

# Indian Institute of Technology, Guwahati



## Department of Computer Science and Engineering

Project report

On

### **“Small tutorial on Vehicles”**

Based on

**Speech recognition system**

Course: CS566 Speech Processing

Submitted to

Prof. P. K. Das

Submitted by:

Abhijeet Padhy (214101001)

Vanshita Bansal(214101056)

## **TABLE OF CONTENTS**

1. Abstract
2. Introduction
3. Proposed Methodology
4. Flowchart
5. Experimental Setup
6. User Interface
7. Confusion Matrix
8. Result
9. Future Work

# ABSTRACT

This document defines a set of evaluation criteria and test methods for speech recognition systems used in recognizing words spoken. The speech recognition system is based on **Hidden Markov Model** (HMM). This project detects vehicle name and displays its image with some information about it. The users of this software can be anyone.

# INTRODUCTION

The idea is to create a **small version of Wikipedia** where the user **searches the vehicle by uttering its name**, in return the application provides the information about the vehicle with its image. The database can be extended to all type of categories.

The project uses speech recognition programs that are human-computer interactive. Training and Testing speech recognition products for universal usability is an important step before considering the product to be a viable solution for its customers later. This document concerns Speech Recognition accuracy in vehicle name searching and retrieving information, which is a critical factor in the development of hands-free human machine interactive devices. There are two separate issues that we want to test: word recognition accuracy and software friendliness.

But before that we need to know what is **speech recognition**?

You provide speech as input to the system and the system converts this speech to text. This text is processed and its features are analyzed properly to give a prediction. The system may predict it correctly or may fail in doing so as well. Our task is to create a speech recognition system that is as accurate as possible.

The new voice recognition systems are certainly much easier to use. You can speak at a normal pace without leaving distinct pauses between words. However, you cannot really use “natural speech” as claimed by the manufacturers. You must speak clearly, as you do when you speak to a Dictaphone or when you leave someone a telephone message. Remember, the computer is relying solely on your spoken words. It cannot interpret your tone or inflection, and it cannot interpret your gestures and facial expressions, which are part of everyday human communication. Some of the systems also look at whole phrases, not just the individual words you speak. They try to get information from the context of your speech, to help work out the correct interpretation.

# **PROPOSED METHODOLOGY**

Basic requirements:

- ✓ Windows OS
- ✓ Microsoft Visual Studio 2010
- ✓ C++ 11 integrated with VS2010
- ✓ Recording Module

With the availability of above software, we further proceed in modelling the logic.

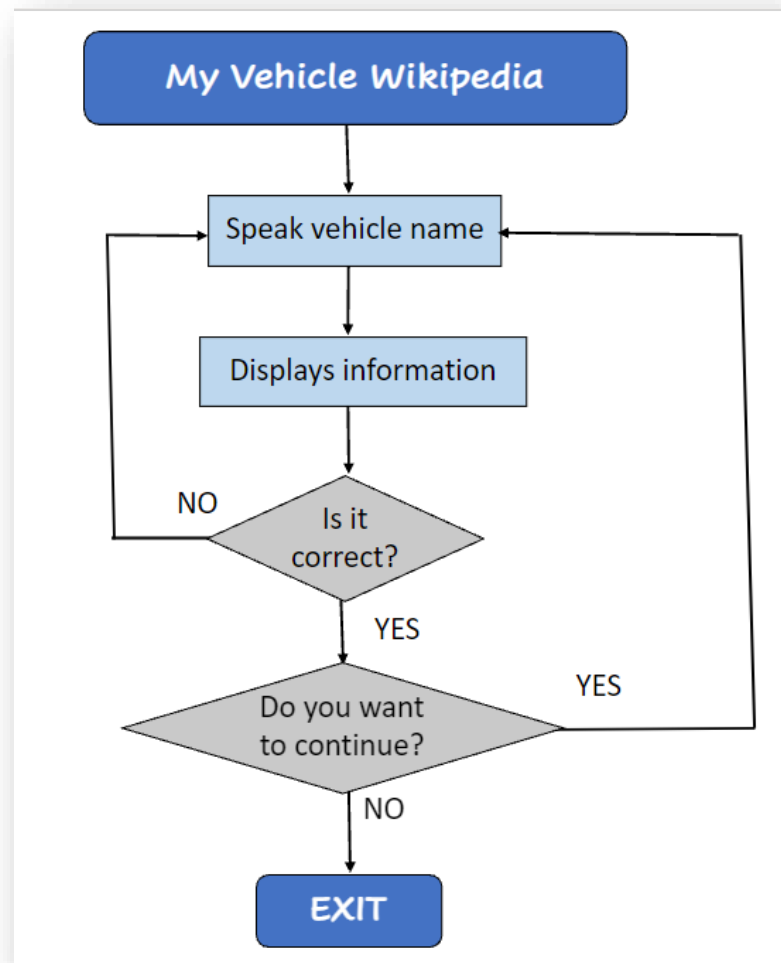
The prerequisites of this project are:

