T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

MMWAVE RADAR BASED TONGUE POSITION ESTIMATION FOR SPEECH THERAPY

ABDULLAH MUHAMMET YIĞIT

SUPERVISOR
PROF. DR. YUSUF SINAN AKGÜL

GEBZE
2024

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

MMWAVE RADAR BASED TONGUE POSITION ESTIMATION FOR SPEECH THERAPY

ABDULLAH MUHAMMET YIĞIT

SUPERVISOR
PROF. DR. YUSUF SINAN AKGÜL

2024
GEBZE

GEBZE TECHNICAL UNIVERSITY	GRADUATION PROJECT JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on by the following jury.

JURY

Member
(Supervisor) : Prof. Dr. Yusuf Sinan Akgül

Member
(Supervisor) : Dr. Yakup Genç

ABSTRACT

This project introduces an innovative approach to diagnose and treat language and speech disorders. It harnesses the advancements in technology, specifically mmWave radars, to automatically derive mappings of the International Phonetic Alphabet (IPA) chart for vowel phonetics in Turkish. The method is tailored for integration into speech therapy rehabilitation. The project represents a significant research opportunity in the field of speech therapy, with the overarching goal of enhancing accessibility and providing appropriate treatment options for a wider audience.

Notably, the use of radar ensures contactless data flow and compliance with data security standards, enabling real-time monitoring of facial and lip movement information in individuals. This contributes to the effectiveness and personalization of silent speech interfaces. Furthermore, the presented project is developing a system that considers both the spatial and temporal positions of the face and lips, resulting in more consistent and phonetically accurate detection over time. The technology's suitability is also being demonstrated.

ÖZET

Bu proje, dil ve konuşma bozukluklarını teşhis etmek ve tedavi etmek için yenilikçi bir yaklaşım sunmaktadır. Özellikle mmDalga radarları gibi teknolojinin ilerlemelerinden faydalanarak, Türkçe'deki ünlü fonetikler için Otomatik Uluslararası Fonetik Alfabesi (IPA) harita eşlemelerini çıkarmaktadır. Bu yöntem, konuşma terapisi rehabilitasyonunda kullanılmak üzere tasarlanmıştır. Proje, konuşma terapisi alanında önemli bir araştırma fırsatı sunmakta olup, daha geniş bir kitle için daha fazla erişim ve uygun tedavi seçenekleri sağlamayı amaçlamaktadır.

Özellikle, radarın kullanımı, veri akışının temas olmaksızın ve veri güvenliği standartlarına uygun bir şekilde sağlanmasına olanak tanır, bu da bireylerin yüz ve dudak hareket bilgilerinin gerçek zamanlı olarak izlenmesine olanak tanır. Bu, sessiz konuşma ara yüzlerini daha etkili ve kişiselleştirilmiş hale getirmeye katkı sağlar. Ayrıca, sunulan proje, yüz ve dudakların hem uzamsal hem de zamansal konumlarını dikkate alan bir sistem geliştirmekte olup, zaman içinde daha tutarlı ve fonetik olarak doğru tespit sağlamaktadır. Teknolojinin uygunluğu da gösterilmektedir.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION TO PROJECT

This study proposes a novel approach that utilizes acoustic information and mmWave radar signals to automatically derive mappings of vowel phonetics on the IPA chart. Subsequently, this process can be employed in a speech therapy rehabilitation designed for individuals with speech disorders who speak Turkish.

1.1. mmWave Radar

[1]To overcome the limitations of traditional ultrasound tongue imaging (UTI) techniques, radar technology will be employed to make it usable in therapy processes, thanks to its ability to operate independently of mobile and environmental conditions. In this process, frequency-modulated continuous wave (FMCW) radar technology will be utilized to determine the tongue positions of specific vowel sounds in Turkish, especially /i/ and /ü/.

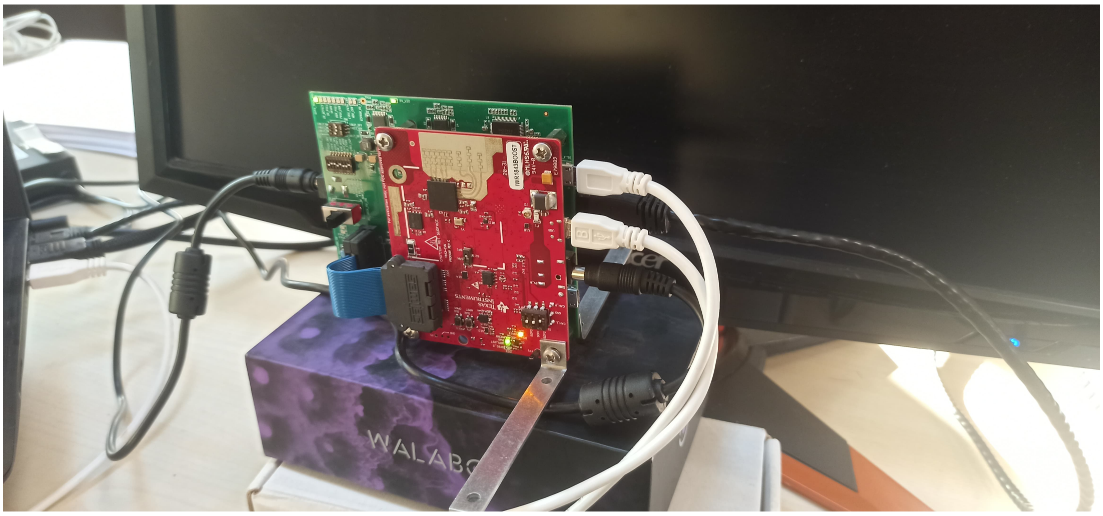


Figure 1.1: mmWave Radar.

1.2. IPA (International Phonetic Alphabet)

[2][3]IPA (International Phonetic Alphabet) is a universally recognized standard for representing sounds used in many languages worldwide. It is crucial for the techniques and data employed in speech therapy to have a broad language independence.

IPA provides a standardized system for representing these universal language sounds and articulatory positions. Understanding which sounds correspond to which articulatory positions is vital in the therapy process. IPA will assist our project by specifying this information.You can see the IPA chart of vowels in Figure 1.2.

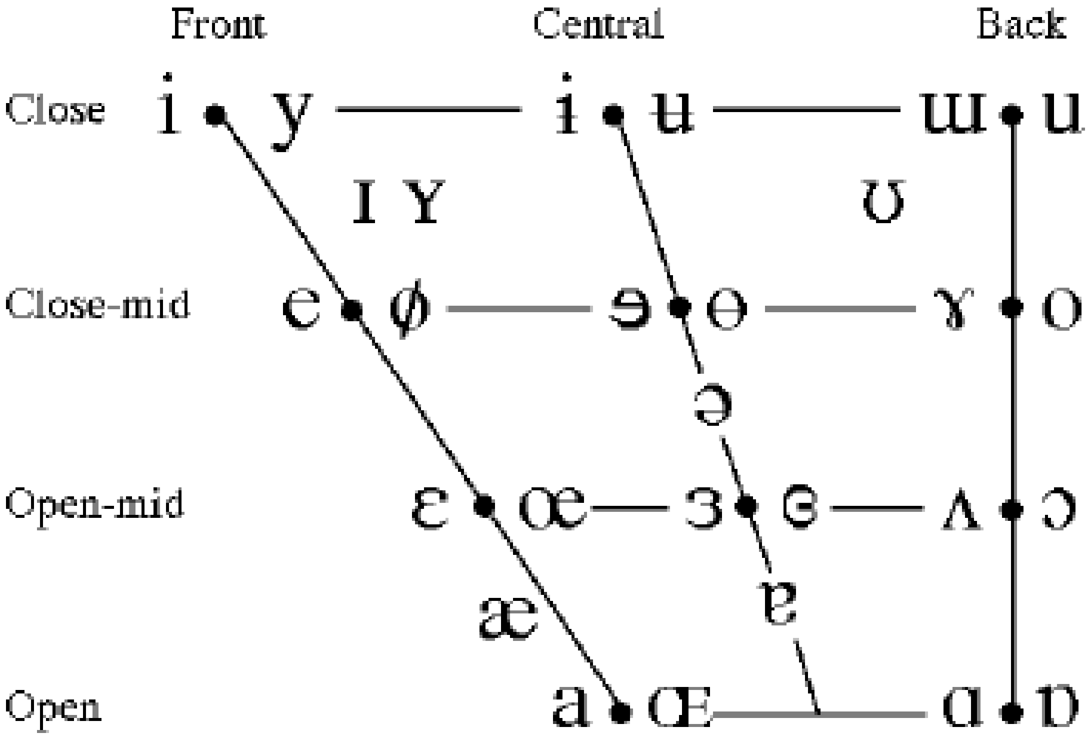


Figure 1.2: International Phonetic Alphabet Chart.[4]

