An Intelligent Garbage Classification Motivator

Intelligent Reasoning Systems Practice Module

Team Members

YUE Yiran

GUO Rui

LIN Zhengxi

ZOU Yufan

Contents

1. Introduction

1.1 Background

Nowadays, all the social members, from presidents to residents, acknowledge the impertinence of sustainable development than ever before. As such, the transition to net zero greenhouse gas emissions must occur as soon as feasibly possible. Cities can do their part by embracing a wide range of options. Singapore government also try its best to create a carbon and climate resilient future. Transformations are happening everywhere, neither is huge or small in industry, economy and society such as more renewable energy, greater energy efficiency, reducing energy consumption and so on. Improving energy efficiency is a key focus of our efforts to reduce emissions, cited in Long-Term Low-Emissions Development Strategy (LEDS) of Singapore.

Fig 1.1 SG Action for Sustainable Development and Recycling

Many processes of waste treatment such as fermentation, transportation, incineration and landfill are important sources of anthropogenic greenhouse gas emissions, which also aggravates the scarcity of land resources (IPCC). So, there are significant and reasonable we need to enhance our waste management system for a low-carbon and green future. Inevitable requirement of urban sustainable development. It can reduce the amount of waste incineration and landfill, improve the recyclability of garbage, promote a circular economy and green development. Finally, all these efforts lead to a much lower carbon emission and environmental pollution.

1.2 Solution

To improve the garbage collection rate, a solution is put forward, solving the problem

from the source. We propose an Intelligent Garbage Classification Motivator, in order to encourage citizens to throw-in the garbage correctly as its' classifications.

Fig 1.2 *EZSORT* LOGO

The picture above is our logo. The team's name, EZSORT, means it is easy for people to sort the garbage correctly, which is an appeal and encouragement for garbage classification. The pattern is based on green, which represents the team's green, sustainable and recyclable vision. The circular pattern symbolizes the earth, the dots above represent white clouds, and the waves below represent the ocean. The logo reflects many social meanings of ***EZSORT***.

EZSORT is committed to influencing and making difference at the source of waste collecting. The system is based on the exist trash cans using an embedded design. In this way, rich functions can be realized with low cost. We will make full use of existing infrastructure, public resources and market education foundation.

EZSORT is easy to be used with high accuracy. When people put garbage into the dustbin, the sound sensor in the system will obtain the sound of garbage falling and the machine learning model we developed will automatically recognize the type of garbage. The model is based on sound's MFCC features extracting and HMM (Hidden Markov Model). Then the intelligent reasoning and interaction will be done.

EZSORT has vivid and inspiring interaction. The system interacts visually with users through the screen, and feedback the environmental benefits brought by correct classification, so as to achieve the purpose of encouraging citizens to correctly classify and throw domestic garbage.

We hope ***EZSORT*** can be the platform to rise the people awareness and provide a

window to allow them to step out. We believe this small action can be a big step tomorrow.

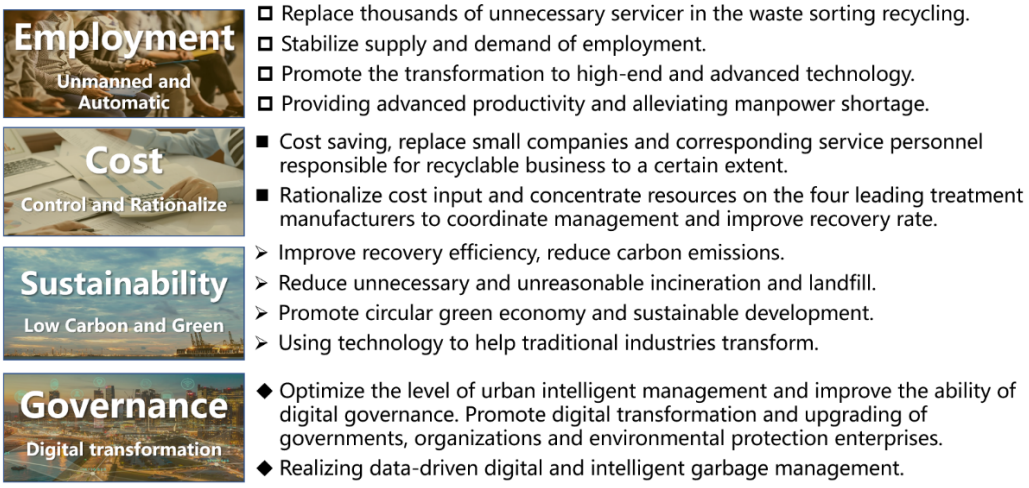


Fig 1.3 Social Meanings: Merit Lies in Present and Future

2. Problem Description and Market

2.1 Problem Description

Singapore's overall waste recycling rate increased to 55% in 2021 from 52% in 2020 but is still below the pre-epidemic level of 59% in 2019. However, Singapore's goal is to achieve a 70% waste recycling rate by 2030 which is still a long way to run. More needs to be done to improve the recycling rate of Plastic waste, Paper/Cardboard, glass, as they remain low.

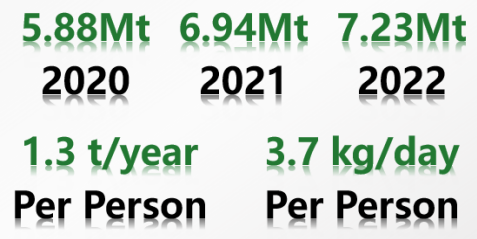


Fig 2.1 Garbage Production in Singapore

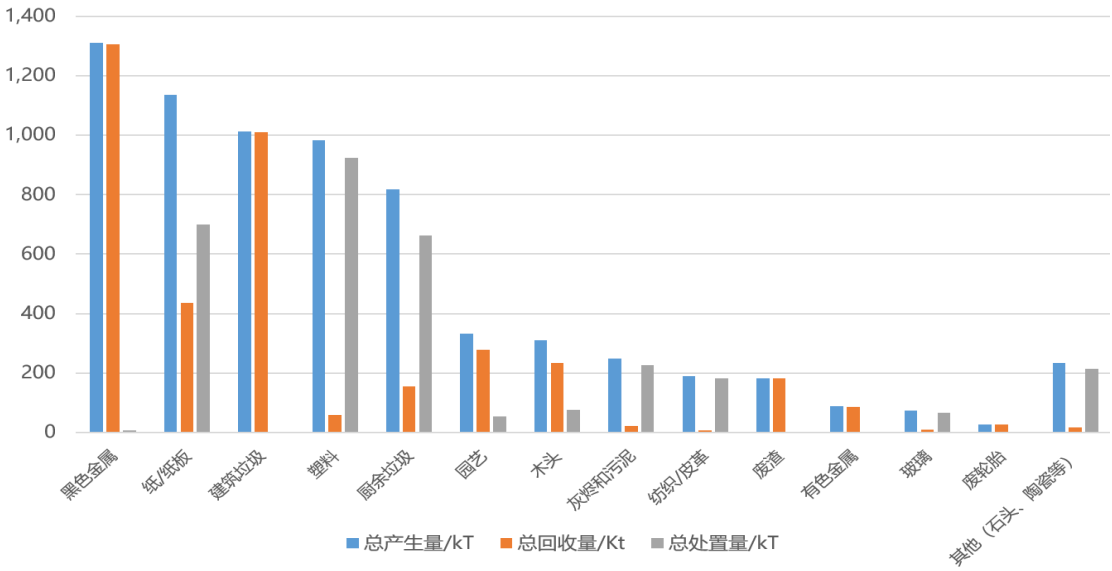


Fig 2.2 2021 Waste Statistics and Overall Recycling in Singapore

Although the Singapore government has adopted many kinds of garbage disposal strategies, the garbage recovery rate has not been significantly improved and the production of garbage has been increasing significantly.

