

Fayoum University
Faculty of Engineering
Communication Engineering Dep



SMART INTERACTIVE E-TEACHER

Pet Life Project



Team member:

1. EsraaMoumenAbdelfatah
2. Heba Ahmed Mohamed

Approvedby :

Associate Professor /Amr M. Gody

Department of Electrical engineering

Abstract:

A work presented on Electronic learning for children. Speech recognition method is utilized to guide the child into an interactive story to teach him the intended learning entities.

Acknowledgments:

The author wishes to express sincere appreciation to Professor Amr for his assistance in the preparation of this project.

Several people have been instrumental in allowing this project to be completed. We would like to thank Prof. Dr. Soliman Mahmoud who accepted us as a group. We would also like to thank especially Eng/ Mohammed Hamdy, Eng/ Ahmed Sultan. Teaching Assistant, for his encouragement and patience from the initial to the final level enabled us to develop an understanding of the work for our project.

Our parents didn't know what we were doing, but they were always eager to help us out in all possible ways; without them it is hard to imagine accomplishing all this work. (Special thanks to our families). Finally, we take this opportunity to express how much we were really good friends all the time of the project without any problems.

The spirit we had is the cause why we completed the project in this manner, that's why we must congratulate ourselves for the cooperation, patience and Insistence to represent a good abstract for what we learned in the college.

TABLE OF CONTENTS

List of Figures	ii
List of Tables	iii
Preface	iv
Introduction.....	1
1. Chapter I: Introduction	2
1.1Statement of Problem	3
1.2Purpose of E-Learning	3
1.3Description of Terms	5
2.Chapter II: Conceptual Framework	12
2.1Static class diagram	13
2.2Interactivity scenario	21
3.Chapter III: Methodology	40
3.1Automatic Speech Recognition	41
3.2SpeechLib	43
3.3C# programming for Multimedia applications	50
3.4Analysis of Data	57
4.Chapter IV: Quick User Guide.....	60
4.1Use cases	63
4.2Basic Settings	65
4.3Training Mode	67
4.4Interactive Mode	69
4.5Step by step tutorial	71
5.Glossary	73
6.Bibliography	75
7.Appendix A: HTK tools.....	77

8.Appendix B: Arabic phonemes International Phonetic Alphabets (IPA)
..... 78

9.Appendix C: Audience Statistics 79

10.Pocket Material: CD of Application and Documentation.

LIST OF FIGURES

Number	<i>Page</i>
1. System block diagram	12
2. Automatic Speech Recognition.....	13
3. Main screen.....	14
4. Training Screen.....	16
5. SpeechLib static class diagram	17
6. Histogram Rank of Application Simplicity	18
7. Static class diagram.....	21
8. Training Sequence Diagram	24
9. Active mode sequence Diagram.....	28
10.Histogram Rank of Complete session	36

CHAPTER 1**1. INTRODUCTION****1.1. Statement of Problem**

Teaching children the basic entities need a lot of patients and talent from the teacher. Some of the problems may be listed as

- 1- Keeping child in focus for long period.
- 2- Avoid daily pressure from affecting teacher's efficiency.

1.2.Purpose of E-Learning

E- Learning stands for Electronic Learning. The purpose of it is to enhancing the bad impact of the mentioned problems in the previous section. Using the unlimited domain of multimedia effects will bring the full focus of child to be a maximum. Having child to interact with an attractive animated story will make it very efficient to conduct the ideas directly to his brain.

Having the electronic system as a teacher will ensure keeping the same efficiency. The human part will be limited to feed new attractive stories to the system.

1.3. Description of Terms

This system is intended to be free of charge. It is an open source that can be used without any conditions in the intended purposes which is created for. In case of using this system in any commercial purposes, it is required to get the approval from Fayoum University as owner of this system.

CHAPTER 2**2.1. Interactivity scenario:**

1-Child will choose interactive mode button from the home page of the screen shot.

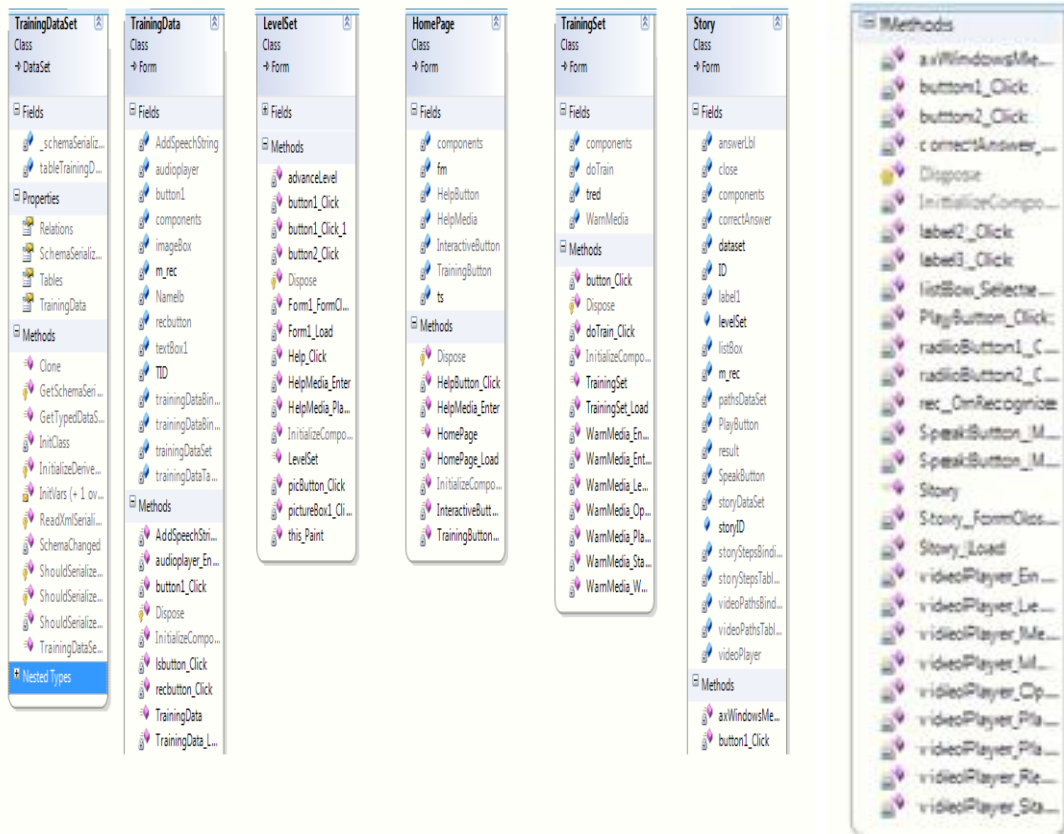
2-He will listen to interesting story at the opening of the screen shot , he must focus all his attention to remember when we ask him.