2. DATASET

The dataset for predicting language positions was derived from the collection of radar and sound data obtained from subjects. To facilitate language position prediction, these data underwent processing, and a model was constructed using a suitable machine learning algorithm.

2.1. Data Collection Process

The data collection process aims to assess the effectiveness of the technology and methods employed to determine the positions of specific vowel sounds within the tongue. These data are subsequently utilized for the training and validation of the algorithm to be developed. Thus, the objective is to associate articulation within the tongue with radar data and make it usable in therapy processes. The data collection phase focuses on providing fundamental data for the therapy process. In order to estimate tongue positions, individuals are required to articulate the vowel letters in the Turkish alphabet. Participants sequentially articulate these vowel letters, and radar information is recorded in the dataset each time a letter is pronounced.

2.1.1. Objective

Each participant was instructed to vocalize a vowel sound within a 2-second timeframe, and this process was repeated consecutively 20 times for each vowel letter. 5 vowel letter are collected which are /a/, /e/, /i/, /o/ and /y/. With this way, a total of 3400 data belonging to 5 classes were collected from 35 participant.

2.2. Processing Dataset

2.2.1. Libraries and Modules Used

The process uses Python libraries and modules including pandas, numpy, os, mmwave, openradar, tensorflow, librosa and some scripts. These libraries provide necessary functions for data processing.

2.2.2. Processing Radar Data

[5]The transformation of radar data into spectrograms was accomplished utilizing the mmwave and openradar modules in Python. Two types of spectrograms were generated in this project. The first one is a Doppler-range spectrogram that visually represents the range and Doppler information of the radar signal, as observed in Figure 2.1. In this representation, the x-coordinate signifies the range, while the y-coordinate signifies the Doppler information. The second type is a velocity-time spectrogram, depicting the velocity and time information of the radar signal, as illustrated in Figure 2.2. In this case, the x-coordinate represents time, while the y-coordinate represents velocity information.

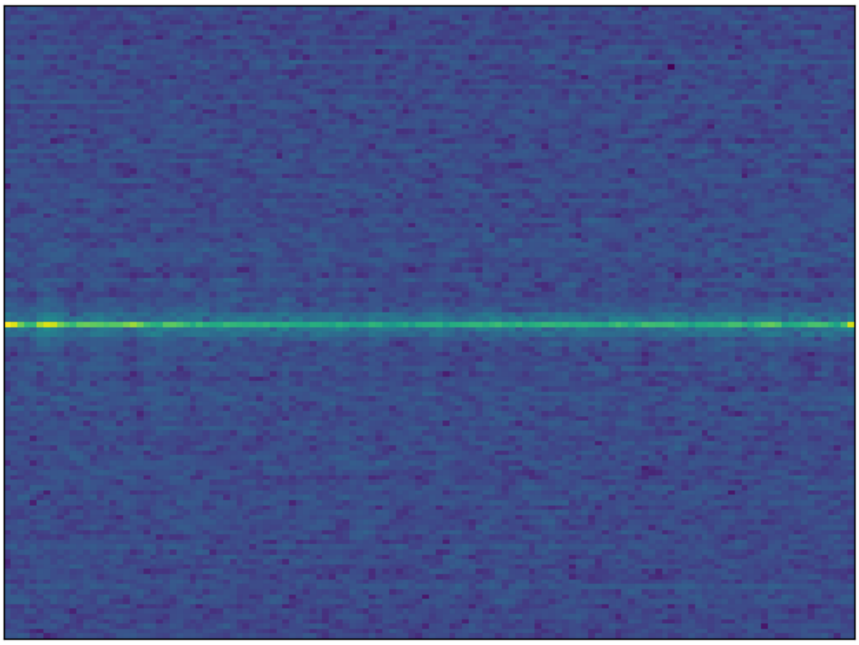


Figure 2.1: Doppler-Range Spectrogram.

2.2.3. Processing Audio Data

The audio data underwent transformation into spectrograms using the librosa module in Python, as depicted in Figure 2.3.

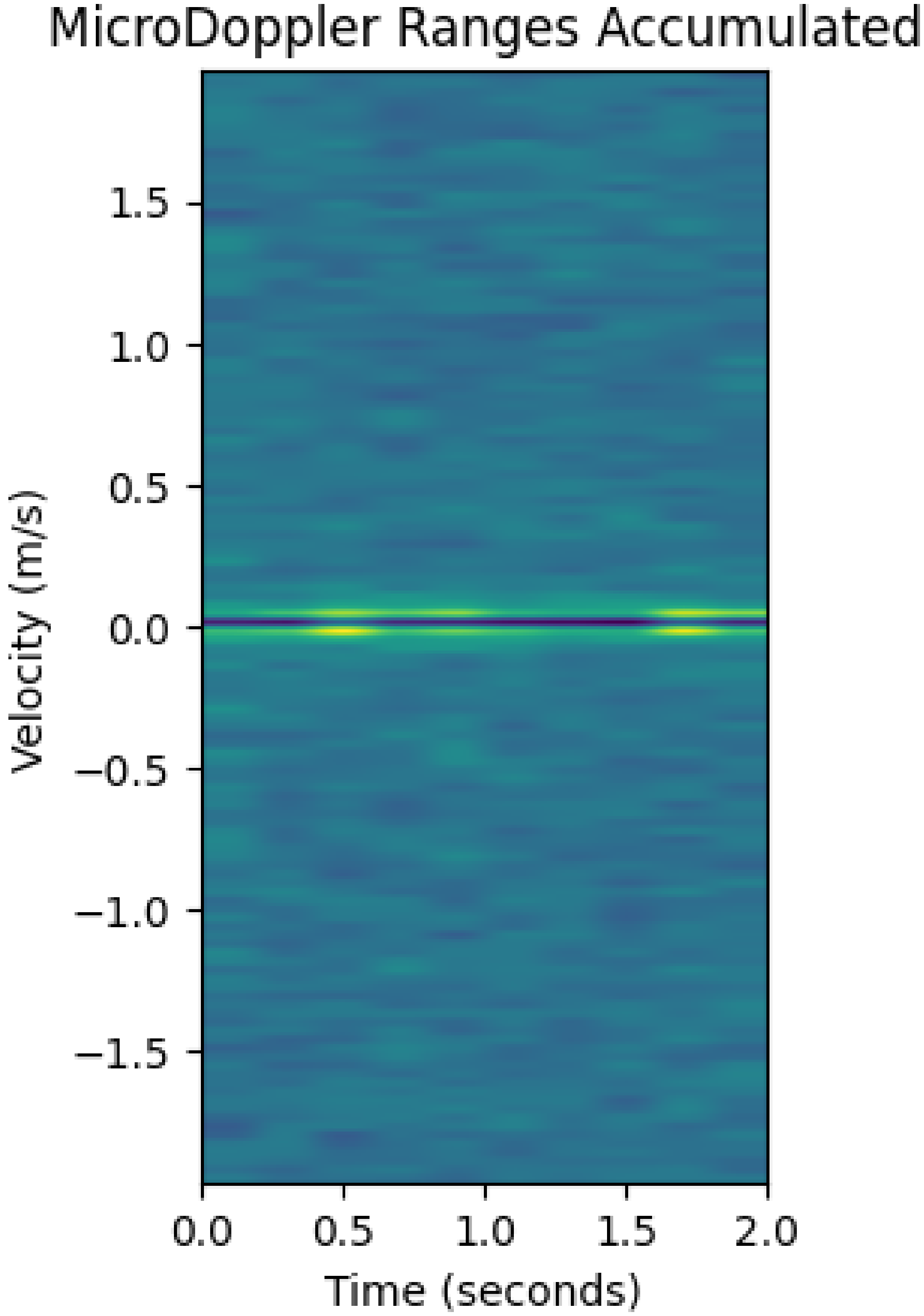


Figure 2.2: Velocity-Time Spectrogram.