- ✓ Basic I/O operations on file
- ✓ Pre-processing of speech data
- ✓ Feature extraction
- ✓ Modelling of extracted feature
- ✓ Enhancing model

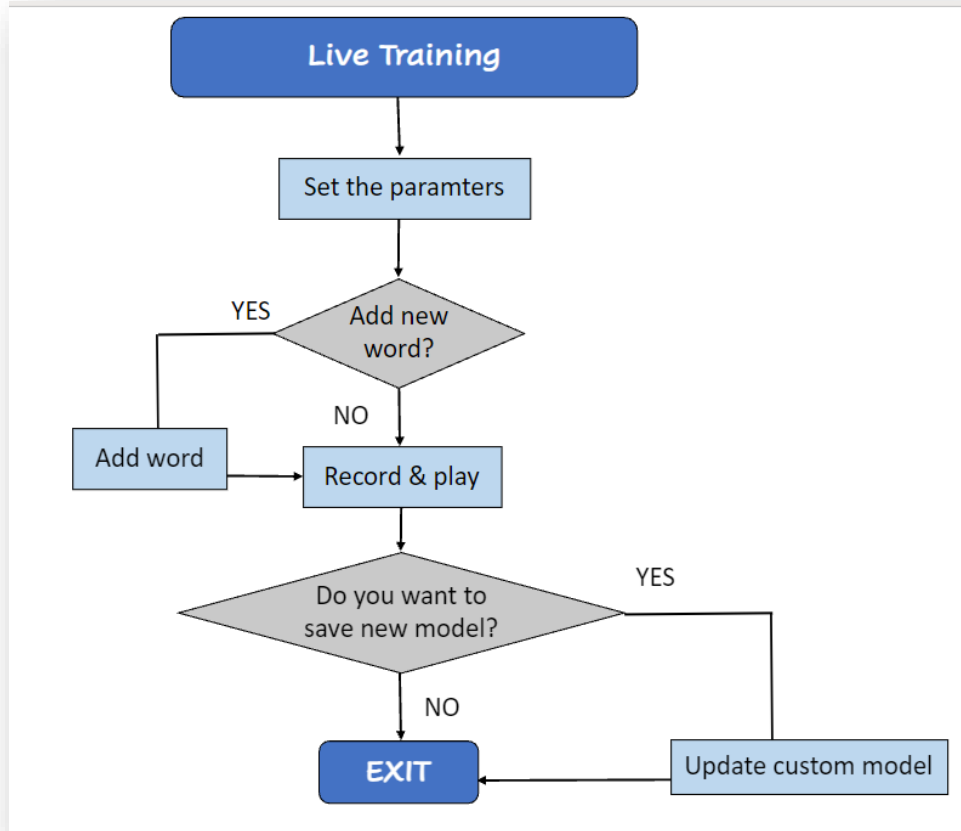
Above discussed topics are broadly elaborated in experimental setup section.

# FLOWCHART

With the availability of above tools, we further proceed. Below is the flow chart for our project.



## Flowchart for Live Training



## **EXPERIMENTAL SETUP**

This project is divided into following modules:

1. Training Module
2. Testing Module
3. Live Training Module

### **1. Training Module**

The flow for training over data is as follows:

- a. Record 20 utterances for each word included in project
- b. Extract frames of speech part from these recordings.
- c. Using local distance analysis (using vector quantization) to find the observation sequence.
- d. Pass this observation sequence to HMM for model designing by using Bakis model as initial model.
- e. Now enhance the model using HMM re-estimation algorithm (Baum welch).