3-He will find several levels to pass to the next story but he must start with level one as it will be the only active level

4- When he start this level we will show him the story as separated parts , after each part he will find a question if his answer and pronunciation is right he will continue to the next part, if it was wrong the same part of the story will be replayed to help him correcting answer and pronunciation.

5-when he finish level one level two will be activated and the child will be allowed to start it and answer our questions and so on.

2.2 Static Class Diagram



FIG(2.1) STATIC CLASS DIAGRAM

CHAPTER 3

3. METHODOLOGY

This chapter gives a full description of how the Kinyarwanda language speech recognition System was developed. The goal of the project was to build a robust whole word recognizer. That means it should be able to generalize both from speaker specific properties and its Training should be more than just instance based learning.

In the HMM paradigm this is supposed to be the case, but the researcher intended to put this into practice.

As the time scope was limited and to be able to focus on more specific issues than HMM in general, the Hidden Markov Model toolkit (HTK) was used.

HTK is a toolkit for building Hidden Markov Models. HMMs can be used to model any time series and the core of HTK is similarly general purpose. However, HTK is primarily designed for building HMM based speech processing tools, in particular recognizers.

Secondly to reduce the difficulties of the task, a very limited language model was used.

Future research can be directed to more extensive language models. In ASR systems acoustic

Information is sampled as a signal suitable for processing by computers and fed into a recognition process.

The output of the system is a hypothesis transcription of the utterances. A speech recognition system performs three primary tasks as shown in Fig(3.1)

Preprocessing:

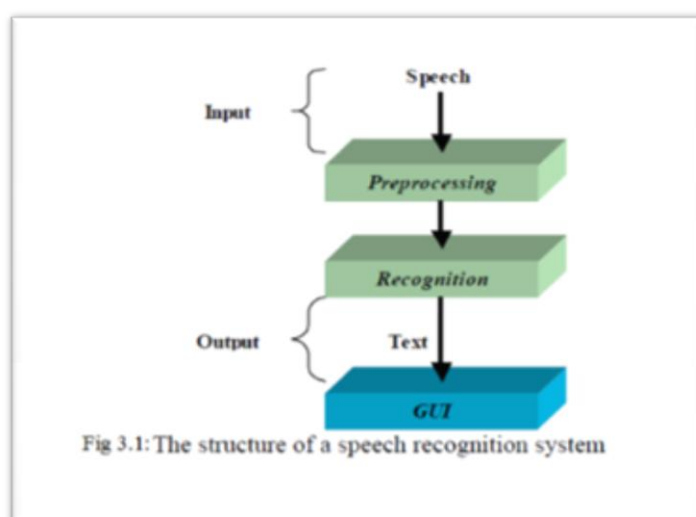
Converts the spoken input into a form the recognizer can process.

Recognition:

Identifies what has been said by comparing the input with the built models.

Communication:

sends the recognized input to the software systems that needs it.



Fig(3.1):The structure of speech recognition system

3.1. Automatic Speech Recognition Introduction:

3.1.1, Speech Recognition Basics:

Speech recognition is the process by which a computer or other type of machine identifies spoken words. Basically, it means talking to your computer, AND having it correctly recognize what you are saying.

The following definitions are the basics needed for understanding speech recognition technology.

i. Utterance

An utterance is the vocalization or speaking of a word or words that represent a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences.

ii. Speaker Dependence

Speaker dependent systems are designed around a specific speaker. They generally are more accurate for the correct speaker, but much less accurate for other speakers. They assume the speaker will speak in a consistent voice and tempo.

Speaker independent systems are designed for a variety of speakers. Adaptive systems usually start as speaker independent systems and utilize training techniques to adapt to the speaker to increase their recognition accuracy.

iii. Vocabularies

Vocabularies or dictionaries are lists of words or utterances that can be recognized by the SR system. Generally, smaller vocabularies are easier for a computer to recognize, while larger vocabularies are more difficult. Unlike normal dictionaries, each entry doesn't have to be a single word. They can be as long as a sentence or two. Smaller vocabularies can have as few as 1 or 2 recognized utterances (e.g. "Wake Up"), while very large vocabularies can have a hundred thousand or more!

iv. Accurate

The ability of a recognizer can be examined by measuring its accuracy or how well it recognizes utterances. This includes not only correctly identifying an utterance but also identifying if the spoken utterance is not in its vocabulary. Good ASR systems have an accuracy of 98% or more. The acceptable accuracy of a system really depends on the application.

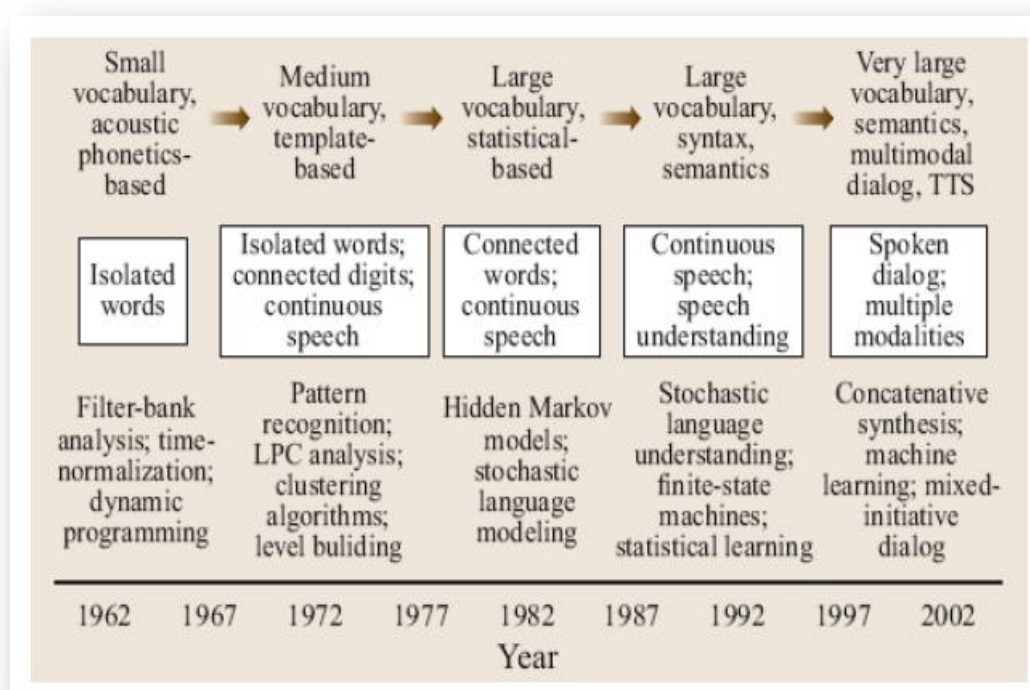
v. Training

Some speech recognizers have the ability to adapt to a speaker. When the system has this ability; it may allow training to take place. An ASR system is trained by having the speaker repeat standard or common phrases and adjusting its comparison algorithms to match that particular speaker.