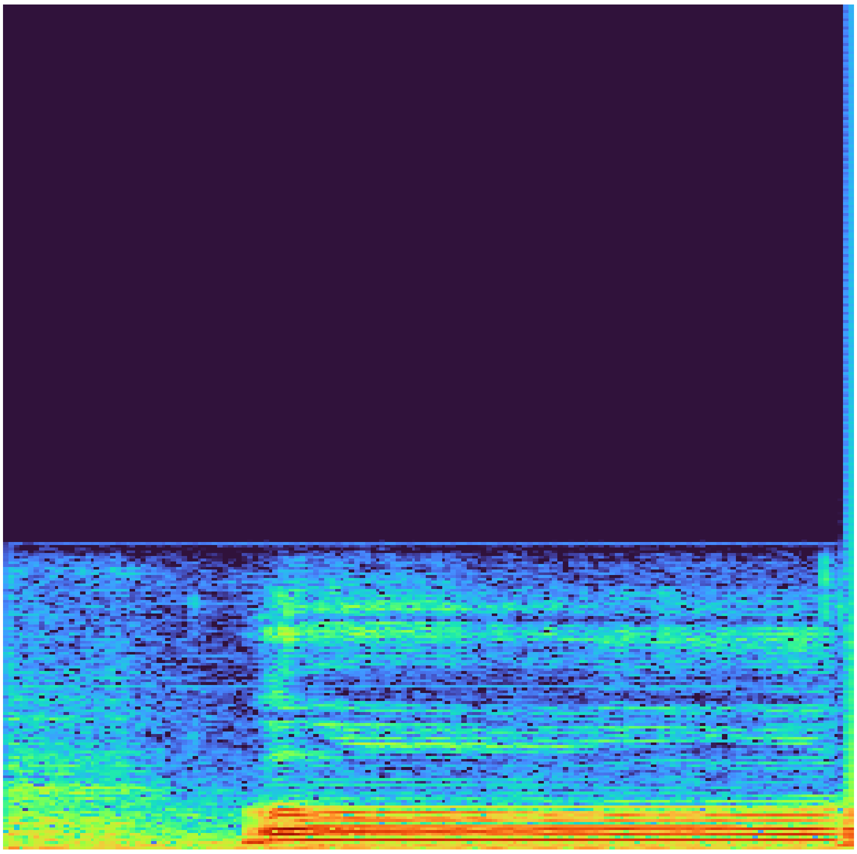


Figure 2.3: Audio Spectrogram.

3. MODELING PHASE

The modeling phase applies to audio spectrograms. The Conv2D function from the TensorFlow library was employed to build a model using the spectrograms generated during the model training phase.

3.1. Learning rate and loss of model

Here is the rate of model learning if Figure 3.2. We can see that learning rate schedules in deep learning models often involve an initial phase where the learning rate is relatively high, followed by a gradual decrease over time. This pattern of a learning rate that ”peaks” before decreasing serves several purposes.

Loss is s a measure of how well a model’s predictions align with the true values or labels of the training data. The loss quantifies the error or mismatch between the predicted output of the model and the desired output.

During the training process, the model iteratively adjusts its parameters or weights to minimize the loss. By minimizing the loss, the model aims to improve its ability to make accurate predictions on unseen data.

3.2. Confusion Matrix

Confusion matrix is a table used in the field of machine learning and statistics to evaluate the performance of a classification algorithm. It provides a detailed and comprehensive overview of the model’s predictive accuracy by comparing predicted and actual classes. In Figure 3.3, the confusion matrix results of the model developed within the project are presented. Classes that exhibit confusion with each other typically involve letters that are closely positioned on the IPA chart. This observation suggests that the IPA chart’s reliability is supported by the model’s performance.

3.3. Why audio spectrograms?

Initially, the modeling phase was implemented on radar spectrograms. However, attempts to model the spectrograms produced in two types resulted in failure. Consequently, various experiments were conducted to identify and address the source of the problem.

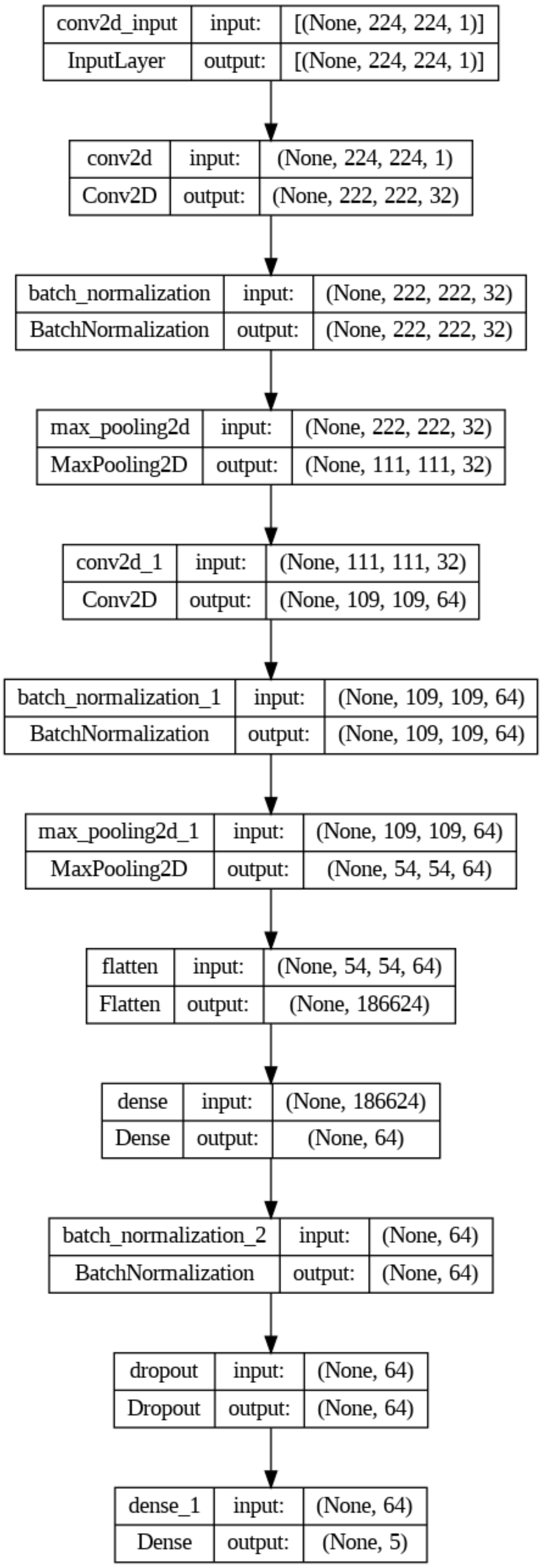

conv2d_input | input: | [(None, 224, 224, 1)]
InputLayer | output: | [(None, 224, 224, 1)]
conv2d | input: | (None, 224, 224, 1)
Conv2D | output: | (None, 222, 222, 32)
batch_normalization | input: | (None, 222, 222, 32)
BatchNormalization | output: | (None, 222, 222, 32)
max_pooling2d | input: | (None, 222, 222, 32)
MaxPooling2D | output: | (None, 111, 111, 32)
conv2d_1 | input: | (None, 111, 111, 32)
Conv2D | output: | (None, 109, 109, 64)
batch_normalization_1 | input: | (None, 109, 109, 64)
BatchNormalization | output: | (None, 109, 109, 64)
max_pooling2d_1 | input: | (None, 109, 109, 64)
MaxPooling2D | output: | (None, 54, 54, 64)
flatten | input: | (None, 54, 54, 64)
Flatten | output: | (None, 186624)
dense | input: | (None, 186624)
Dense | output: | (None, 64)
batch_normalization_2 | input: | (None, 64)
BatchNormalization | output: | (None, 64)
dropout | input: | (None, 64)
Dropout | output: | (None, 64)
dense_1 | input: | (None, 64)
Dense | output: | (None, 5)