Fig 2.3 Current Solid Waste Treatment Methods in Singapore

The low garbage recovery rate could bring about a series of serious social problems. Firstly, **Excessive human resource occupancy and property investment**. A large number of Workers have to spend much time and effort on manual classification from tons of garbage in bad situations. Secondly, **Large amount of unnecessary exhaust emissions.** Carbon emission from excessive transfer, incineration and landfill activities. Unreasonable classification of valuable wastes leads to incineration or landfill.

2.2 Market Analysis

We realize that the low garbage recovery rate may be caused by two factors: garbage **collection** and **disposal**. In the process of garbage disposal, a large number of workers are required to manually sort complex garbage, including a lot of garbage from Recyclable Waste Bins. Besides, recyclable items are easily contaminated during transportation. It is easy and often polluted by other wastes such as kitchen waste during collection and transfer. Additionally, some people throw non-recyclable rubbish to the recyclable ashbin by mistake or they cannot figure out the correct kind of rubbish into put in which makes a large number of recyclables lose their recycling value.

Therefore, we conducted a much more in-depth behavioral survey on users. There are five questions in the questionnaire. We surveyed more than 50 people around us, including young people, middle-aged people and the elderly. Most of them engage in

education-related occupations (including teachers and students, 57%), which means they have a relatively stable and representative lifestyle.

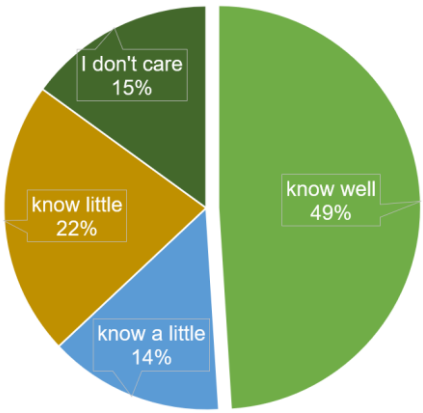


Fig 2.4 Do You Understand the Garbage Classification Requirements

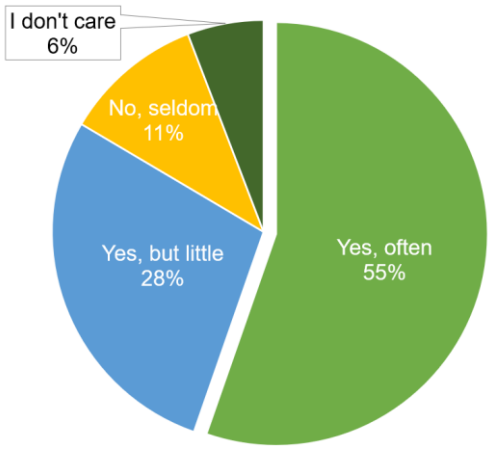


Fig 2.5 Have You Learned about the Publicity of Garbage Classification

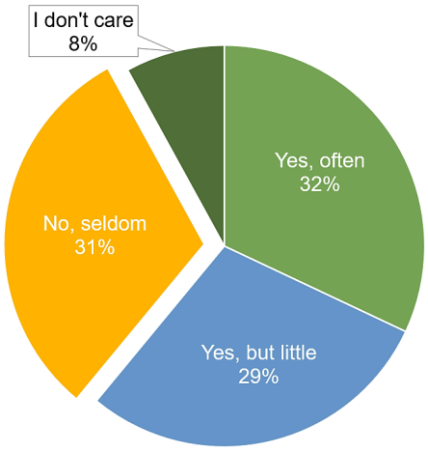


Fig 2.6 Do You Pay Special Attention to Garbage Classification in Your Life

The above 3 figures show the answers to the three core questions. Based on our survey and interview, we can infer a reliable conclusion that the process of people dropping

garbage which has not achieved correct classification is the source of the low-recovery-rate problem. We used to educate citizens to classify their garbage before throw-in. But the outcome is poor, which leads the bad influence mentioned above. So, we hope to solve the problem of domestic waste classification from the source to improve the recovery rate, reduce the pressure of manual classification sorting and labor, physical, land and financial demands.

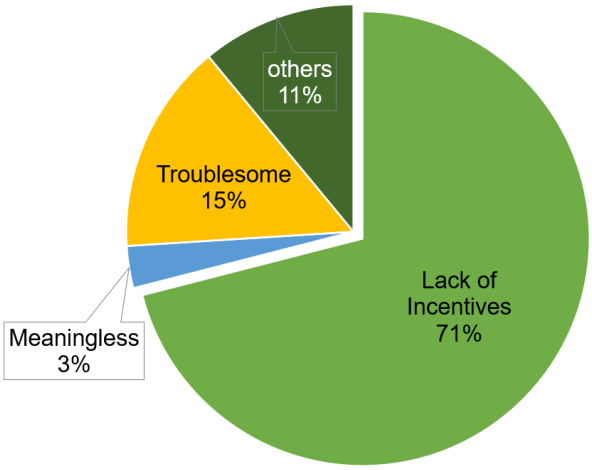


Fig 2.7 Reasons for NOT Classifying

Therefore, ***EZSORT*** sees the need a set of intelligent and digital solutions to deal with domestic waste collection, which can encourage people to put garbage correctly.

3. System Modeling and Algorithm

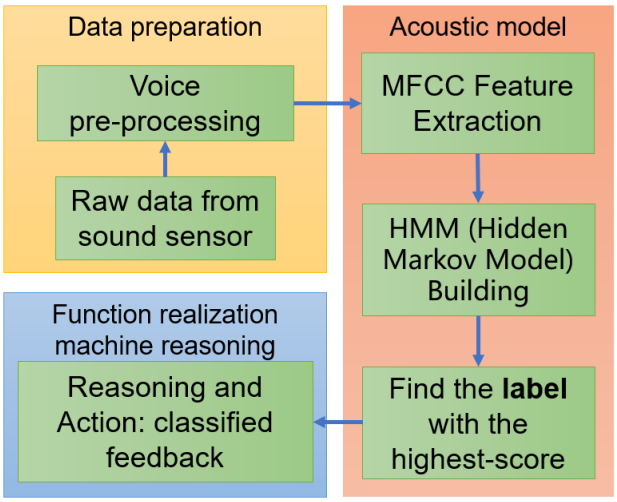


Fig 3.1 Technical Roadmap of *EZSORT*

The whole model system consists of three parts, **data preparation**, **acoustic model** and **function realization with machine reasoning**.

3.1 Data Preparation

The data preparation of the system includes two parts: training set (knowledge base) data and input data.