Training a recognizer usually improves its accuracy. Training can also be used by speakers that have difficulty speaking, or pronouncing certain words. As long as the speaker can consistently repeat an utterance, ASR systems with training should be able to adapt.

3.1.2.Types of Speech Recognition:

Speech recognition systems can be separated in several different classes by describing what types of utterances they have the ability to recognize. These classes are based on the fact that one of the difficulties of ASR is the ability to determine when a speaker starts and finishes an utterance. Most packages can fit into more than one class, depending on which mode they're using.



Fig(3.2):Milestones in speech recognition and understanding technology over the past 40 years

i. Isolated Words

Isolated word recognizers usually require each utterance to have quietlack of an audio signal on BOTH sides of the sample window.

It doesn't mean that it accepts single words, but does require a single utterance at a time.

Often these systems have Listen/ Not Listen states where they require the speaker to wait between utterances usually doing processing during the pauses. Isolated Utterance might be a better name for this class.

ii. Connected Words

Connect word systems or more correctly 'connected utterances' are similar to Isolated words, but allow separate utterances to be 'run together' with a minimal pause between them..

iii. Continuous Speech

Continuous recognition is the next step .Recognizers with continuous speech capabilities are some of the most difficult to create because they must utilize special methods to determine utterance boundaries. Continuous speech recognizers allow users to speak almost naturally, while the computer determines the content Basically, it's computer dictation.

iv. Spontaneous Speech

There appears to be a variety of definitions for what spontaneous speech actually is .At a basic level, it can be thought of as speech that is natural sounding and not rehearsed .An ASR system with spontaneous speech ability should be able to handle a variety of natural speech features such as words being run together, "ums "and "ahs", and even slight stutters.

v. Voice Verification/Identification

Some ASR systems have the ability to identify specific users. This document doesn't cover verification or security systems.

3.1.3. Uses and Applications:

Although any task that involves interfacing with a computer can potentially use ASR, the following applications are the most common right now.

i. Dictation

Dictation is the most common use for ASR systems today.This includes medical transcriptions, legal and business dictation, as well as general word processing. In some cases special vocabularies are used to increase the accuracy of the system

ii. Command and Control

ASR systems that are designed to perform functions and actions on the system are defined as Command and Control systems .Utterances like ."Open Netscape "and "Start a new xterm "will do just that .

iii. Telephony

Some PBX/Voice Mail systems allow callers to speak commands instead of pressing buttons to send specific tones.

iv. Wearables

Because inputs are limited for wearable devices, speaking is a natural possibility.

v. Medical/Disabilities

Many people have difficulty typing due to physical limitations such as repetitive strain injuries RSI, muscular dystrophy, and many others. For example, people with difficulty hearing could use a system connected to their telephone to convert the caller's speech to text.

vi. Embedded Applications

Some newer cellular phones include C&C speech recognition that allows utterances such as Call Home. This could be a major factor in the future of ASR and Linux. Why can't I talk to my television yet.

vii. Military

- 1- High-performance fighter aircraft
- 2- Helicopters
- 3- Battle management
- 4- Training air traffic controllers

viii. Further applications

- ✓ Automatic translation;
- ✓ Automotive speech recognition)e.g., Ford Sync;
- ✓ Telematics.e.g .vehicle Navigation Systems;

- ✓ Court reportingReal time Voice Writing;
- ✓ Hands-free computing:voice command recognition computer user interface;
- ✓ Home automation;
- ✓ Interactive voice response;
- ✓ Mobile telephony, including mobile email;
- ✓ Multimodal interaction;
- ✓ Pronunciation evaluation in computer-aided language learning applications;
- ✓ Robotics;
- ✓ Video games, with Tom Clancy's End War and Lifeline as working examples;
- ✓ Transcriptiondigital speech-to-text;
- ✓ Speech-to-text transcription of speech into mobile text messages;
- ✓ Air Traffic Control Speech Recognition.

3.1.4.What is the Benefit of ASR?

There are fundamentally three major reasons why so much research and effort has gone into the problem of trying to teach machines to recognize and understand speech:

- ✓ Accessibility for the deaf and hard of hearing.
- ✓ Cost reduction through automation.
- ✓ Searchable text capability.

3.1.5. Future Directions

In 1992, the U.S .National Science Foundation sponsored a workshop to identify the key research challenges in the area of human language technology, and the infrastructure needed to support the work. Research in the following areas for speech recognition were identified

I. Robustness:

In a robust system, performance degrades gracefully rather than catastrophically as conditions become more different from those under which it was trained.

Differences in channel characteristics and acoustic environment should receive particular attention.

II. Portability:

Portability refers to the goal of rapidly designing, developing and deploying systems for new applications.

At present, systems tend to suffer significant degradation when moved to a new task.

In order to return to peak performance, they must be trained on examples specific to the new task, which is time consuming and expensive.

III. Adaptation:

How can systems continuously adapt to changing conditions(new speakers, microphone, task, etc)and improve through use. Such adaptation can occur at many levels in systems, sub word models, word pronunciations, language models, etc.

IV. Language Modeling:

Current systems use statistical language models to help reduce the searchspace and resolve acoustic ambiguity.

As vocabulary size grows and other constraints are relaxed to create more habitable systems, it will be increasingly important to get as much constraint as possible from language models; perhaps incorporating syntactic and semantic constraints that cannot be captured by purely statistical models.

V. Confidence Measures:

Most speech recognition systems assign scores to hypotheses for the purpose of rank ordering them.

These scores do not provide a good indication of whether a hypothesis is correct or not, just that it is better than the other hypotheses.

As we move to tasks that require actions, we need better methods to evaluate the absolute correctness of hypotheses.

VI. Out-of-Vocabulary Words:

Systems are designed for use with a particular set of words, but system users may not know exactly which words are in the system vocabulary. This leads to a certain percentage of out of vocabulary words in natural conditions. Systems must have some method of detecting such out of vocabulary words, or they will end up mapping a word from the vocabulary onto the unknown word, causing an error.