Now reference models for each word are ready for our project. The training of data is not integrated with GUI application. This is different module which will just evaluate reference model.



## **2. Testing Module**

You need search the word by speaking into the microphone and its data will be displayed on screen.

The flow of testing is as follows:

- a. Live recording of data is done on clicking on search, it displays the spoken word's waveform.
- b. Testing is done on the data with pretrained models.
- c. Verifying the vehicle name detected with user.
- d. If verification is successful display details.
- e. If verification fails, you need to record the input again.

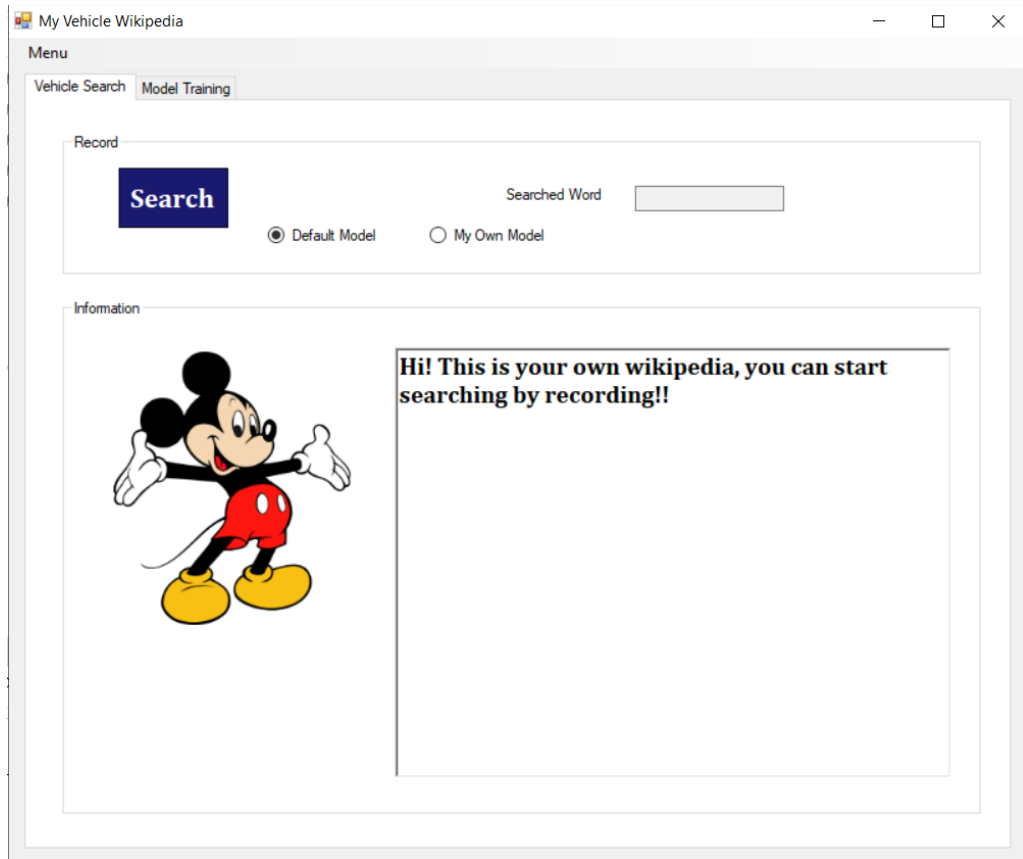
## **3. Live Training Module**

The flow for live training over data is as follows:

- a. Fill the utterance count of each new word you want to record.
- b. Record these utterances in one go.
- c. Extract frames of speech part from these recordings.
- d. Using local distance analysis (using vector quantization) to find the observation sequence.
- e. Pass this observation sequence to HMM for model designing by using Bakis model as initial model.
- f. Now enhance the model using HMM re-estimation algorithm (Baum welch).