3.1.1 Knowledge Base

Our training dataset, an all-self-built dataset, is composed of more than ten kinds of garbage falling sound with different materials and sizes. These sound knowledges are stored in the back-end database in the form of *.wav* files. All sounds are divided into three categories: metal, plastic and carton, which is our classification label. These labels are known for the model because our machine learning model is based on supervised HMM. The following table shows details of our training datasets.

Table 3.1 Details of Datasets

Dataset Characteristics		Information
Total Number of Data		Almost 150, metal(17+), plastic(69+) and carton(62+)
Number of each kind of items	Big Plastic Bottle	16
	Middle Plastic Bottle 1	12
	Middle Plastic Bottle 2	9

	Small Plastic Bottle	14
	Plastic Cup	18
	Milk box	16
	Big Carton	21
	Small Carton	20
	Recyclable Paper	5
	Metal Can	7
	Other Metal	10
Average Size of Files		54.2 kb
Average Length of Sounds		0.47 s

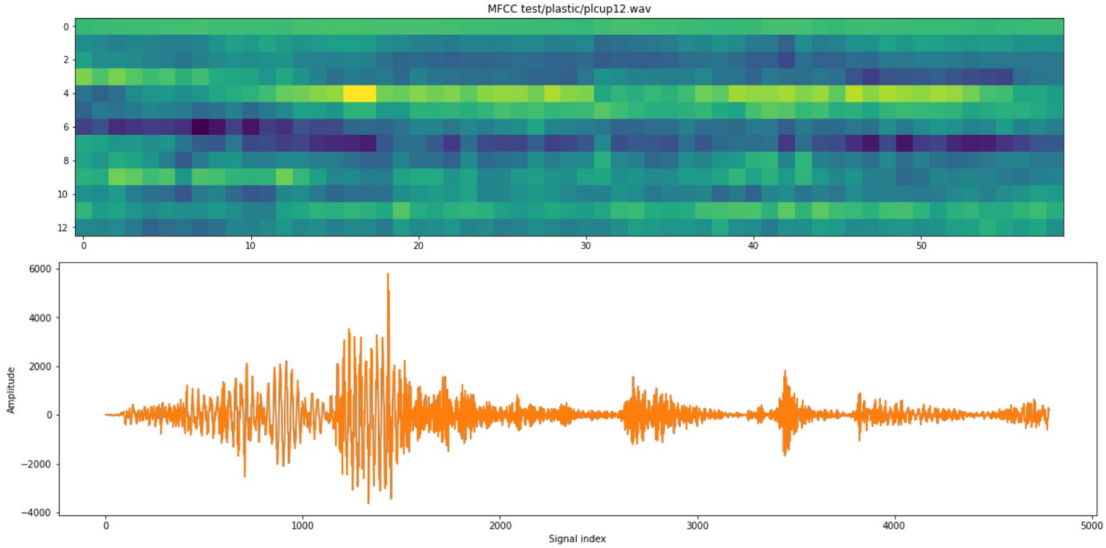


Fig 3.2 Sound from a Plastic Cup (MFCC features and voice wave)

3.1.2 Data Preprocessing

After obtaining the garbage falling sound in the real world, preprocessing is needed, which is both for the training data and input data. The preprocessing work is divided into four parts: **clipping, speed change, channel integration and noise reduction**.

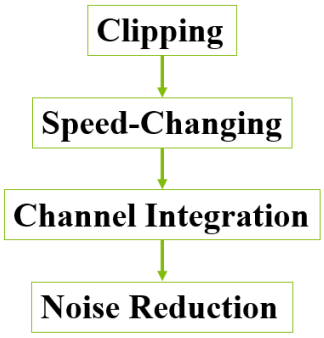


Fig 3.3 Data Preprocessing

(1) Cut out the part without sound information. Clipping aims to edit the raw 2-second audio into shorter segment with all valid information. After that, the sound length is about one second.

(2) Because FFT limits the signal length (512), all the edited audios need to have the processing of 1.5-times speed. After that, the sound length is about 0.7 second, which can meet the requirements of algorithm without losing the characteristics of sound signal.

(3) The sound data collected from sound sensors (like microphone) are generally consists of left and right channels. In order to reduce the pressure of data processing while maintaining the amount of information as much as possible. We reduce the dimension of the two-sound channel, average the parameters of the left and right channels.

(4) Because the sound contains some environmental noise, the recognition effect may be affected. Therefore, we will have a unified noise reduction on audios.

3.2 Acoustic Model

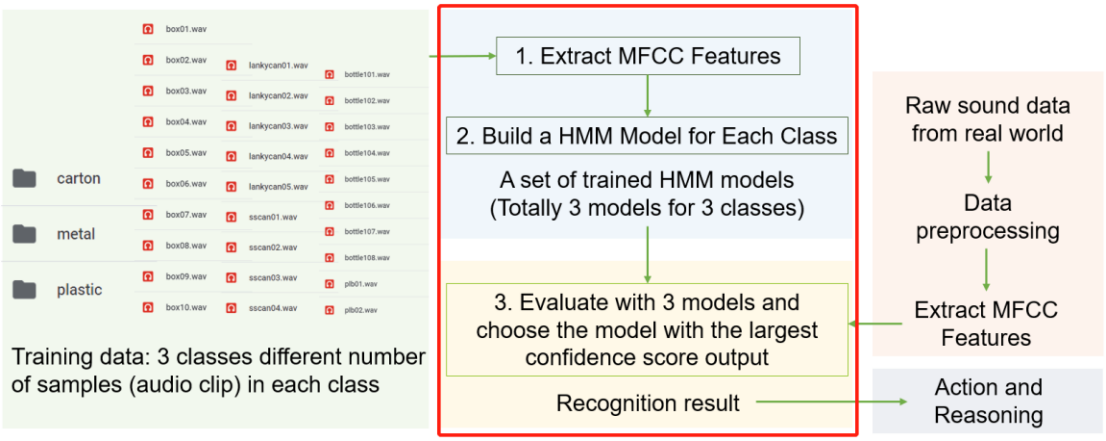


Fig 3.4 Acoustic Model (In red box)

The sound signal is generated by the garbage vibration caused by the collision between the garbage and the bottom of the garbage can. The preprocessed sound clip is inputs of our acoustic model, which outputs the garbage classification labels. The acoustic model building consists of three steps: feature extraction, HMM building and results

output.

3.2.1 Feature Extraction

Through data preprocessing, the voice knowledge was turned into audio sequence for processing.

The key to distinguish the sound of garbage made of different materials is to distinguish the different shapes of sound channels. If we can accurately know the shape of the sound channel, the representation of different phonemes can be gotten. The shape of the sound channel is embodied in the envelope of the *short-time power spectrum* of the sound signal, so feature extraction needs to accurately represent the envelope information. We need to extract their MFCC (Mel-scale Frequency Cepstral Coefficients) features.