VII. Spontaneous Speech:

Systems that are deployed for real use must deal with a variety of spontaneous speech phenomena, such as filled pauses, false starts, hesitations, ungrammatical constructions and other common behaviors not found in read speech. Development on the ATIS task has resulted in progress in this area, but much work remains to be done.

VIII. Prosody:

Prosody refers to acoustic structure that extends over several segments or words. Stress, intonation, and rhythm convey important information for word recognition and the user's intentions (e.g. sarcasm, anger). Current systems do not capture prosodic structure. How to integrate prosodic information into the recognition architecture is a critical question that has not yet been answered.

IX. Modeling Dynamics:

Systems assume a sequence of input frames which are treated as if they were independent. But it is known that perceptual cues for words and phonemes require the integration of features that reflect the movements of the articulators, which are dynamic in nature. How to model dynamics and incorporate this information into recognition systems is an unsolved.

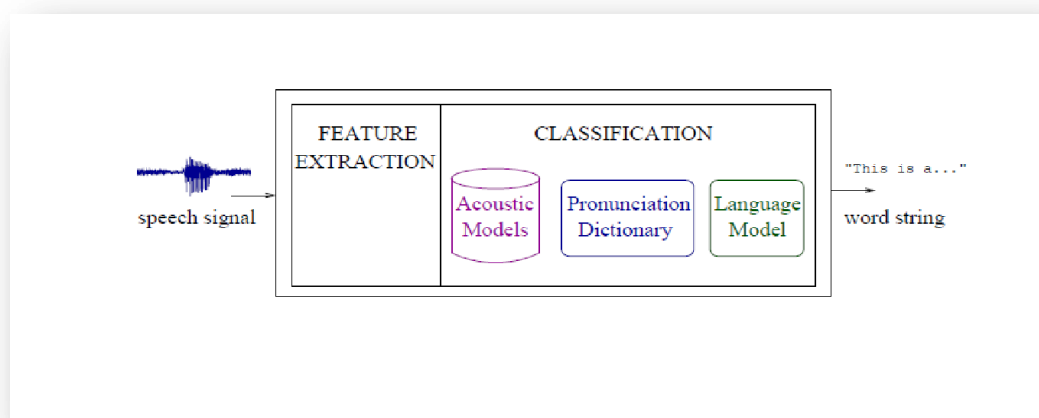
3.1.6. Inside Speech Recognition:

3.1.6.1 Components of ASR

In automatic speech recognition (ASR) systems acoustic information is sampled as a signal suitable for processing of the utterance. Speech recognition is a complicated task and state of -the- art recognition systems are very complex. There are a big number of different approaches for the implementation of the components

Here we only want to provide an overview over ASR, some of its main difficulties, the basic components, their functionality and interaction.

Figure(3.3) shows the main components of an ASR system. In the first step, the Feature Extraction, the sampled speech signal is parameterized. The goal is to extract a number of parameters 'features' from the signal that has a maximum of information relevant for the following classification.



Fig(3.3):Principle components of an ASR system

For implementation purposes the following sub

Processes were taken:

- i. Building the task grammar .
- ii. Constructing a dictionary for the models .
- iii. Recording the data .
- iv. Creating transcription files for training data .
 - v. (feature processing) Encoding the data .
- vi. Retraining the acoustic models) .
- vii. Evaluating the recognizers against the test data .
- viii. Reporting recognition results .

3.1.7. Problems of Speech recognition

1.Accuracy of recognition

Accuracy of recognition can be thought as converting words spoken by a user accurately to its corresponding text. As per information available, almost all ASR engine has high accuracy for detecting two words, YES and NO.

Apart from them, other words like numbers, date of births etc has lower

Accuracy in recognizing

2.Different accent:

Using ASR in a big country where people speak different languages or same language with different accents, speech recognition accuracy is bound to fare worse.

3.Confirming YES/NO again and again is irritating:

People may design IVR very intelligently to confirm for any doubtful word recognition by YES or NO, but it is still irritating for many people and it slows time of fetching information

3.1.8 Arabic language, speech recognition and template matching

Linguistically speaking, Arabic language does not have a normalized form that is used in all circumstances of speech and writing .

Arabic used in daily informal communication is not the same form of Arabic that is used in books, magazines, newspapers and on TV to broadcast the news. While writing Arabic in text materials is standardized and is the same in the entire Arab world, there is no standardization for Arabic that is spoken informally.

This lack of standardization and lack of rules caused the spoken Arabic to be considerably varietal from one region to another.

The forms of Arabic are as follows:

i. Classical or formal Arabic :

Is the old form of the language. It can be seen in the Jaheliah poetry.

ii. Modern Standard Arabic (MSA):

Is a version of classical Arabic with modernized vocabulary. It is considered to be formal language that is common in all Arabic speaking countries. Modern standard Arabic is the form of Arabic used in all written texts .

iii. Colloquial or dialectical Arabic

There are many different dialects that differ considerably from each other and from the Modern Standard Arabics. According to colloquial Arabic can be divided into two groups :Western Arabic and Eastern Arabic.

iv. Lebanese Colloquial Arabic

ArabicLebanese colloquial Arabic is the spoken Arabic used by the Lebanese people in oral communication.

3.2.2 Difficulties with Arabic Speech Recognition

Some of the difficulties encountered by a speech recognition system that are related to the Arabic language are :

i. Word knowledge:

Speech is not just acoustic sound patterns additional knowledge, as word meanings ,is needed in order to recognize exactly the intended speech . Therefore, words with widely different meanings may share the same sequence of sound patterns.For example:

The word ' أَلَّ ' that means exhausted and the word ' أَلَّا ' that means no or” never

The word ' جَرَّ ' that means to drag, the word ' جَرَّى ' that means to make something to stream and the word

' جَرَّة ' that means a jar.

ii. Variability caused by dialectal differences

Variability in dialect between Arab countries and even dialectal difference in the same country causes the word to be pronounced in a different way. This variability in word pronunciation might cause an error in recognition. An example of dialectal difference between Arab countries speakers in Egypt pronounce the phoneme 'ج' in word "جمال" as the letter g in 'get'. While speakers in Lebanon pronounce the phoneme similar to the letter j in 'jar'

An example of dialectal difference in the same country people that live in Beirut spell the word

'أنا' as 'أني'

iii. Coarticulation effects

The acoustic realization of a phoneme may heavily depend on the acoustic context in which it occurs. This effect is usually called coarticulation.

Thus, the acoustic feature of a phoneme is affected by the neighboring phonemes, the position of a phoneme in a word and the position of this word in a sentence. Such acoustic features are very different from those of isolated phonemes, since the articulatory organs do not move as much in continuous speech as in isolated utterances.

