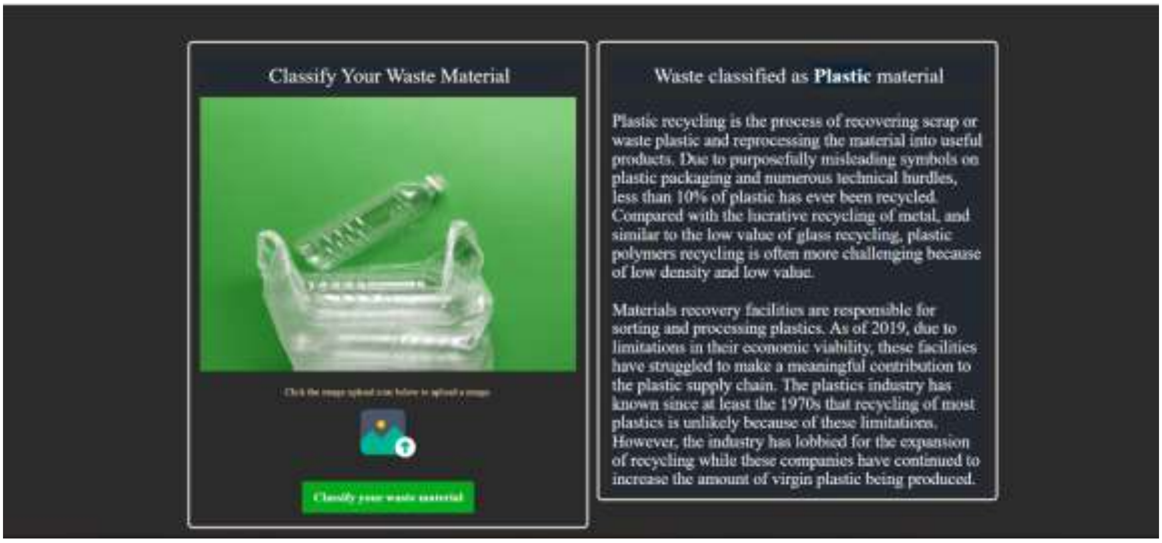


Figure iii: Classification of image of class plastic

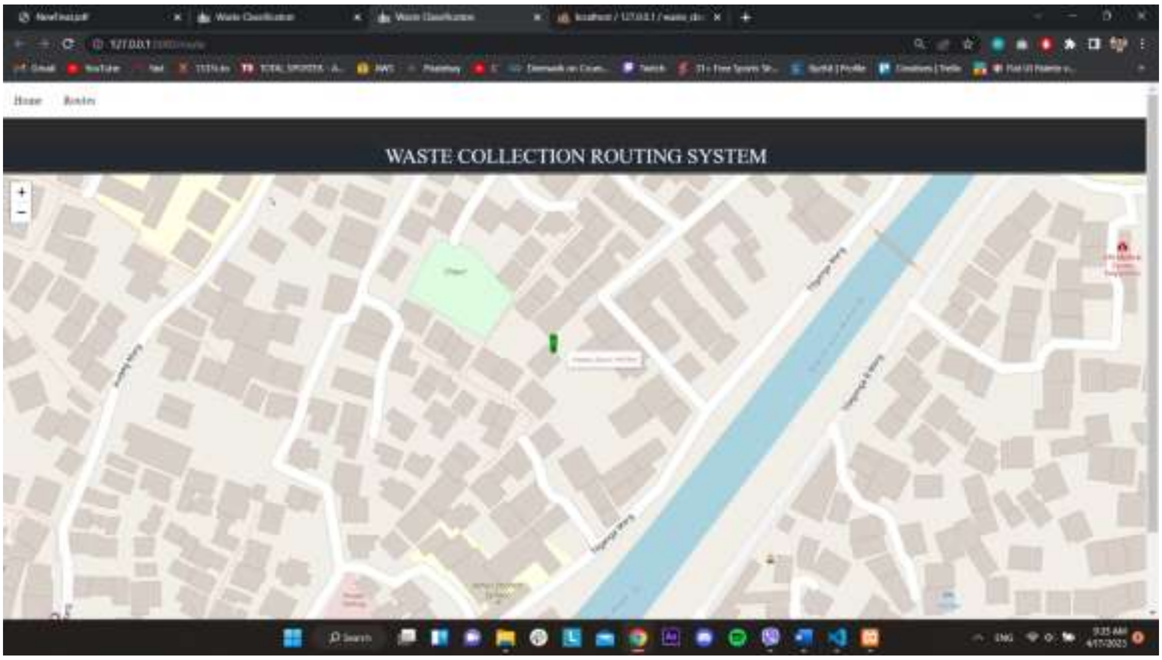


Figure iv: Organic Waste Filled Percentage