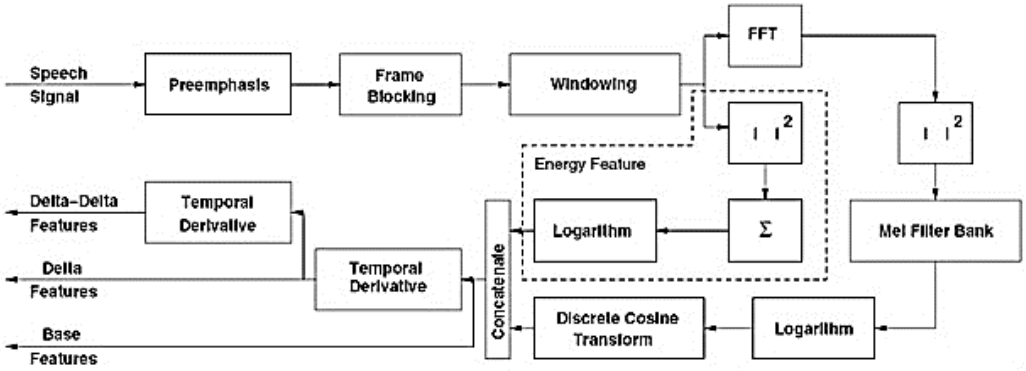
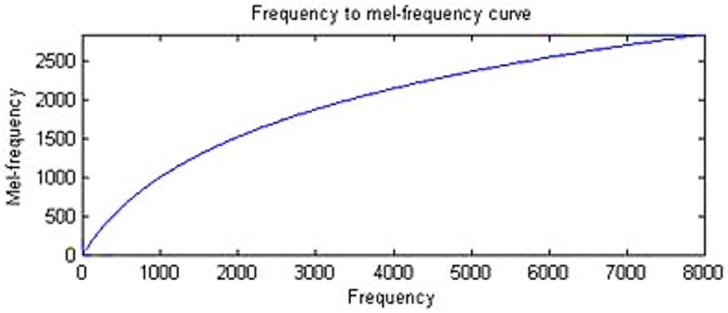


Fig 3.5 MFCC Features Extraction Basic Flow

First, the pending signal is pre-emphasized, divided into frames, windowed and FFT (Fast Fourier Transform). Then, in the calculation of power spectrum, the obtained power spectrum is passed through a triangular band-pass filter, and the filtered output results are converted into logarithmic form using the relationship between Mel-scale and linear frequency. Finally, the MFCC coefficients are obtained by DCT transformation.



$$f_{mel}(f) = 2595 \cdot \log\left(1 + \frac{f}{700Hz}\right)$$

Fig 3.6 Relation Between Mel-Frequency and Linear Frequency

Besides, the feature extraction of voice signal can also help the model to understand the sound content and discard other things (such as background noise and emotion, etc.).

In the project implementation, we use the existing python function *mfcc()* in the library *python_speech_features* to extract the MFCC features of sound signals.

```
def mfcc(signal,samplerate=16000,winlen=0.025,winstep=0.01,numcep=13,
         nfilt=26,nfft=None,lowfreq=0,highfreq=None,preemph=0.97,ceplifter=22,appendEnergy=True,
         winfunc=lambda x:numpy.ones((x,))):
    """Compute MFCC features from an audio signal.

    :param signal: the audio signal from which to compute features. Should be an N*1 array
    :param samplerate: the sample rate of the signal we are working with, in Hz.
    :param winlen: the length of the analysis window in seconds. Default is 0.025s (25 milliseconds)
    :param winstep: the step between successive windows in seconds. Default is 0.01s (10 milliseconds)
    :param numcep: the number of cepstrum to return, default 13
    :param nfilt: the number of filters in the filterbank, default 26.
    :param nfft: the FFT size. Default is None, which uses the calculate_nfft function to choose the smallest size that does not drop sample data.
    :param lowfreq: lowest band edge of mel filters. In Hz, default is 0.
    :param highfreq: highest band edge of mel filters. In Hz, default is samplerate/2
    :param preemph: apply preemphasis filter with preemph as coefficient. 0 is no filter. Default is 0.97.
    :param ceplifter: apply a lifter to final cepstral coefficients. 0 is no lifter. Default is 22.
    :param appendEnergy: if this is true, the zeroth cepstral coefficient is replaced with the log of the total frame energy.
    :param winfunc: the analysis window to apply to each frame. By default no window is applied. You can use numpy window functions here e.g. winfunc=numpy.hamming
    :returns: A numpy array of size (NUMFRAMES by numcep) containing features. Each row holds 1 feature vector.
    """
    nfft = nfft or calculate_nfft(samplerate, winlen)
    feat,energy = fbank(signal,samplerate,winlen,winstep,nfilt,nfft,lowfreq,highfreq,preemph,winfunc)
    feat = numpy.log(feat)
    feat = dct(feat, type=2, axis=1, norm='ortho')[:,:numcep]
    feat = lifter(feat,ceplifter)
    if appendEnergy: feat[:,0] = numpy.log(energy) # replace first cepstral coefficient with log of frame energy
    return feat
```

Fig 3.7 MFCC Function in the *python_speech_features* Library

```
#python_speech_features库提取每一帧的mfcc并计算13个特征的平均值
def mfcc_mean_features(filename):
    mfcc_mean_features = list()
    (rate,sig) = wav.read(filename)
    mfcc_feat = mfcc(sig,rate)
    #d_mfcc_feat = delta(mfcc_feat, 2)
    #logfbank生成26个特征
    #fbank_feat = logfbank(sig,rate)
    for el in mfcc_feat.transpose():
        mfcc_mean_features.append(np.mean(el))
    return mfcc_mean_features
```

Fig 3.8 Call the MFCC Function and Realize Feature Extraction

3.2.2 HMM Building

Hidden Markov model (HMM) is one in which you observe a sequence of emissions, but do not know the sequence of states the model went through to generate the emissions. Analyses of hidden Markov models seek to recover the sequence of states from the observed data.

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

Fig 3.9 The Components Specifying HMM

A *Hidden Markov Model* (HMM) allows us to talk about both observed events (like words that we see in the input) and hidden events (like part-of-speech tags) that we think of as causal factors in our probabilistic model.

HMM has been the mainstream method of actual sound recognition system since 1980s when is the first time to be used in speech recognition. Even today, End-to-End sound recognition systems continue to work. Many mainstream systems in the industry still are HMM-based methods. In many cases, deep neural networks are introduced to add into the traditional HMM/GMM model.

HMM has three key issues: (1) **Likelihood**: the probability of an HMM generating a series of observation sequences x---the Forward-Backward Algorithm is used. (2) **Decoding**: Given a string of observation sequences x, find the most likely subordinate HMM state sequence---the Viterbi Algorithm is used. (3) **Learning**: Given an observation sequence x, train HMM parameters λ ---the B-W Algorithm is used.