We can see the effect of coarticulation in the following phrase "و في الأيام"

Here the phoneme "ي"

In the word "في"

is affected by the neighboring phoneme

and the phoneme "ل" and "ف"

in the word "الأيام"

Therefore the acoustic realization is different from the stand alone phoneme.

iv. **Diacritization**

Diacritics that are described in the section 3 play an important part in written Arabic material. The absence of diacritics in most Arabic texts causes many ambiguities in the pronunciation of words .Therefore, a speaker using an Automatic Speech Recognition system (ASR) while reading form an ondiacritized

source might cause him to mispronounce some words thus causing errors in recognition the diacritic variation for the word ‘Some of

رَحِمَ' are :', 'رُحِمَ"رَحِمَ. , رَحِمَ', رَحِمَ'

v. **Morphology**

The Arabic language is morphologically rich, thus causing a high vocabulary growth rate.This high growth rate is problematic for language models by causing a large number of out –of-vocabulary words

Papers address the effect of morphology on Arabic language speech recognition systems.

3.2 Speech library

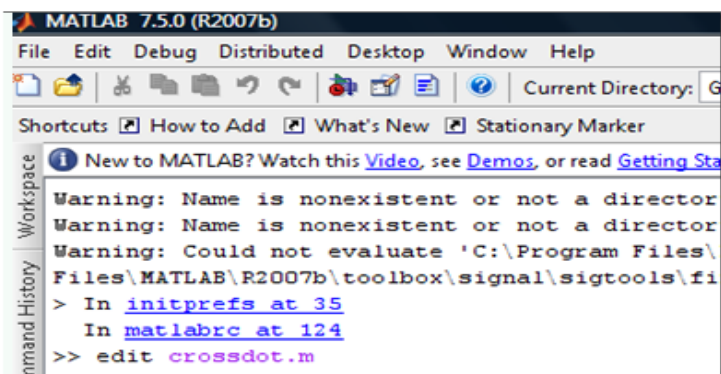
Microsoft C# interfacing with MatLab

In this article I'll illustrate how it could be interfaced to MatLab engine from a .net application .Microsoft C sharp (C#)will be used for this illustration. The following are the steps

Step1 :Packaging the needed MatLab functions into the Class library.

Write a function named crossdot in Matlab editor .

From the Matlab prompt write

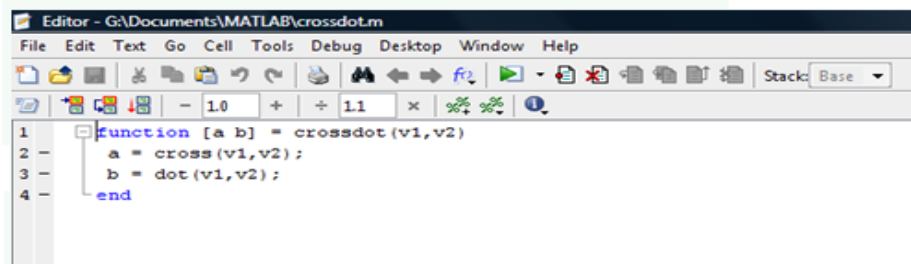


```

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Shortcuts How to Add What's New Stationary Marker
New to MATLAB? Watch this Video, see Demos, or read Getting Started
Warning: Name is nonexistent or not a directory
Warning: Name is nonexistent or not a directory
Warning: Could not evaluate 'C:\Program Files\
Files\MATLAB\R2007b\toolbox\signal\sigtools\fi
> In initprefs at 35
In matlabrc at 124
>> edit crossdot.m

```

Write the function as indicated below

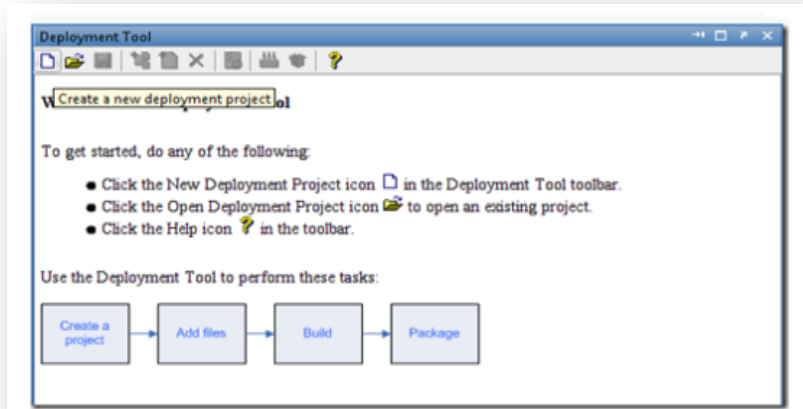


```

Editor - G:\Documents\MATLAB\crossdot.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
1 function [a b] = crossdot(v1,v2)
2     a = cross(v1,v2);
3     b = dot(v1,v2);
4     end

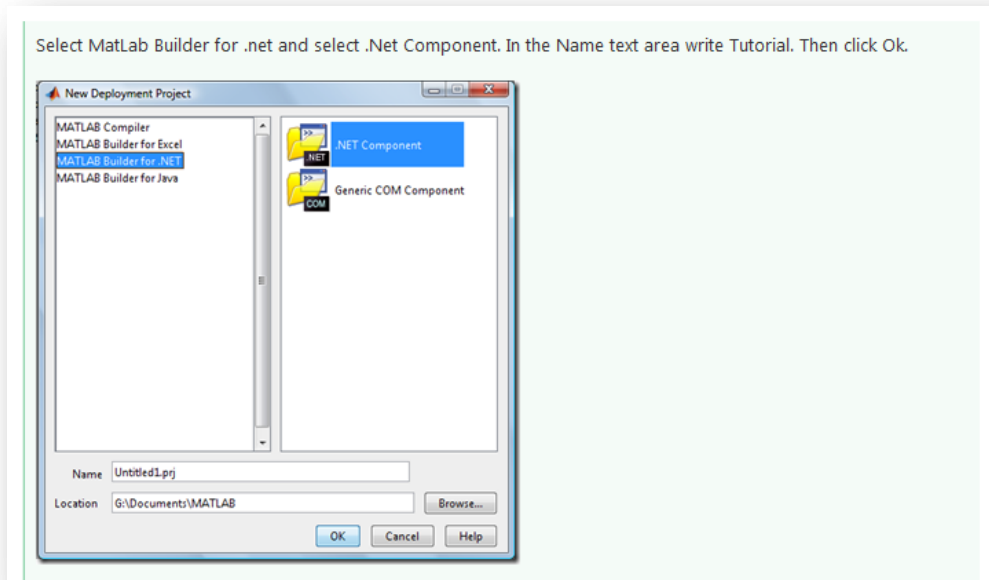
```