Figure 3.1: Architecture of the Model

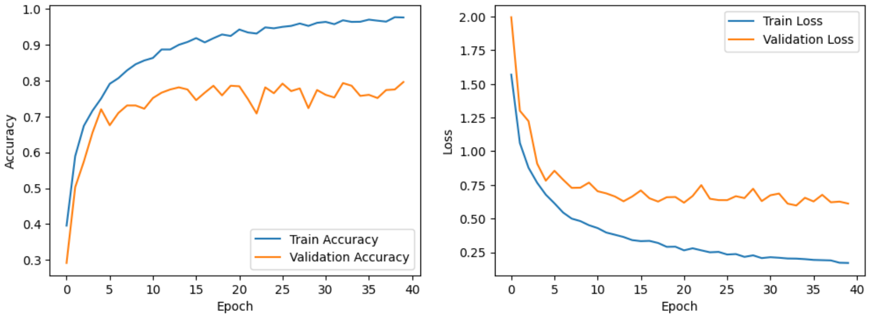
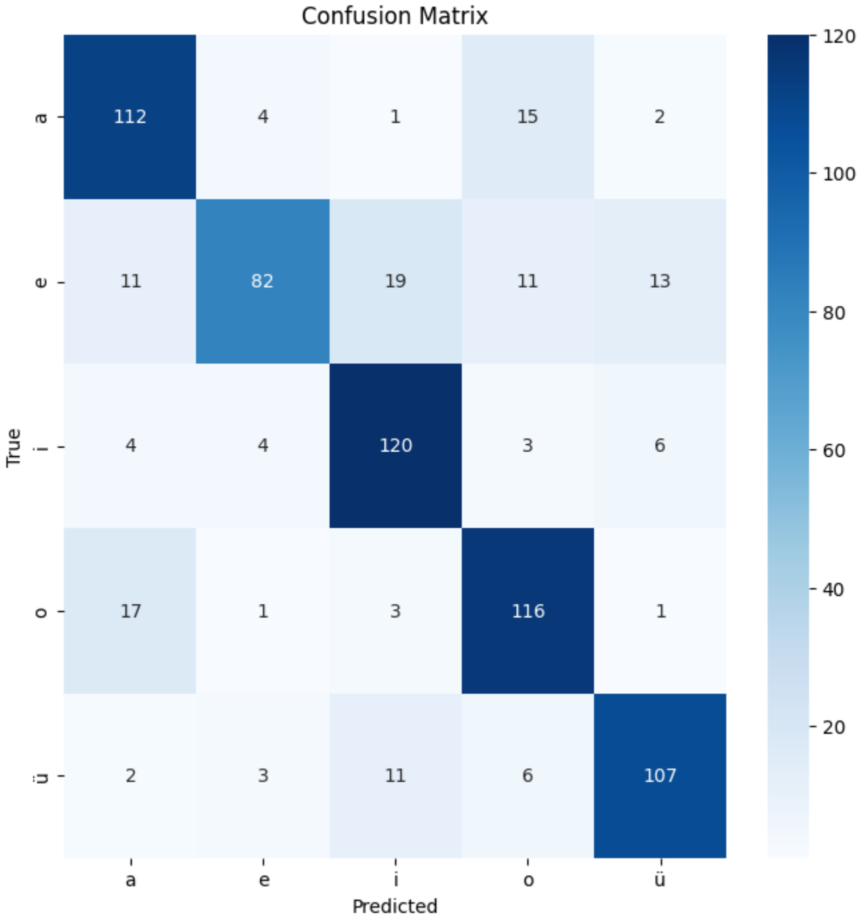


Figure 3.2: Accuracy and loss values



	a	e	i	o	ü
a	112	4	1	15	2
e	11	82	19	11	13
i	4	4	120	3	6
o	17	1	3	116	1
ü	2	3	11	6	107

Figure 3.3: Confusion Matrix

Classification Report Table			
	precision	recall	f1-score
a	0.77	0.84	0.80
e	0.87	0.60	0.71
i	0.78	0.88	0.82
o	0.77	0.84	0.80
ü	0.83	0.83	0.83
accuracy			0.80
macro avg	0.80	0.80	0.79
weighted avg	0.80	0.80	0.79

Table 3.1: Classification Report Table

4. EXPERIMENTS

Several experiments were conducted to ascertain the proper processing of radar signals. The outcomes of these experiments played a pivotal role in gaining insights into the characteristics of radar signals.

4.1. Oscillating a pencil

The pen, suspended by a rope from a high point, is allowed to oscillate parallel to the radar positioned in front of it. Upon reviewing Figure 4.1, the harmonic movement of the pen is clearly evident.

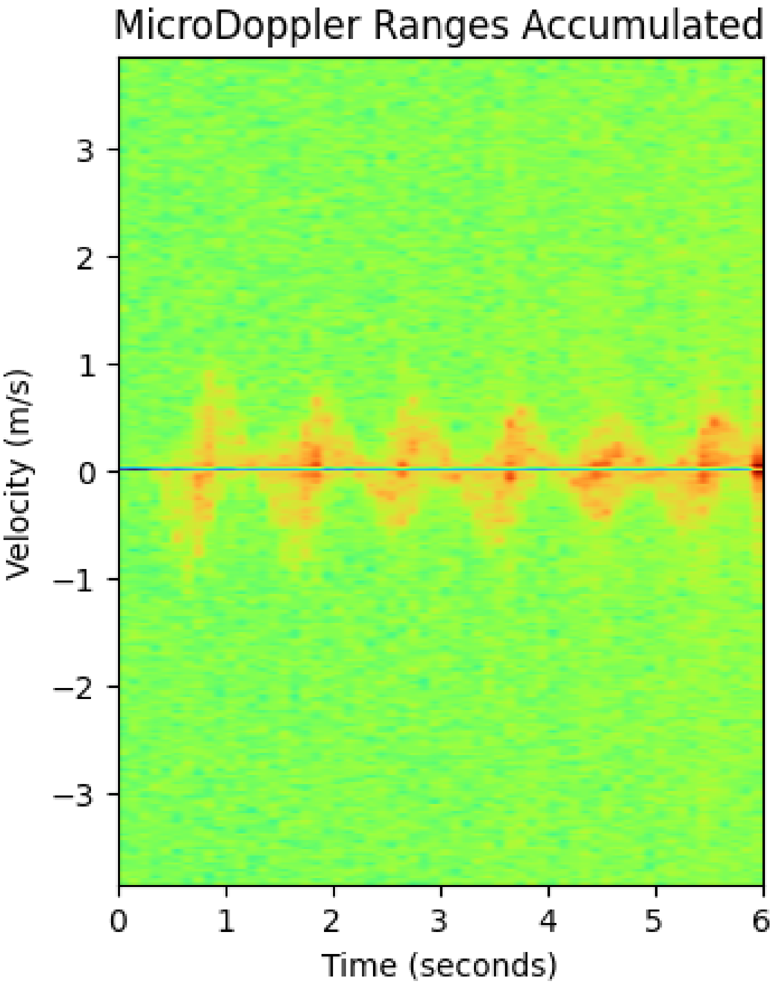


Figure 4.1: Spectrogram of oscillating a pencil.