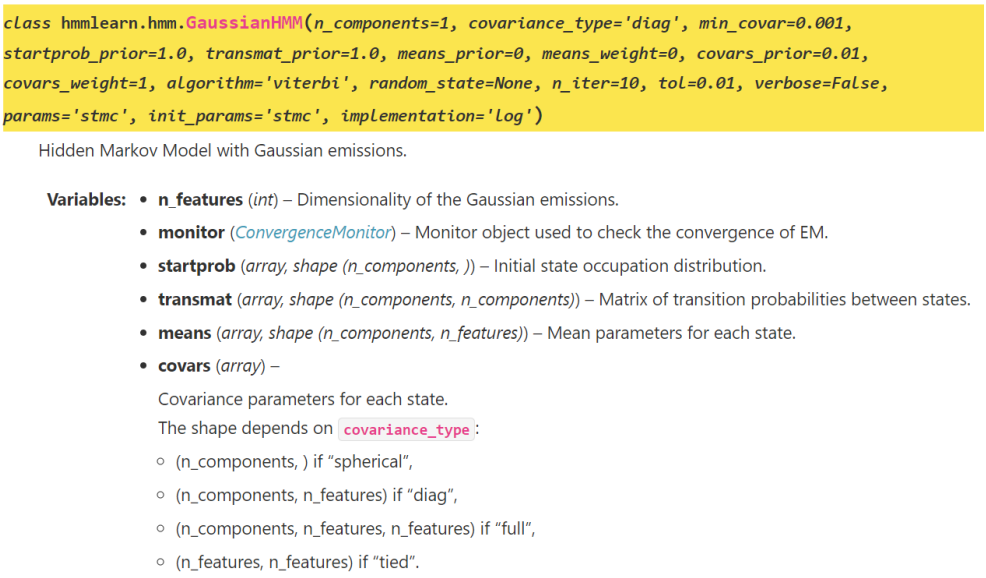
Key idea of HMM/GMM: “Optimal policy is composed of optimal sub-policies”.

1. **Initialization**: Calculate probability of the first day state based on first day observation and (assumed to be equal) prior probability starting from all possible states.
2. **Recursion**: For all following days, calculate probability of each state based on

current observation and the largest (previous state probability×transition probability) from the previous day. Record the 'best path' ending at current state from the previous day.

3. **Termination and back tracing**: For the last day, choose the state with the highest probability. Trace back according to the recorded most probable path.

In the acoustic model of this project, we have selected GaussianHMM as our basic, which can be understood as a Hidden Markov Model with Gaussian Distribution. The python library *hmmlearn* has provide us a quick way to build models. We use the function *GaussianHMM* to define the class which can handle all HMM related processing and create, train and save a HMM model.



```
class hmmlearn.hmm.GaussianHMM(n_components=1, covariance_type='diag', min_covar=0.001,
startprob_prior=1.0, transmat_prior=1.0, means_prior=0, means_weight=0, covars_prior=0.01,
covars_weight=1, algorithm='viterbi', random_state=None, n_iter=10, tol=0.01, verbose=False,
params='stmc', init_params='stmc', implementation='log')
```

Hidden Markov Model with Gaussian emissions.

Variables:
- **n_features** (*int*) – Dimensionality of the Gaussian emissions.
- **monitor** (*ConvergenceMonitor*) – Monitor object used to check the convergence of EM.
- **startprob** (*array, shape (n_components,)*) – Initial state occupation distribution.
- **transmat** (*array, shape (n_components, n_components)*) – Matrix of transition probabilities between states.
- **means** (*array, shape (n_components, n_features)*) – Mean parameters for each state.
- **covars** (*array*) –

  Covariance parameters for each state.

  The shape depends on `covariance_type`:
  - (n_components,) if "spherical",
  - (n_components, n_features) if "diag",
  - (n_components, n_features, n_features) if "full",
  - (n_features, n_features) if "tied".

Fig 3.10 GaussianHMM Function in Python Library

3.3.3 Results Output

Our acoustic model has learned the voice knowledge. According to the classification of the training data set, we get three HMMs (build an HMM for each class). The next step is to classify the garbage types according to the input sound data. Pre-processing and feature extraction are necessary for the raw sound from real world. Then compare it with the model. Iterate through all HMM models and pick the one with the highest score.

```
# Results Output
# 1: Input audio and preporcessing
test_file_name = ' '
raw_sampling_freq, raw_audio = wavfile.read(test_file_name)
sampling_freq, audio = preprocess(audio, sampling_freq)

# 2: Extract MFCC features
mfcc_features = mfcc(audio, sampling_freq)
max_score = None
output_label = None

# 3: Iterate through all HMM models and
#    pick the one with the highest score
for item in hmm_models:
        hmm_model, label = item
        score = hmm_model.get_score(mfcc_features)
        if max_score is None:
                max_score = score
                output_label = label
        if score > max_score:
                max_score = score
                output_label = label
print('File:%s,Pred:%s,Score:%.4f'%(test_file_name,output_label,max_score))
```

Fig 3.11 Results Output Part

The code above shows our idea. Evaluate with three models and choose the model with the largest confidence score output.

3.3 Function and Machine Reasoning

After the acoustic model identifies the current garbage classification, the garbage type label will be passed to the function implementation and machine reasoning parts through the API.

In this part, our system will realize the interaction with users and reasoning according to the type of garbage. These different contents will be vividly displayed to users, so that users can clearly know the contribution of their environment-friendly behaviors to the earth and sustainable development, thus encourage and promote the behaviors of garbage classification.

Label: Plastic
- Result:
  - You put in a piece of recyclable plastic
- Environmental Gain:
  - About 25g of plastic was recycled
  - About 125g of oil resources were saved
  - About 100g CO2 emissions are reduced

Label: Metal
- Result:
  - You put in a piece of recyclable metal
- Environmental Gain:
  - About 15g of plastic was recycled
  - About 300g of metal mineral were saved
  - About 80g CO2 emissions are reduced

Label: Paper
- Result:
  - You put in a piece of recyclable carton
- Environmental Gain:
  - About 50g of paper was recycled
  - About 400g forest resources were saved
  - About 300g CO2 emissions are reduced

Label: Glass
- Result:
  - You put in a piece of recyclable glass
- Environmental Gain:
  - About 300g of glass was recycled
  - About 220g quartz sand was saved
  - About 2kg CO2 emissions are reduced

Fig 3.12 Knowledge Base for Interaction and Encouragement

We have sorted out the knowledge base of recyclable waste environment gains according to the public information such as the network, literature, interviews and news reports. Each piece of knowledge includes three levels of content. Label is the principal linkage which link the acoustic model. Result represents the meanings of such a case in the real world. Environment gain shows three main environment gain indicators, such as the volume of recovery, CO2 emission reduction, energy saving and so on.

When the garbage is misplaced, such as a metal garbage is thrown into carton class or the non-recyclable garbage is thrown into the three types of recyclable garbage we set, we will show users the feedback of "*Your Classified Error! The correct category is XXX*" and remind users of the correct classification knowledge.

4. Solution Implementation

4.1 Overall Design

Figure 4.1 shows an overview of our system architecture design.