Fig(3.4):snapshot 1

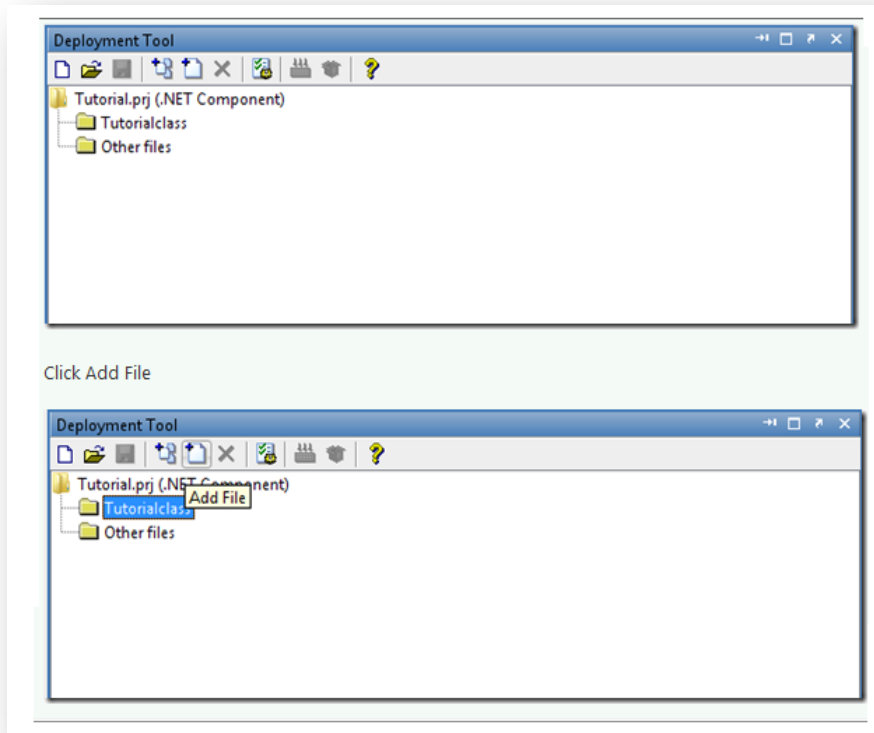


Fig(3.5):snapshot 2

Deploytool.  Then click on New

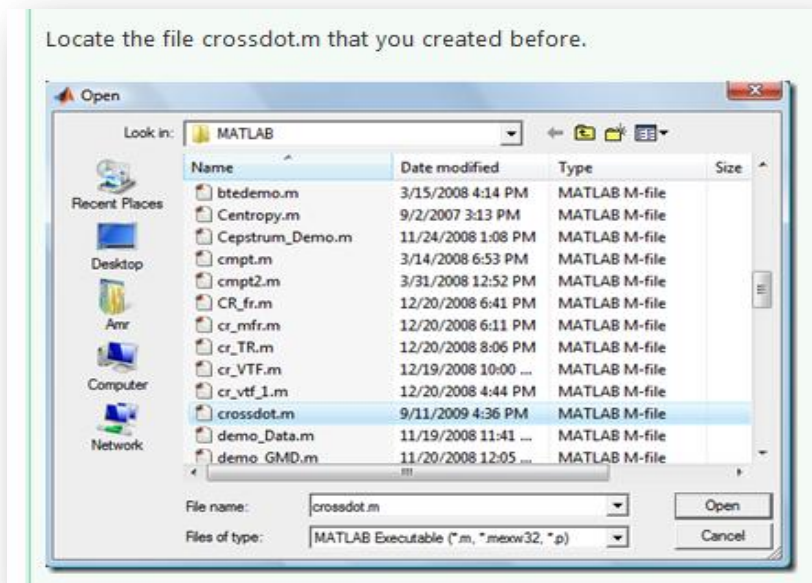


Fig(3.6):snapshot3

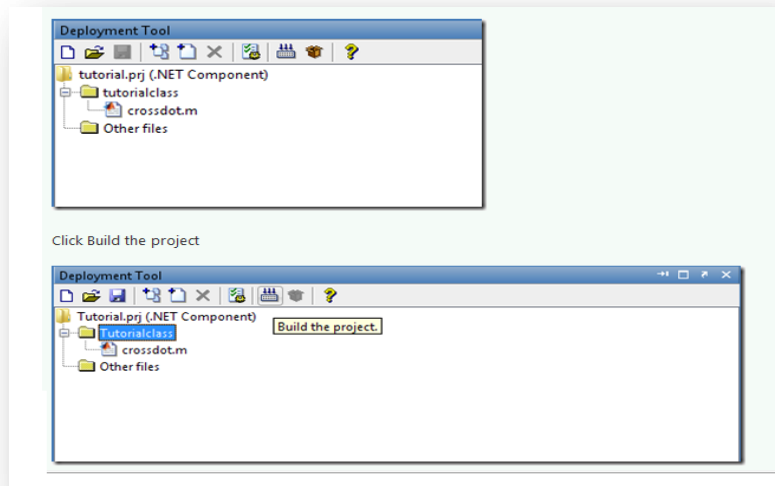


Click Add File

Fig(3.7):snapshot 4

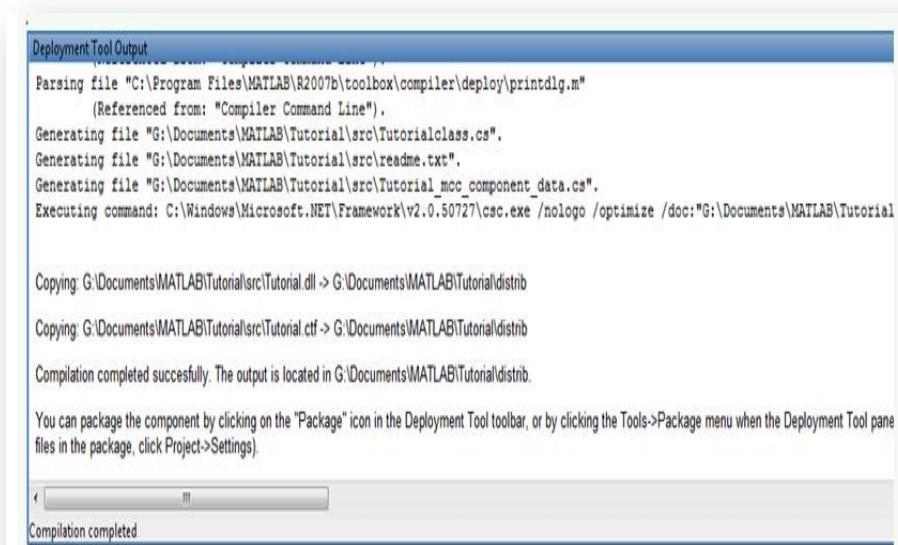


Fig(3.8):snapshot 5



Fig(3.9):snapshot 6

The class library is created at the indicated path as shown in the figure . Remember the path as you will need it to locate the class library DLL file later on from Microsoft Visual Studio C# development environment