4.2. Walking movement - 1

A person approached and moved away from the radar in a plane perpendicular to the radar within a 6-second timeframe. As illustrated in Figure 4.2, the value decreased below zero when approaching the radar, indicating a positive velocity value while moving away.

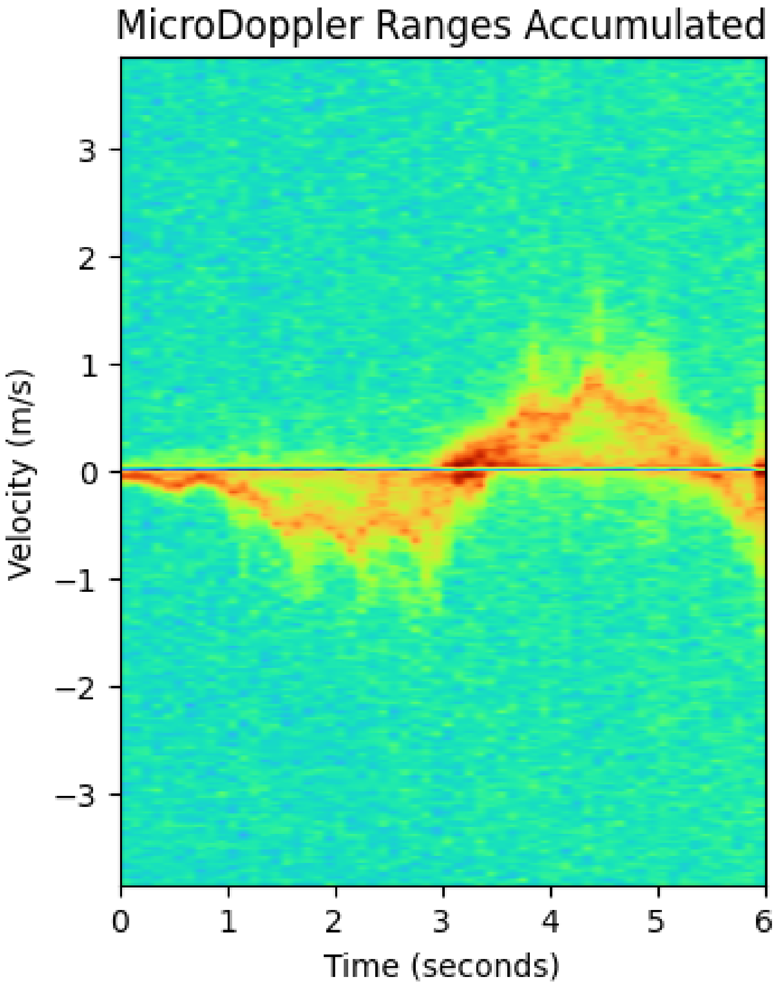


Figure 4.2: Spectrogram of walking movement - 1.

4.3. Walking movement - 2

Two individuals performed an approach-and-recede movement in a plane perpendicular to the radar. Initially, one of the subjects approached the radar and then moved away. Towards the end of the movement of the first subject, the second person repeated the same sequence of actions. As can be seen in Figure 4.3, the movement of two people can be distinguished very clearly.

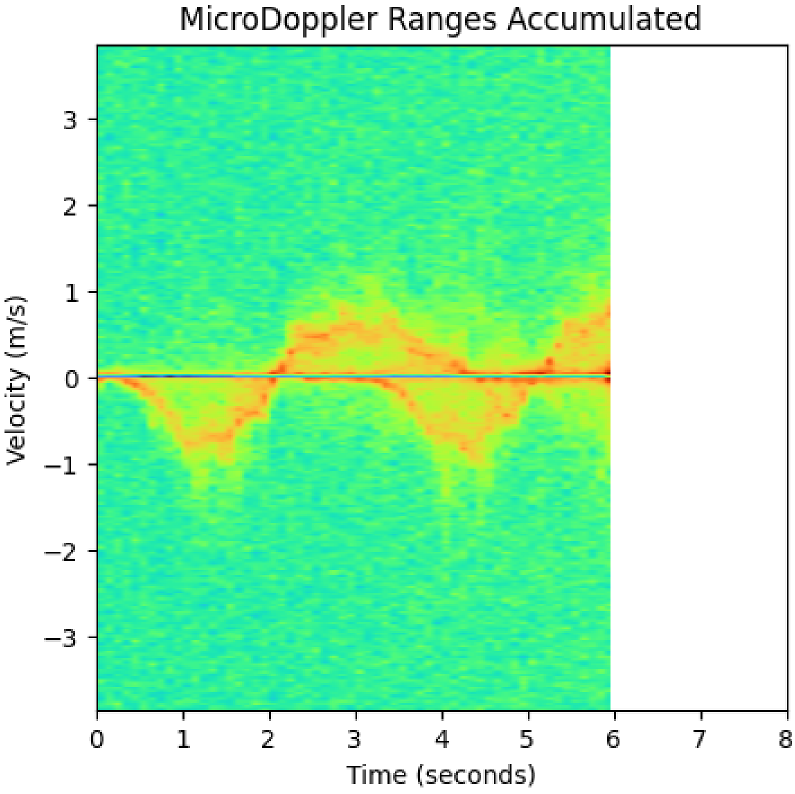


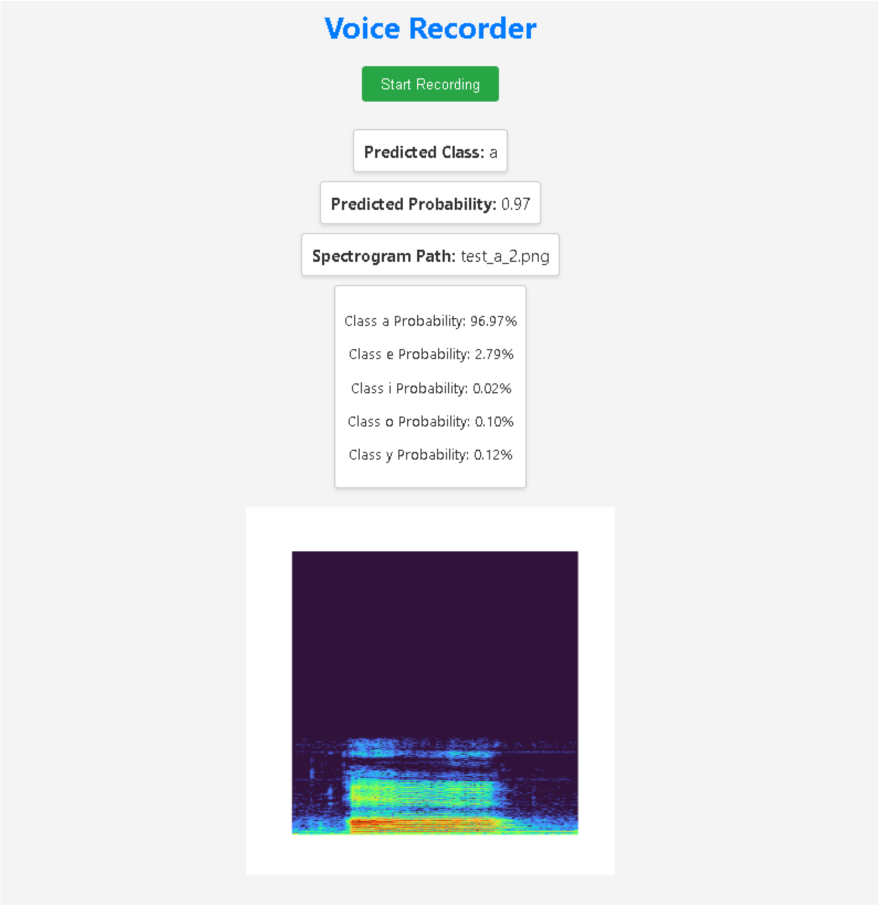
Figure 4.3: Spectrogram of walking movement - 2.