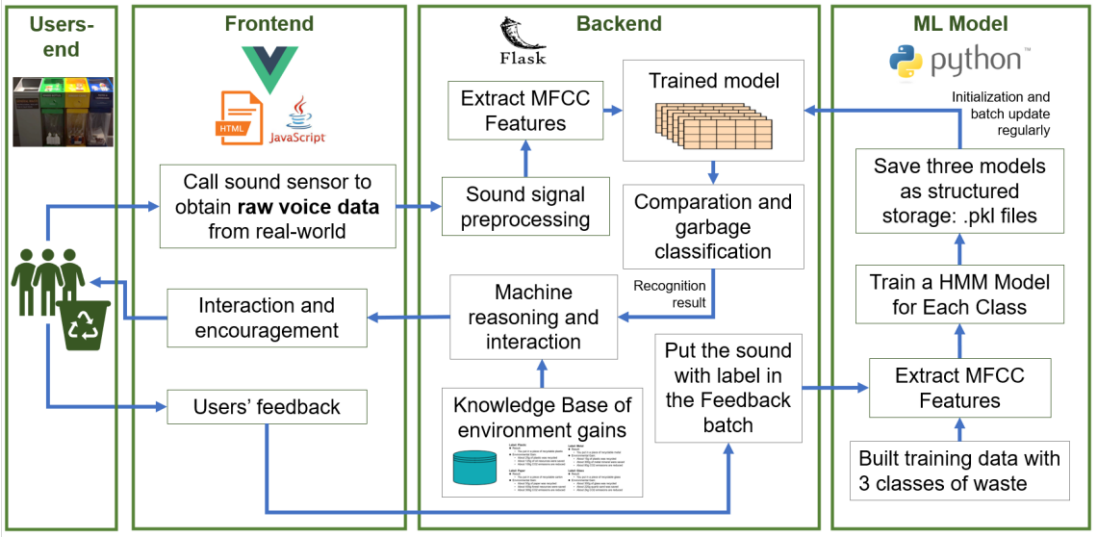


Fig 4.1 Overview of System Architecture and Process Flow

The Users-end in Fig 4.1 doesn't exist in MVP. We simulate the actual functions through the frontend of VUE. When users throw garbage into the trash can, mechanical sensor embedded in frontend will call sound sensor to obtain raw voice data from real-world. The frontend programs will save this voice as .wav file and send it to backend. After the processing of backend programs and models, the classification and reasoning results are transmitted back to the frontend. The frontend program will display the information.

The frontend is integrated with the backend request management module which is built based on the python Flask application. Backend program is the core of the entire executable system. We finish sound signal preprocessing, MFCC features extraction, model comparation and garbage classification and machine reasoning here. Our machine learning model is trained offline. We save the model with all parameters in three .pkl files (each file for each model). Backend read these structured storages as knowledge learning so that predict the classifications.

In addition, the system architecture is designed with online hosted web application in mind. When users use the device, they will receive a feedback pop-up interface. The system will ask users whether this classification is correct. System will record this feedback and store the feedback results, labels and recordings locally. After accumulating to a certain batch or a certain period (such as a week), the system will send it to the cloud backend over the Internet. Engineers from ***EZSORT*** will clean the data and re-train the model. The new model will be sent back to the terminal device to strengthen the function and update the parameters.

4.2 System Frontend

As an MVP, the frontend user interface design ought to be simple but practical. The framework for the system frontend is a local-web application that is rendered based on HTML and JavaScripts. The user interface is rendered in web browser for interaction. In addition, some of the on-page interactive functions was built using JavaScripts ajax.

Specifically, we use VUE as our frontend framework that eliminates DOM operations in native JavaScript and simplifies writing. VUE is a progressive framework for building user interfaces. VUE implements data bidirectional binding based on MVVM (Model View_ViewModel) idea, realizes data bidirectional binding. It is designed from the ground up to be incrementally adoptable, and can easily scale between a library and a framework depending on different use cases. It consists of an approachable core library that focuses on the view layer only, and an ecosystem of supporting libraries that helps you tackle complexity in large Single-Page Applications.

Below shows the UI of our web with a button for user to record a piece of voice. The system calls sound sensor and sends the voice to backend.

Fig 4.2 Sound Collection Interface

After recording and uploading, our backend program will process the audio and give the prediction and reasoning. Then the result will be output to users in the frontend.

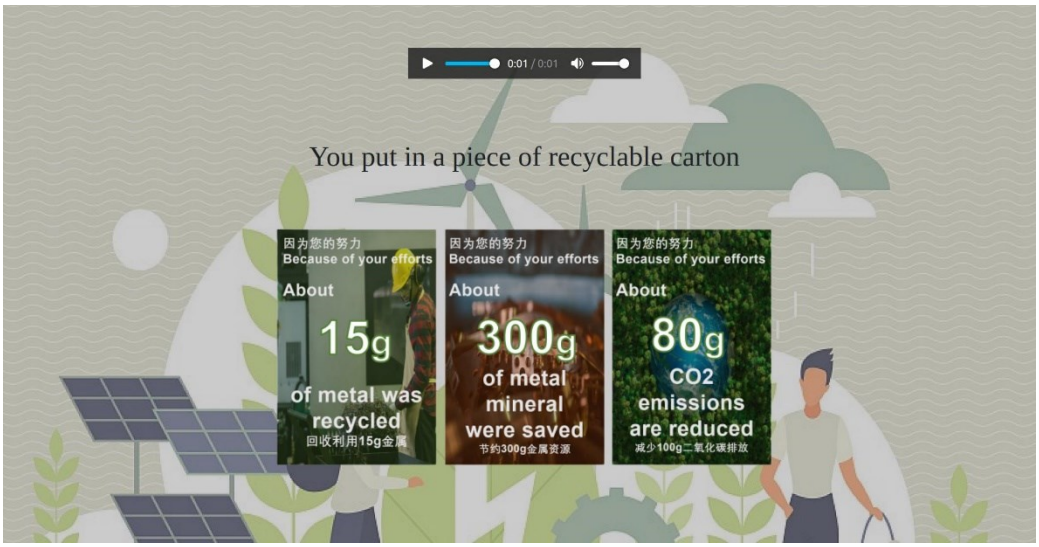


Fig 4.3 Prediction and Encouragement Interface

4.3 System Backend and Model

We have introduced the details of backend programs and model building in Chapter III.

The frontend is integrated with the backend request management module which is built based on the python Flask application. Backend program is the core of the entire executable system. We finish sound signal preprocessing, MFCC features extraction, model comparation and garbage classification and machine reasoning here. Our machine learning model is trained offline. We save the model with all parameters in three .pkl files (each file for each model). Backend read these structured storages as knowledge learning so that predict the classifications.

4.4 System Evaluation

4.4.1 Developer Testing

The develop team of ***EZSORT*** simulated and tested various situations that may occur in actual using, which including intentional or unintentional misplacement, putting multiple pieces of garbage together, different falling ways, etc. The test proves the robustness of our system. Comprehensive accuracy is above 95%.

(1) **Misplacement**. When the garbage is misplaced, such as a metal garbage is thrown into carton class or the recyclable garbage is thrown into the three types of non-recyclable garbage we set, we will show users the feedback of "*Your Classified Error! The correct category is XXX*" and remind users of the correct classification knowledge. We have tested about 50 pieces of voice for misplacement, the accuracy is about 90%. Almost all the errors occur in the case that general garbage is thrown into the three types of recyclable garbage we set. This part is a blind spot of this system. While the whole accuracy is satisfied. We consider the test is passed.

Table 4.1 Some Testing Sample for Misplacement

Item	Classification	Thrown into	Prediction	Evaluation
Used Tissue	General	Carton	Metal, MISP	Na PASS
Used Plastic Cup	General	Plastic	Plastic, CORP	Na PASS
300ml Pop Can	Metal	Plastic	Metal, MISP	PASS
500ml Plastic Bottle	Plastic	Carton	Plastic, MISP	PASS
1L Milk Box	Carton	General	Carton, MISP	PASS
150g Drinking Box	Carton	Carton	Carton, CORP	PASS

(2) **Multiple pieces of garbage**. We have tested about 20 pieces of voice for multiple pieces of garbage, the accuracy is about 70%. This part is a blind spot of this system. While the whole accuracy is satisfied. We consider the test is passed. In the actual using case, we need to educate users to put only a single piece of garbage at a time.