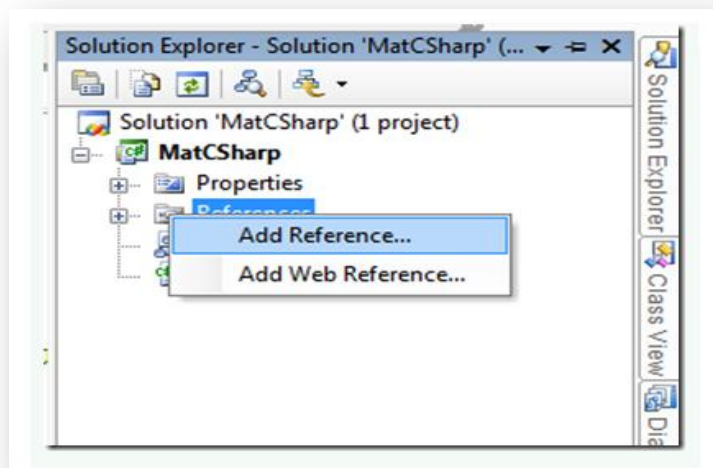


Fig(3.10):snapshot 7

➤ Step2 :Add a reference to Matlab Arrays.

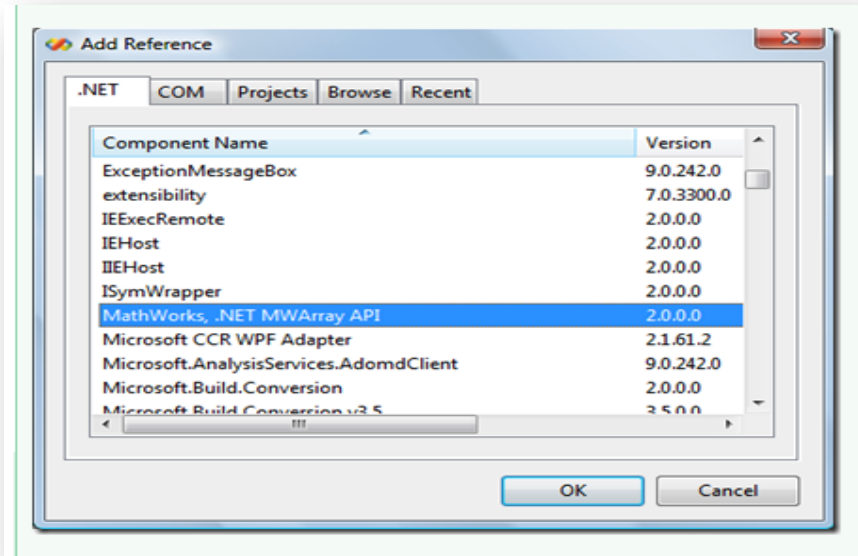
Create a C# project .you may create a Console application project in this tutorial to avoid any complexity in dealing with GUIcomponents.

Go to solution explorer, Right click reference node, then select AddReference



Fig(3.11):snapshot 8

Locate Mathworks, >NET MWArray API class library as shown below .
Then Click OK.

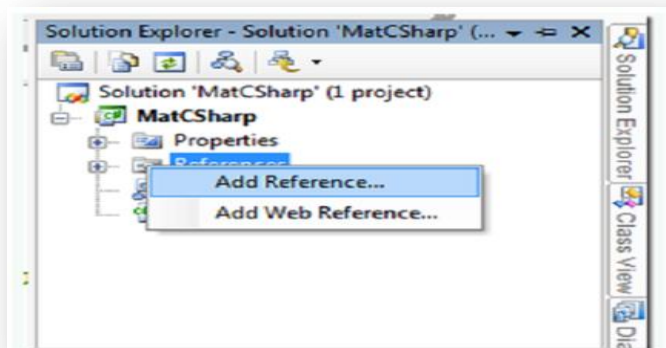


Fig(3.12):snapshot 9

The MWArray Reference should appear under references node as shown below.

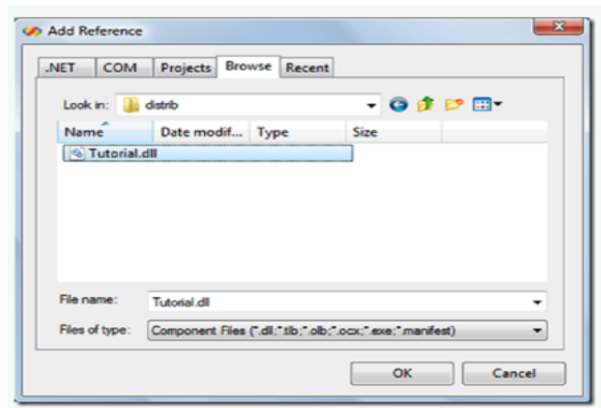
Step3 :Add a reference to Matlab Class library.

Go to solution explorer, Right click reference node, then select Add Reference



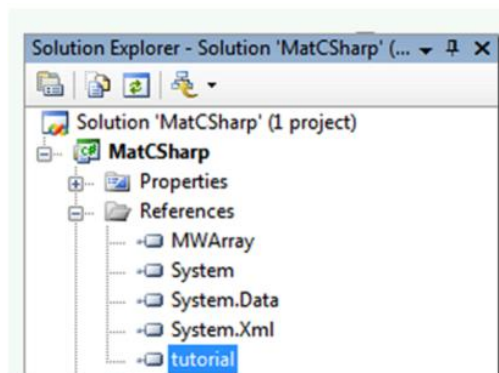
Fig(3.13):snapshot 10

Locate the Tutorial DLL that we created using MatLab .Then click OK.



Fig(3.14):snapshot 11

The Tutorial reference should appear .You may check it in the solution explore when expanding reference node.



Fig(3.15):snapshot 12

Step4 :Using Matlab arrays to pass function variables to the Class library.