5. CONCLUSION

As a result, it is our observation that lip movement and vocal cord vibration cannot be accurately measured with radar based on the experiments conducted and the data collected. While more precise results were observed in larger movements, no definitive conclusions could be drawn from the radar data we gathered. On the contrary, a model with an 80 percent accuracy rate has been successfully developed using audio data. Upon analyzing the predictions, it has been noted that the model tends to make errors with small differences, with no excessively large inaccuracies in the values it misclassifies. Additionally, the letters that were confused during the modeling phase were found to be close to each other according to the IPA chart.

5.1. Results

You can see some results for voice classification in the Figures below. When we look at the results in general, vowels can be confused with vowels that are close to each other according to the IPA Chart. In the last comparison, the vowel letter ’e’ was perceived as the vowel letter ’ü’. But the probability of it being ’e’ is also close to 24%. These results have also shown us the accuracy and reliability of the IPA Chart.



Voice Recorder

Predicted Class: a

Predicted Probability: 0.97

Spectrogram Path: test_a_2.png

Class a Probability: 96.97%

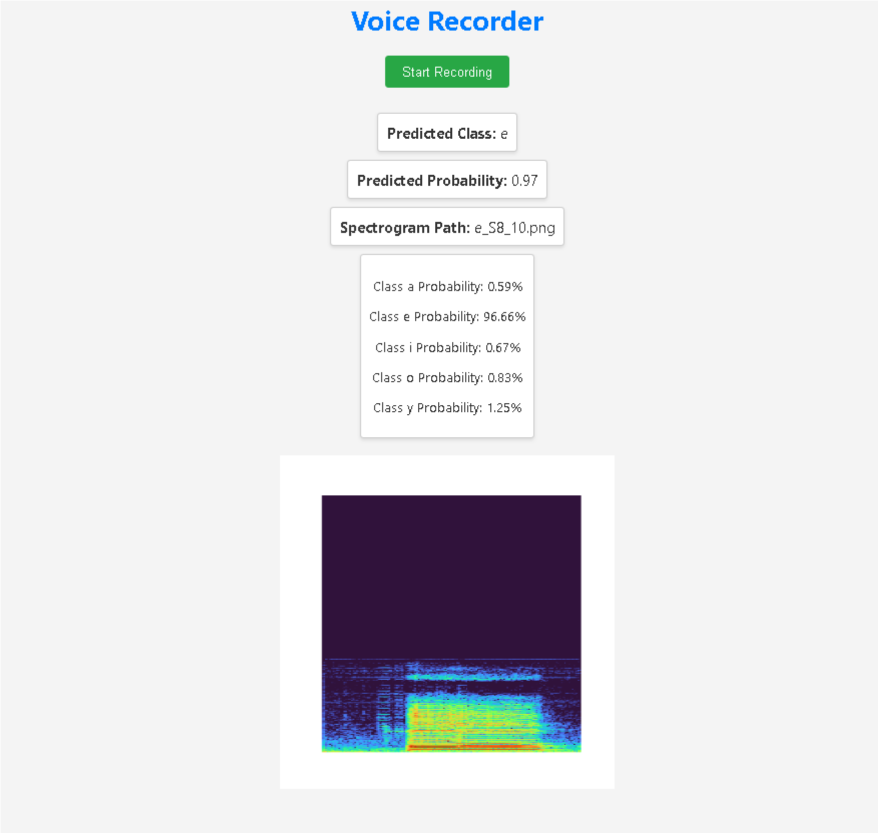
Class e Probability: 2.79%

Class i Probability: 0.02%

Class o Probability: 0.10%

Class y Probability: 0.12%

Figure 5.1: Result of letter ’a’.



Predicted Class: e

Predicted Probability: 0.97

Spectrogram Path: e_S8_10.png

Class a Probability: 0.59%

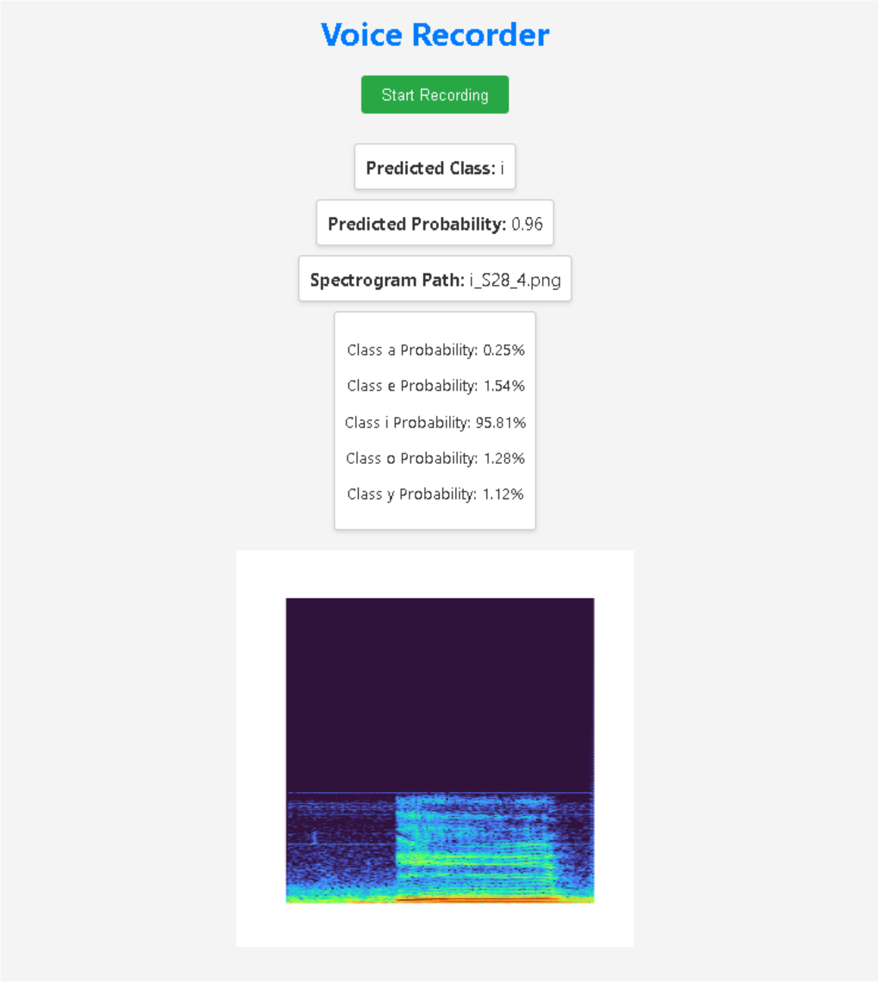
Class e Probability: 96.66%

Class i Probability: 0.67%

Class o Probability: 0.83%

Class y Probability: 1.25%

Figure 5.2: Result of letter ’e’.