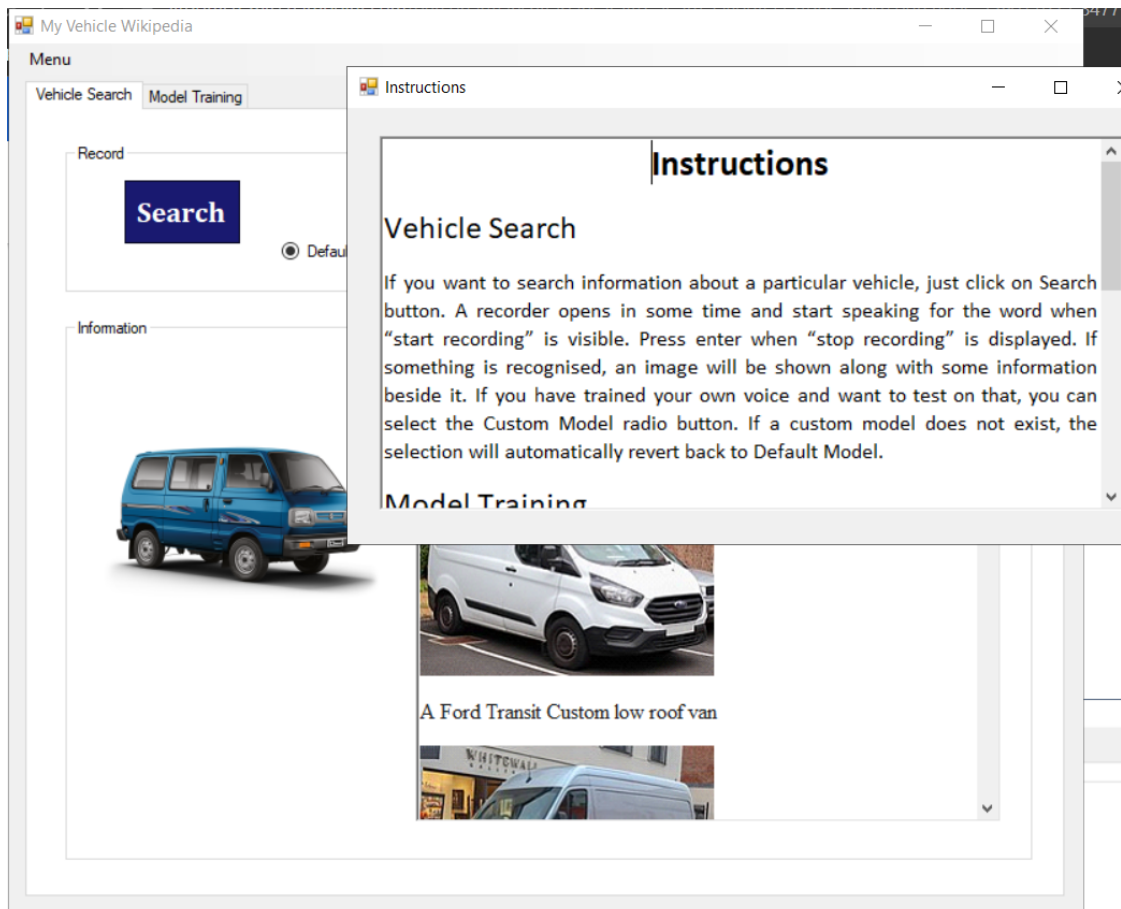
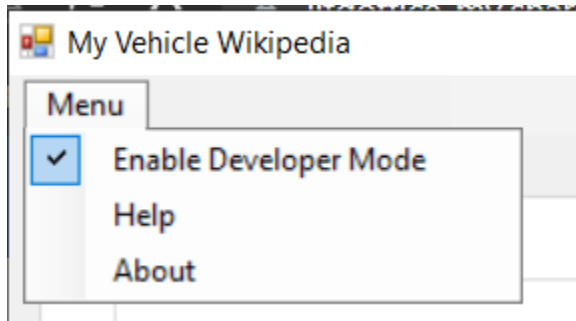
Now the reference models for each word are ready which can be saved as custom models and can be used for later updating to default model. This training of data is integrated with GUI application.

# USER INTERFACE



On clicking search button you will be able to record the word and after recording the output image and information will be shown with the word shown in right.