Table 4.2 Some Testing Sample for Multiple Putting

Item	Classification	Thrown into	Prediction	Evaluation
300ml Pop Can+500ml Plastic Bottle	Metal + Plastic	Plastic	Metal, MISP	PASS
300ml Pop Can+500ml Plastic Bottle	Metal + Plastic	Metal	Metal, CORP	Na PASS
300ml Pop Can+1L Milk Box	Metal + Carton	Metal	Carton, MISP	PASS
500ml Plastic Bottle+ 1L Milk Box	Plastic + Carton	Carton	Plastic, MISP	PASS

(3) **Different falling**. We have tested about 30 pieces of voice for different falling ways, the accuracy is above 97%. The fall methods, directions, impact materials, height and speed have no significant effect on the prediction results.

Table 4.3 Some Testing Sample for Different Falling				
Item	**Classification**	**Falling**	**Prediction**	**Evaluation**
300ml Pop Can	Metal	0.5m height horizontal metal plate	Metal, CORP	PASS
500ml Plastic Bottle	Plastic	0.3m height vertical wooden board	Plastic, CORP	PASS
1L Milk Box	Carton	0.25m height vertical plastic board	Carton, CORP	PASS
150g Drinking Box	Carton	0.4m height horizontal carton garbage	Carton, CORP	PASS

4.4.2 Users' Feedback

EZSORT invited about 10 friends to participate in the use as internal test users. **89%** of them think that the intelligent garbage classification motivator is useful with great social, economic and environmental value. **56%** of them think there is some room for improvement in our system. **All of them are willing to accept and use the new system** and think that this motivator will encourage people to sort their waste correctly successfully.

Fig 4.4 Users' Feedback

5. Conclusion

5.1 Summary

We have successfully built a functioning MVP of intelligent garbage classification motivator, which could automatically identify the type of garbage and make reasoning based on the category label, encouraging people to classify garbage according to the correct requirements. The system produces satisfactory results, and the user feedbacks are good as well. We have done a good practice of project management too, keeping progressing throughout the project. Most importantly, the system we have developed has the potential to play a role within the waste collection system by providing chances to improve the recovery rate

Throughout the whole project period, we have a better understanding of the course content and utilize them in our projects. In this project, we applied the knowledge-based reasoning systems we have learned from machine reasoning system courses, as well as HMM and speech recognition from cognitive system courses. These practices enhanced our understanding of the classroom material. Our team also had a wonderful time working on this project, and along the way we have gained valuable knowledge and practical skills while achieving the prescribed objectives of this project.

5.2 Limitation

1. Due to limited time, we only simulated the function of the system, and did not design the hardware system and structure. The simulation results tell us that the system is feasible.

2. Our system is currently very good at identifying recyclable waste. While the ability of classifying non-recyclable waste is still need to be improved. And in the recyclable garbage, there are only three categories that appear most frequently in daily life.

3. The recognition results rely on high-quality sound knowledge as the training

dataset for the model. The datasets in this project were collected and established manually by project members, which took a lot of time and manpower.

5.3 Future Plan

1. From an algorithm perspective, we will likely look for more potential models to achieve sound recognition, including deep neural networks, etc.
2. We may not be limited to sound recognition. The introduction of sensors such as vision or infrared is a potential solution that will help improve accuracy and expand the range of recognition.
3. If time and conditions allow, it would be ideal to design the hardware structure and implement the complete project content on the hardware system.
4. Our team members are communicating with some environmental protection companies from China (like Qingyujike), hoping to translate the research results into practice

References

[1] 2014 UN-Habitat, “WORLD CITIES REPORT 2022: Envisaging the Future of Cities

[2] National Environment Agency official website, https://www.nea.gov.sg/

[3] Song Lijie, Chen Shanping, Zhao Youcai. Sustainable domestic waste treatment and recycling technology [M]. Beijing: Chemical Industry Press,

[4] Li Bing, Zhao Youcai, Shi Qingyan, et al. Research on the sorting mode of domestic waste in Shanghai [J]. Journal of Tongji University (Natural Science Edition), 2007, 35(4): 507-510

[5] Xidian University. A horizontal garbage bag breaking machine: CN104960719A[P] . 2015-10-07

[6] Yizhong Group Dalian Design and Research Institute Co., Ltd. A domestic garbage bag breaking machine: CN102513195A[P] . 2012-06-27

[7] An isolated word recognition. Prateek Joshi, Python Machine Learning Cookbook, Packt Publishing, 2016, Code available at https://github.com/PacktPublishing/Python-Machine-Learning-Cookbook

[8] Lu, G., Wang, Y. and Yang, H. (2020) One-Dimensional Convolutional Neural Networks for Acoustic Waste Sorting. Journal of Cleaner Production, 271, Article ID: 122393. https://doi.org/10.1016/j.jclepro.2020.122393

Appendix A: Project Proposal

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)
PRACTICE MODULE: Project Proposal

Date of proposal: 18 September 2022
Project Title: ISS Project – EZSORT, An Intelligent Garbage Classification Motivator
Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)* Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore NATIONAL UNIVERSITY OF SINGAPORE (NUS) Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg
Background/Aims/Objectives: The sound and reasonable waste management carrys great significance of for low-carbon and green future. We propose an intelligent garbage classification motivator, in order to encourage citizens to throw-in the garbage correctly as its' classifications. EZSORT can be the platform to rise the people awareness and provide a window to allow them to step out. We believe this small action can be a big step tomorrow.
Requirements Overview: • Research ability • Programming ability • System integration ability
Resource Requirements (please list Hardware, Software and any other resources) Hardware proposed for consideration: • Local and cloud server (CPU), sound sensor, etc. Software proposed for consideration: • Software development framework, e.g., VUE, Python Flask, etc. • Deep learning tools, e.g., Python Speech Recognition, Python HMM, etc. • Cloud computing/server, e.g., Amazon, Google, IBM, Azure, etc.

Number of Learner Interns required: (Please specify their tasks if possible)

A group of four to 4 project members (or individual work upon lecturer approval)

Methods and Standards:

Procedures	Objective	Key Activities
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of project and ensure the achievement of business objectives.	1. Gather & Analyze Requirements 2. Define internal and External Design 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	· To develop the source code in accordance to the design. · To perform unit testing to ensure the quality before the components are integrated as a whole project	1. Setup Development Environment 2. Understand the System Context, Design 3. Perform Coding 4. Conduct Unit Testing
Integration Testing and acceptance testing	To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing.	1. Prepare System Test Specifications 2. Prepare for Test Execution 3. Conduct System Integration Testing 4. Evaluate Testing 5. Establish Product Baseline
Acceptance Testing	To obtain ISS user acceptance that the system meets the requirements.	1. Plan for Acceptance Testing 2. Conduct Training for Acceptance Testing 3. Prepare for Acceptance Test Execution 4. ISS Evaluate Testing 5. Obtain Customer Acceptance Sign-off
Delivery	To deploy the system into production (ISS standalone server) environment.	1. Software must be packed by following ISS's standard 2. Deployment guideline must be provided in ISS production (ISS standalone server) format 3. Production (ISS standalone server) support and troubleshooting process must be defined.