Voice Recorder

Predicted Class: i

Predicted Probability: 0.96

Spectrogram Path: i_S28_4.png

Class a Probability: 0.25%

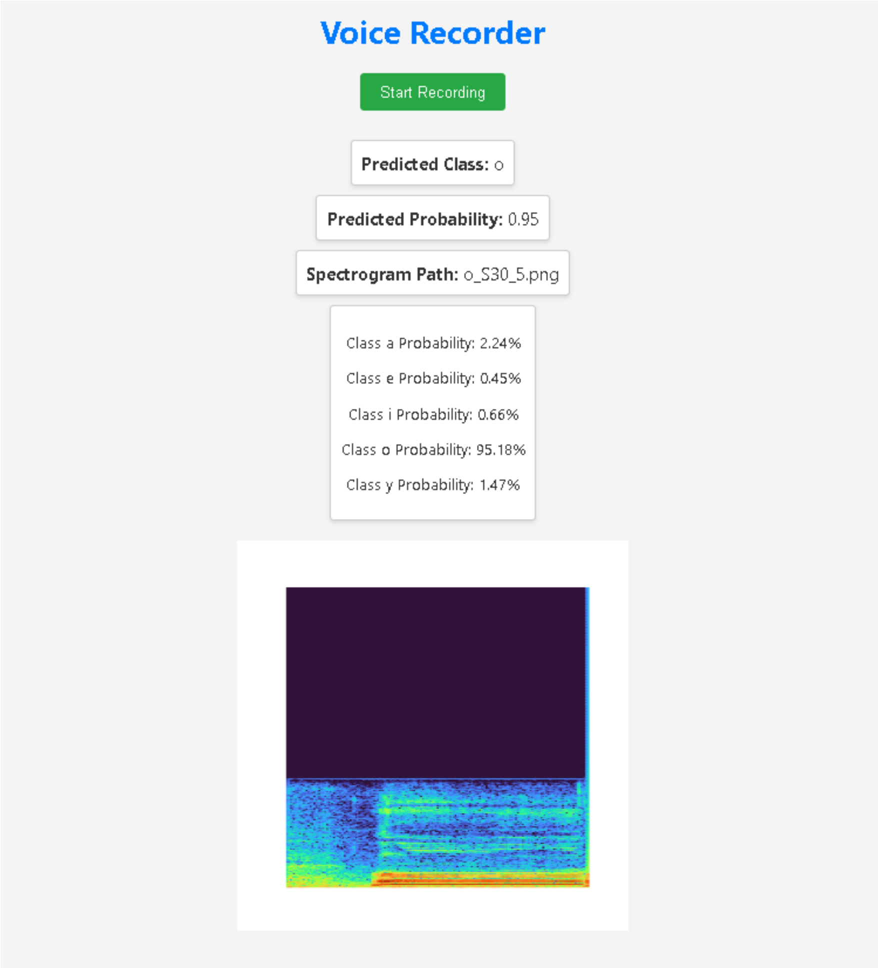
Class e Probability: 1.54%

Class i Probability: 95.81%

Class o Probability: 1.28%

Class y Probability: 1.12%

Figure 5.3: Result of letter 'i'.



Predicted Class: o

Predicted Probability: 0.95

Spectrogram Path: o_S30_5.png

Class a Probability: 2.24%

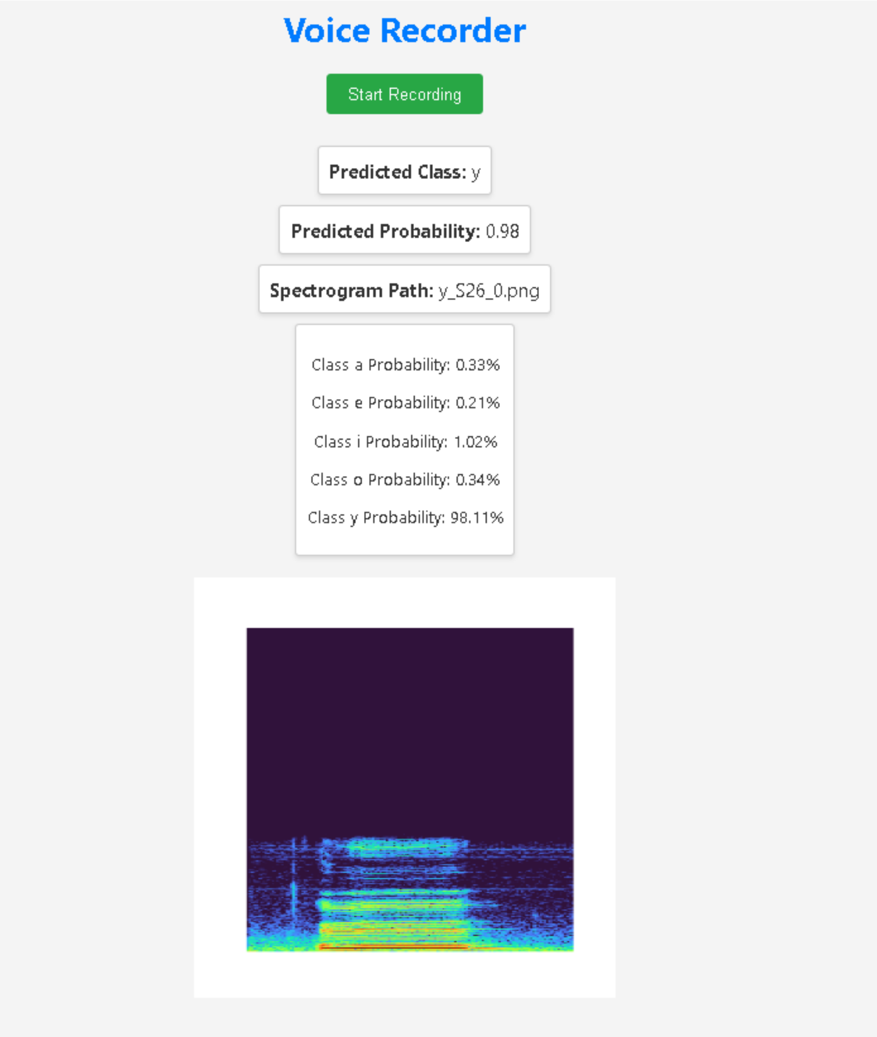
Class e Probability: 0.45%

Class i Probability: 0.66%

Class o Probability: 95.18%

Class y Probability: 1.47%

Figure 5.4: Result of letter 'o'.



Predicted Class: y

Predicted Probability: 0.98

Spectrogram Path: y_S26_0.png

Class a Probability: 0.33%

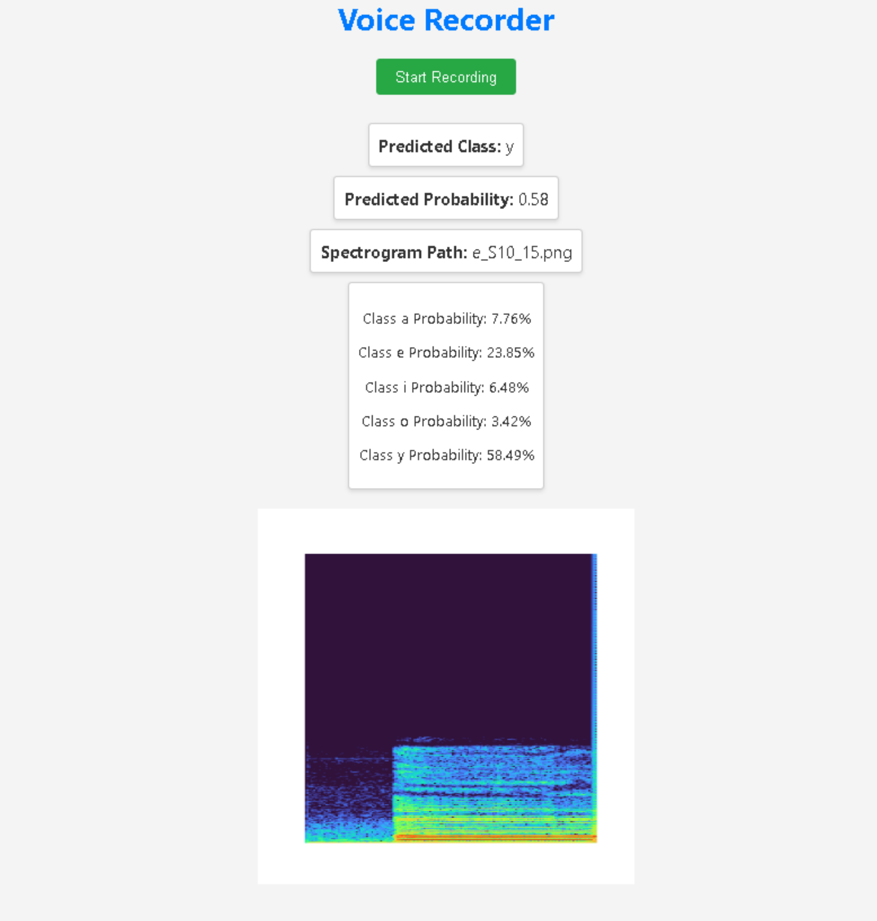
Class e Probability: 0.21%

Class i Probability: 1.02%

Class o Probability: 0.34%

Class y Probability: 98.11%

Figure 5.5: Result of letter 'y'.