Figure v: Routing optimization with shortest distance

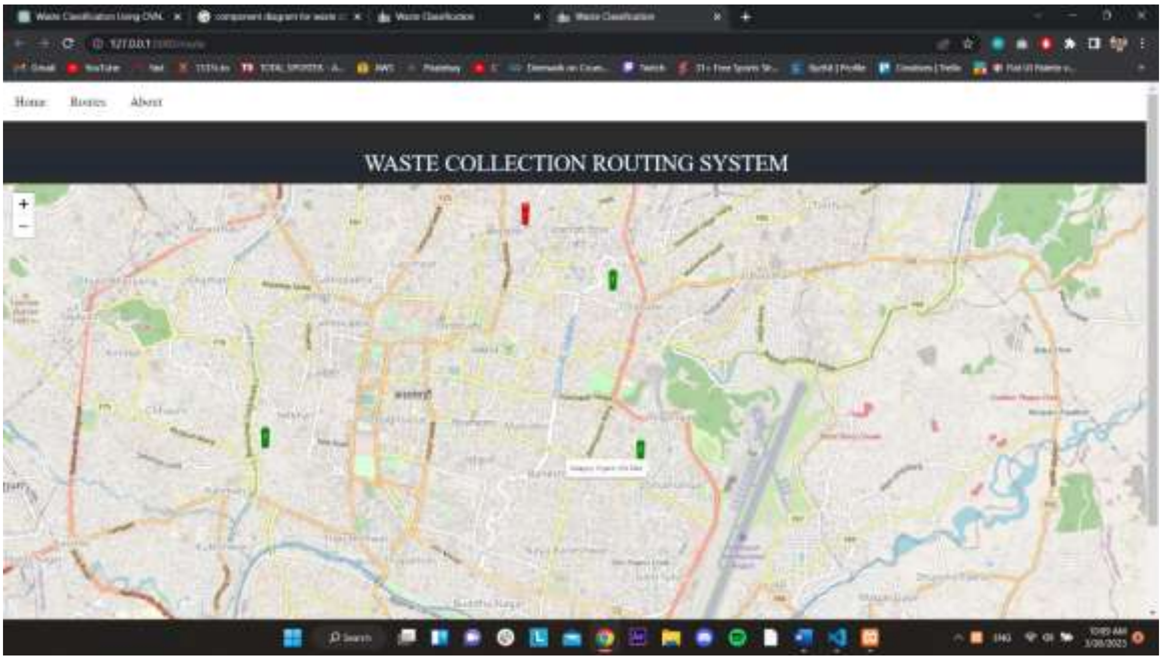


Figure vi: Waste fill percentage of the bin

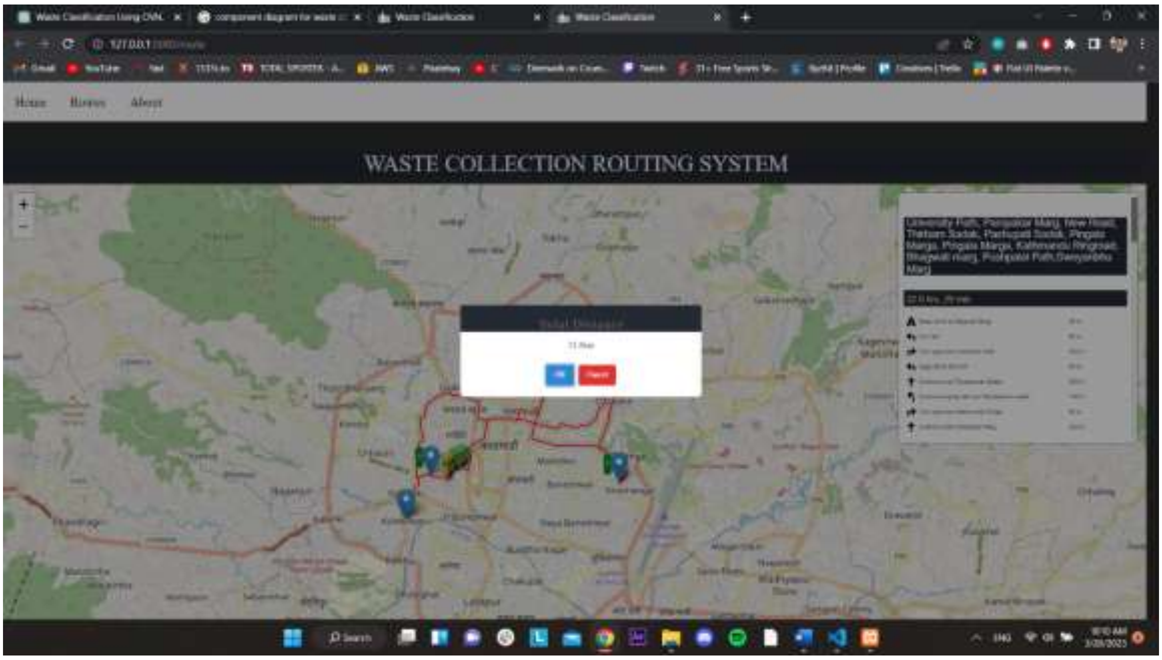