## Menu:



My Vehicle Wikipedia

Menu

Vehicle Search


Model Training

Record

Search

Searched WordVan


Information




About

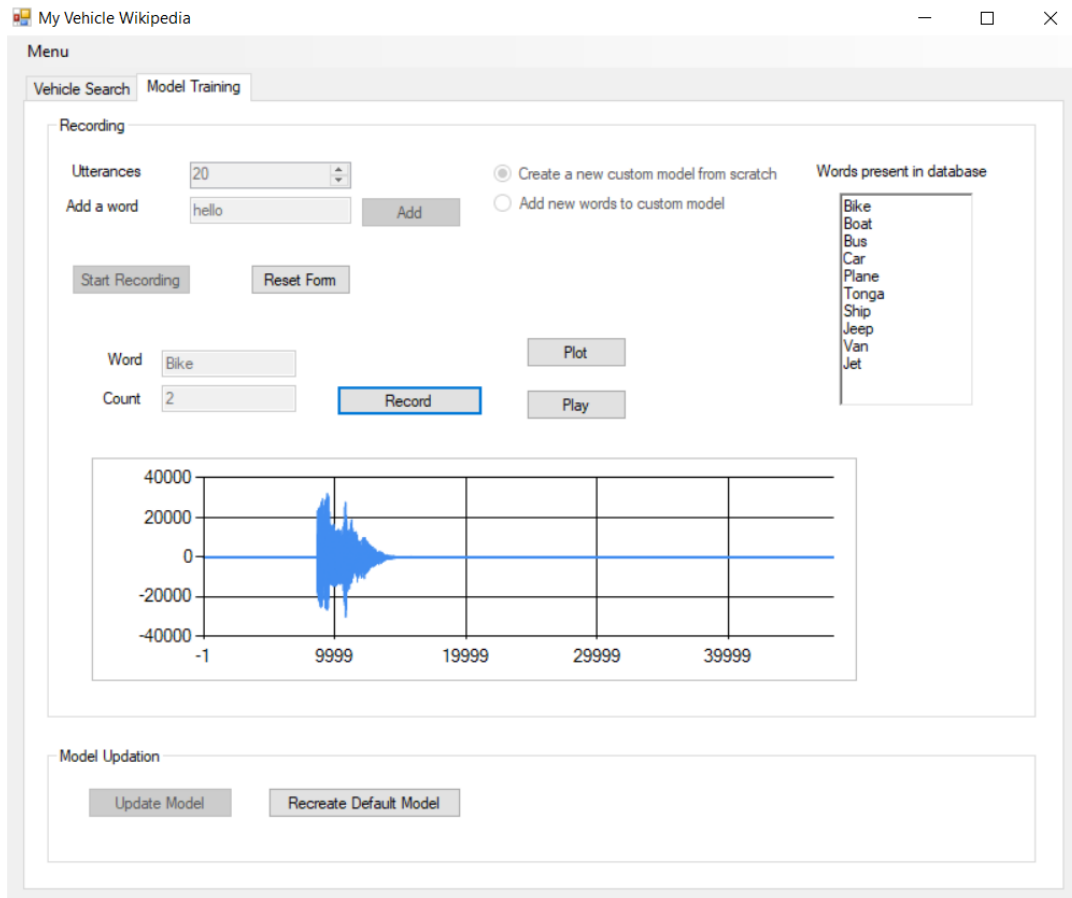
Developed in IIT Guwahati by

- Abhijeet Padhy(214101001)
- Vanshita Bansal(214101056)



A Ford Transit Custom low roof van





This interface is used to do the live training part. Developer can add new words and can also create models for these words. This model is saved as custom model. We can switch to default model again using recreate default model option.

You can see the plot of word simultaneously while recording one by one to check the waveform. Words are added to the database and can be included in application after updating current model.