Predicted Class: y

Predicted Probability: 0.58

Spectrogram Path: e_S10_15.png

Class a Probability: 7.76%

Class e Probability: 23.85%

Class i Probability: 6.48%

Class o Probability: 3.42%

Class y Probability: 58.49%

Figure 5.6: Result of letter ’e’ - 2.

BIBLIOGRAPHY

[1] X. L. Y. L. Long Fan Lei Xie, “Mmmic:multi-modal speech recognition based on mmwave radar,” 2019.

[2] E. L. Satsuki Nakai David Beavan, “Viewing speech in action: Speech articulation videos in the public domain that demonstrate the sounds of the international phonetic alphabet (ipa). innovation in language learning and teaching,” vol. 3, no. 12, pp. 212–220, 2018.

[3] F. T. A. Sherwin P. Trazo, “International phonetic alphabet (ipa) front vowel sound recognition of beginner foreign learners,” vol. 5, no. 12, 2019.

[4] “Mmmic:multi-modal speech recognition based on mmwave radar,” [Online]. Available: `https://www.researchgate.net/publication/338420504/figure/fig4/AS:845451553153028@1578582766644/IPA-Vowel-Chart-http-wwwinternationalphoneticassociationorg-content-ipa-chart.ppm`.

[5] F. C. David Ferreira Samuel Silva, “Rassper: Radar-based silent speech recognition,” 2021.

CV

Personal Information

- **Name**: Abdullah Muhammet Yiğit
- **Date of Birth**: 14.10.1998
- **Address**: İstasyon Mahallesi, Alemdağ Sokak, Elit Hayat Sitesi H1 Blok No:4/2 Daire:5 Tuzla/İstanbul
- **Phone**: +905422614431
- **Email**: abdullahmygt@gmail.com

Education

- **Bachelor's Degree**: Gebze Technical University, Computer Engineering, 2017-Present
- **High School**: Tokat İMBK Anadolu Öğretmen Lisesi, 2016

Experience

- **Business to Future (B2F)**, Intern, July 2022 - August 2022
- **Business to Future (B2F)**, Developer, August 2022 - February 2023
  - Developing modules under Microsoft Business Central in the ERP domain
  - Preparing reports with Power BI

Skills

Programming Languages:

- Python
- C

- C++
- Java
- Flutter

Software and Technologies:

- Artificial Intelligence
- Mobile Development
- Web Development

Language Proficiency:

- English (Professional)
- Turkish (Native)

APPENDICES

Velocity-Time Spectrogram and Doppler-Range GIF

```
if file_name.endswith(".bin"):

adc_data = np.fromfile(file_path, dtype=np.int16)

adc_data_padded = np.ones(num_frames*4*num_adc_samples*num_chirps_per_frame*2)*1E-8
adc_data_padded[:adc_data.shape[0]] = adc_data
adc_data = adc_data_padded.reshape(num_frames, -1)

adc_data = np.apply_along_axis(DCA1000.organize, 1, adc_data,
num_chirps=num_chirps_per_frame,num_rx=num_rx_antennas, num_samples=num_adc_samples)

dataCube = adc_data

micro_doppler_data = np.zeros((num_frames, num_loops_per_frame, num_adc_samples),
dtype=np.float64)
fig, ax = plt.subplots()
frames = []
for i, frame in enumerate(dataCube):

radar_cube = dsp.range_processing(frame,window_type_1d=Window.BLACKMAN)
assert radar_cube.shape == (
num_chirps_per_frame, num_rx_antennas, num_adc_samples),
"[ERROR] Radar cube is not the correct shape!"

det_matrix , aoa_input = dsp.doppler_processing(radar_cube, num_tx_antennas=2,
clutter_removal_enabled=True,interleaved =False, window_type_2d=Window.HAMMING)

det_matrix_vis = np.fft.fftshift(det_matrix, axes=1)
if plot_range_doppler_flag:

det_matrix_display = det_matrix_vis
plt.title("Range-Doppler plot " + str(i))
plt.imshow(det_matrix_display.T,cmap='turbo')
plt.pause(0.05)
plt.clf()

micro_doppler_data[i,:,:] = det_matrix_vis[:,:]
frames.append([plt.imshow(det_matrix_vis[:,:].T, cmap=’turbo’, animated=True)])

ani = animation.ArtistAnimation(fig, frames, interval=50, blit=True)

write_path = f’/content/drive/My Drive/mmwave_radar_dataset/pervane/T1/’
ani.save(write_path + f’{subfolder}_{file_name}.gif’, writer=’imagemagick’, fps=20)

write_path = f’/content/drive/My Drive/mmwave_radar_dataset/pervane/T1/’
print(write_path)
plt.figure()
plt.imshow(np.sum(micro_doppler_data, axis=1)[:,:].T,origin=’lower’,
extent=(0,chirp_period*micro_doppler_data[:,120,:].shape[0],-
micro_doppler_data[:,120,:].shape[1]*doppler_resolution/2,
micro_doppler_data[:,120,:].shape[1]*doppler_resolution/2))

plt.title("MicroDoppler Ranges Accumulated")
plt.ylabel("Velocity (m/s)")
plt.xlabel("Time (seconds)")

plt.savefig(write_path + f’{subfolder}\\{subfolder}_{file_name}.png’,
bbox_inches = ’tight’,pad_inches = 0.25)
```

Doppler-Range Spectrogram

```
for i, frame in enumerate(dataCube):

from mmwave.dsp.utils import Window

radar_cube = dsp.range_processing(frame, window_type_1d=Window.BLACKMAN)

print("radar_cube.shape: {}".format(radar_cube.shape))

det_matrix, aoa_input = dsp.doppler_processing(radar_cube, num_tx_antennas=2,
interleaved = False, clutter_removal_enabled=False)

print("det_matrix.shape: {}".format(det_matrix.shape))
```

```
det_matrix_vis = np.fft.fftshift(det_matrix, axes=1)

print("det_matrix_vis.shape: {}".format(det_matrix_vis.shape))

print(det_matrix_vis.std())
print(det_matrix_vis.mean())
print(np.median(det_matrix_vis))

min_val = det_matrix_vis.min()
max_val = det_matrix_vis.max()
plt.figure(figsize=(8, 6))
normalized_matrix = np.round((det_matrix_vis - min_val) / (max_val - min_val) * 255)
normalized_matrix = normalized_matrix.astype(np.uint8)
plt.imshow(normalized_matrix.T, aspect='auto')
plt.xticks([])
plt.yticks([])
```

Audio Spectrogram

```
file_path = f'/output_20240125_103750.wav'

y, sr = librosa.load(file_path, sr=None)
D = librosa.amplitude_to_db(np.abs(librosa.stft(y)), ref=np.max)

plt.figure(figsize=(4, 4))
librosa.display.specshow(D, sr=sr, cmap='turbo')
plt.axis('off')

write_path = f'wavs_all/'
!mkdir -p "$write_path"

plt.savefig(write_path + f'deneyy.png')
plt.close()
```