Team Formation & Registration

Team Name: ***EZSORT***
Project Title (repeated): EZSORT---An Intelligent Garbage Classification Motivator
System Name (if decided): EZSORT
Team Member 1 Name: YUE Yiran
Team Member 1 Matriculation Number: A0261678R
Team Member 1 Contact (Mobile/Email): 87891108/e0983072@u.nus.edu
Team Member 2 Name: Lin Zhengxi
Team Member 2 Matriculation Number: A0261674Y
Team Member 2 Contact (Mobile/Email): e0983068@u.nus.edu/87908202
Team Member 3 Name: GUO Rui
Team Member 3 Matriculation Number: A0176188N

Team Member 3 Contact (Mobile/Email): e0224928@u.nus.edu/83353951
Team Member 4 Name: ZOU Yufan
Team Member 4 Matriculation Number: A0261786R
Team Member 4 Contact (Mobile/Email): zou.yufan@u.nus.edu/98602313

Appendix B: Techniques and Skills

In this project, we applied the knowledge-based reasoning systems we have learned from machine reasoning system courses, as well as HMM and speech recognition from cognitive system courses. Here is the mapping of techniques and skills in this project.

Module Coerces	Knowledge and Skills
Machine Reasoning	Decision Automation Knowledge Modelling and Representation Rule Based System
Reasoning Systems	Retrieval Reasoning Systems (Analytic Task)
Cognitive Systems	Cognitive Knowledge Representation and Reasoning Speech Cognitive Systems Hidden Markov Model (HMM)

Appendix C: Personal Report

D.1 YUE Yiran (A0261678R)

Personal Contribution	➢ Control Progress and Overall Planning ➢ Data Preparation and Acoustic Model ➢ Market and Business Analysis
Learned	1) In the project implementation, I was responsible for the establishment of the voice knowledge base and the acoustic model, which are both had learned in courses. Happily, I have successfully applied what I have learned to the practical project. I took a deep dive into knowledge-based reasoning and machine learning-based retrieval reasoning systems. I experience the process such as gathering data, data clean up, model building, training and testing the model. Also, I have improved a lot in coding skills. 2) At the same time, I played the role as a PM, determined the topic and planed the progress with team members, which gave me the opportunity to look at the problem from the perspective of systems science once again. We tried to identify the problem, break it down, propose potential solutions, demonstrate feasibility, implement the project and submit an MVP. I think it's more important to have a solution to a problem than to have any one specific skill or knowledge. In addition, we'll always remember the sentence, "Teamwork is Victory", we have learnt in NUS-ISS Orientation Day.
Future	I learned a lot of modeling, knowledge representation and reasoning skills. The knowledge of proper knowledge modelling and

	representation will help to guide me in the process of designing an AI solution, especially when I am planning for an interview with an expert to elicit knowledge. Such model can also help to communicate the solution design and testing assumption, making sure that the solution meets the needs of the business.

D.2 LIN Zhengxi (A0261674Y)

Personal Contribution	➢ Market and Business Search ➢ Frontend and Backend Development ➢ UI Interface Design and Function Implementation
Learned	How to develop a front end and back end in Linux. You need to try a lot of new staff such as virtualbox, command and software. For front end design, the basic knowledge about html, CSS, JavaScript. Since we figure out what the system for and how it works, I try some framework to realize the system. First I learn a little about streamlit but I abandon the plan. Then I only want to try flask and the html for front end. Then, to design the system we need to finish the core code for function, so I help to improve it, try something about how to transform the data to npz, decide the algorithm weather should be supervised learning or unsupervised learning. Also add some demand to the task for teammates to make things done. Bond the front end and back end is also important and hard, I need to debug, change plans to make it work. Finally, we find data processing seems can work easily using Vue, so the frontend part update, some new knowledge need to learn, like axios. In conclude, I learn a lot from the project, turn the knowledge into the exercise and finish successfully. Also, I learn a lot about data processing, which I never use before.
Future	Frontend and backend can be used in any project, so the successful experience can help me to familiar with the structure. Then when I

	face the difficulties in future, I can use it to overcome. Next time I can try different programming language to develop project.

D.3 GUO Rui (A0176188N)

Personal Contribution	➢ Model Building and Core Algorithm ➢ Model Testing and Evaluation
Learned	1) After discuss with the group to basically establish the project theme, I learned a lot of important parameters and knowledge about audio. Checked out numerous papers and projects on voice recognition. All these works have enabled us to fully demonstrate the feasibility and effectiveness from a technical point of view. It is also the basis of my later work. 2) In the project, I use Hidden Markov Model as the base model to build machine learning algorithms. In this process, I have a deeper understanding of the principle, implementation, training and application of Hidden Markov Models. And finally realized a machine learning algorithm with high accuracy 3) In model testing and evaluation, I found that the length of the audio, the number of channels, noise, etc. will have a greater impact on the use and accuracy of the model. So I learned about noise reduction, audio compression and other methods and found suitable parameters so that the algorithm can have better accuracy and can meet the business requirements of the project 4) This is my first time developing a complete project using python. This project has improved my python programming ability, gained a lot of knowledge and experience in python development, and gave me a better understanding and experience of some important libraries in python such as numpy, scipy, glob, hmmlearn, etc. Communication and collaboration with other teammates is

	essential during project development. This project gave me more teamwork experience to communicate and collaborate efficiently with my teammates. During the course of the project, I learned a lot of knowledge and skills from my teammates. The project also gave me and my teammates a better understanding and made our team more united
Future	In this project, my teammates and I fully participated in the whole process from a real world problem to completing an intelligent system to solve the problem. We have successively completed the process of problem discovery and proposal, research, business requirements, business scenario analysis, knowledge base construction, logical disassembly, technical route selection, and technical implementation. This project give me a deeper understanding and opportunity to practice what I learned in the course, and it also let me to combine knowledge with real-world problems and gain valuable practical experience. I believe this project will be a great help to my future work in AI and algorithm development

D.4 ZOU Yufan (A0261786R)

Personal Contribution	➢ Data Preprocessing ➢ Backend Development ➢ Software Testing and Evaluation
Learned	1) Choose appropriate data preprocessing methods. When collecting recordings using my smartphone for model training, I realized that these sounds have a background noise, which cannot be ignored. Therefore, I found a Python library *noisereduce* to solve this issue and it worked well. I also tried to use Peak Locating method to locate the fragment in the recording where an object actually falls down, but the performance was less satisfying since we are handling several kinds of objects, and patterns of their falling sounds differ, causing it hard to find a general set of parameters to extract such fragments. Finally, simply cutting recording to a fixed length works. 2) Meanwhile, I built a backend as a bridge between users' input(recordings) and predict outcomes given by models. I learned to use Flask to accomplish this goal, including how to design interfaces for backend-frontend interaction, as well as page rendering for displaying the detection result and corresponding environmental gain.
Future	At present, my work of software testing still highly relies on the manual operation, while there are many automatic test tools available. In the future, automatic testing would be adopted to improve test case coverage and decrease test time.