## CONFUSION MATRIX

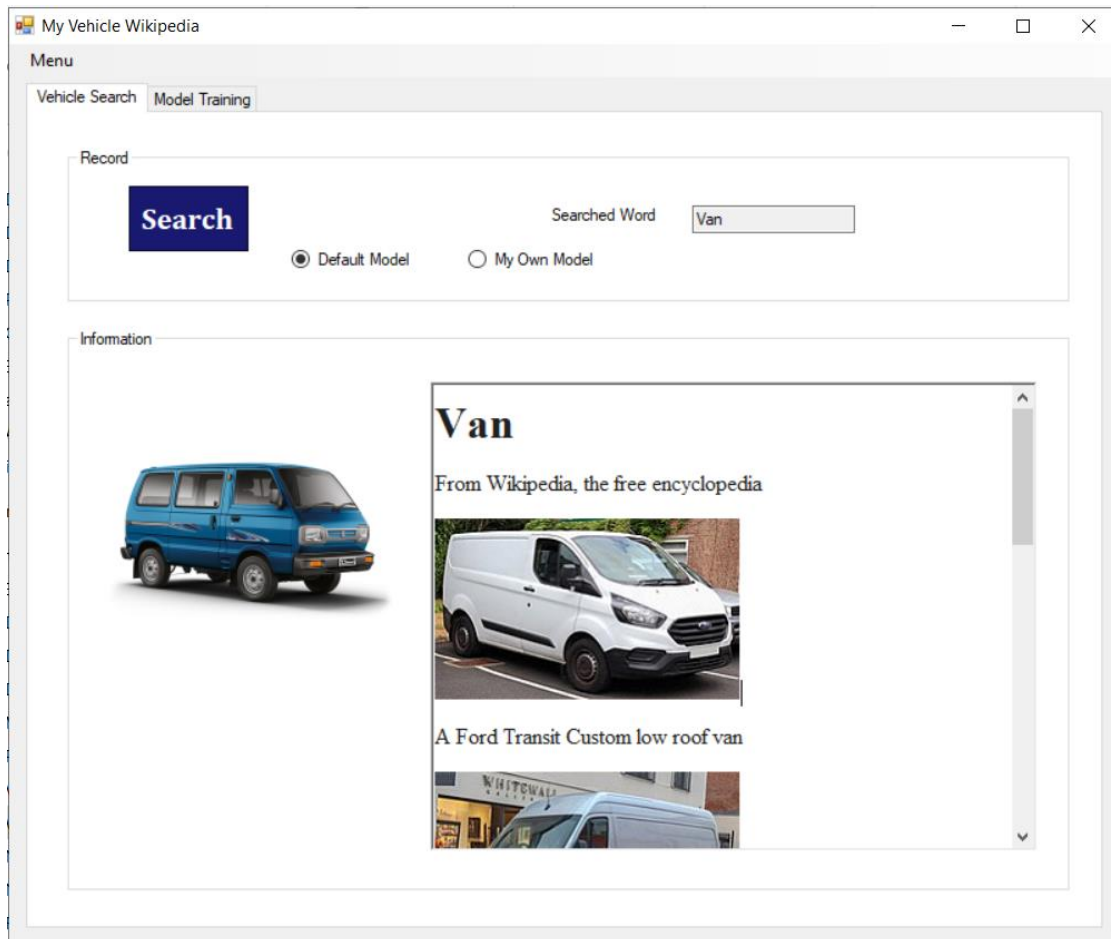
	Bike	Boat	Bus	Car	Plane	Tonga	Ship	Jeep	Van	Jet
Bike	1	0	0	0	0	0	0	0	0	0
Boat	0	0	0	1	0	0	0	0	0	0
Bus	0	0	1	0	0	0	0	0	0	0
Car	0	0	0	1	0	0	0	0	0	0
Plane	0	0	0	0	1	0	0	0	0	0
Tonga	0	0	0	0	0	1	0	0	0	0
Ship	0	0	0	0	0	0	0	1	0	0
Jeep	0	0	0	0	0	0	0	1	0	0
Van	0	0	0	0	0	0	0	0	1	0
Jet	0	0	0	0	0	0	0	0	0	1

The green cells represent correct prediction whereas red cells represent wrong prediction. Row fields represent input word and column fields represent predicted word.

These results were obtained in live testing.

## RESULT

We have details of 10 vehicles which are already stored in a data file. Fetching of data based on speaker's requirement is successfully implemented.



## **FUTURE WORK**

1. Add an option of Re-recording previous word, if we want to.
2. Various categories can be included other than vehicle name.
3. And at large this can be implemented as voice enabled Wikipedia.
4. We can add the facility to upload new images and data to database.