Figure vii: Total distance to be covered to collect bin across the city

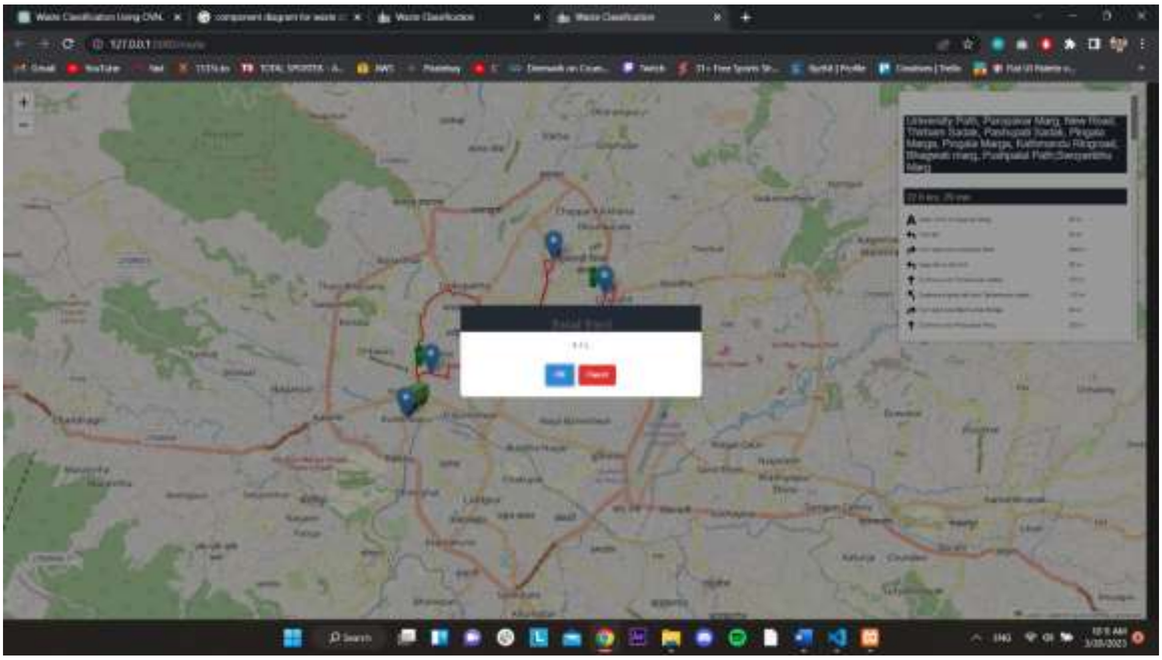


Figure viii: Approximate fuel consumption to collect bin across the city