Add the namespaces as shown below.

```
using MathWorks.MATLAB.NET.Arrays;  
using tutorial;  
namespace MatCSharp  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
              
        }  
    }  
}
```

Fig(3.16):snapshot 13

Define two arrays to hold data .Those arrays will be passed later on to the Matlab function that will apply the cross and the dot products on them.

Also make an instance of the tutorial class as shown below.

```
namespace MatCSharp
{
    class Program
    {
        static void Main(string[] args)
        {

            float[] v1 = { 1, 2, 3 };
            float[] v2 = {4, 5, 6.8f};
            MArray[] outs; // Define array of type MArray to receive Matlab outputs
            // Make instance of the Matlab class.
            tutorialclass tc = new tutorial.tutorialclass();
```

Fig(3.17):snapshot 14

Make a call to the crossdot function .As shown below the function is called with three variables while it is a two variables function as in Matlab .

The extra variable is the variable number 1 .It is used to indicate how many output should be returned from the function .This is important as C# functions should return only one output while Matlab function can return as many variable as we need .To resolve this problem the Matlab class returns all the variables into single Array, then later on in C# you can parse the array for the specific output you need .

To make allocation for this Returned array, you should pass the number of elements expected to the Matlab function .In our case we are expecting the function to return 2 variables a and b as indicated above in the crossdot.m .Note that the array outs will receive the two variables as shown below.

```
// This function has 2 inputs and 2 outputs. It is called like this appeared below
outs = tc.crossdot(2, (MWNumericArray)v1, (MWNumericArray)v2);
// 2 is the number of outputs that will be received by the outs array
// Note that you should cast each array to MWNumericArray when is being passed to Matlab
```

Then we can retrieve each output individually as shown below. They are returned into MWNumericArray class as shown below.

```
MWNumericArray cp = (MWNumericArray) outs [0];
MWNumericArray dp = (MWNumericArray) outs [1];
```

You may need to cast the MWNumeric array into a standard float array to be used smoothly within the application. This may be done like that indicated below. Now the cross product CP_C float array may be used within the C# program.

```
float[] cp_c = (float []) cp.ToVector(MWArrayComponent.Real);
```

Fig(3.18):snapshot 15

Isolated Arabic word recognition system

This tutorial will illustrate how you can use SPLib to build a simple word recognition system .

Objectives:

1. Understating basic steps to develop speech recognition system .
2. Illustrate the use of Class “IsolatedWordRec” in SpLib.

Resources:

- SPLib .
- HTK tools .
- Microsoft Visual C# 2005 or later .
- Suitable Microphone

Procedure:

- Figure 1 explains the class diagram of IsolatedWordRec class .
- **It consists of the following interface functions**

```
public void AddString(string wavFile, string []WordList)
```

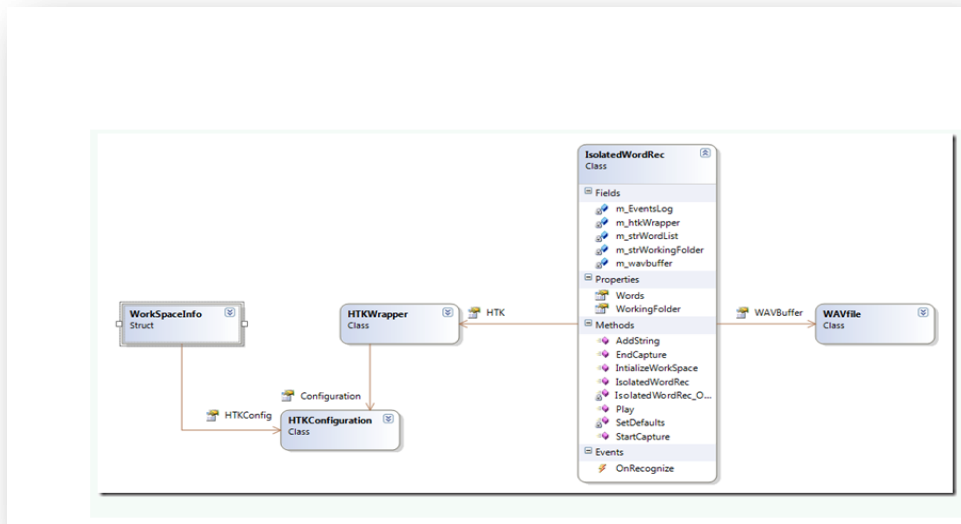
```
public void StartCapture ()
```

```
public void EndCapture ()
```

```
public void Play ()
```

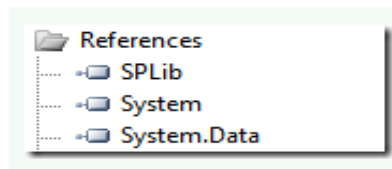
- The following properties

- HTK
- WAVBuffer
- Words
- WorkingFolder

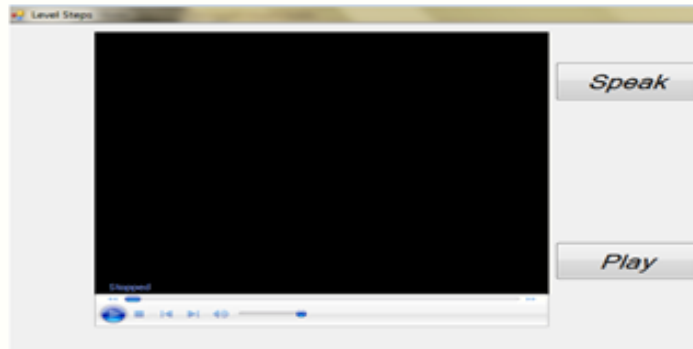


Fig(3.19) Library Static Class Diagram

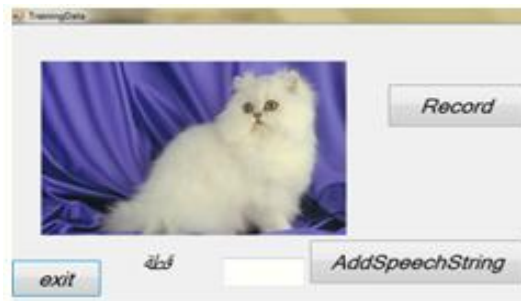
Add SPLib to the references of your project



Fig(3.20):snapshot 16



Fig(3.21):snapshot 17



Fig(3.22):snapshot 18

- Make the following Interface GUI .Name it Story
- Add instants of IsolatedWordRec Class to the form “Story”, "TrainingData", and "Traing Set".

```
private SPLib.IsolatedWordRecm_rec = new SPLib.IsolatedWordRec();
```

Add The following code into Story constructor

You should assign certain working folder for the recognizer to put the intermediate and temp files .In this folder it is expected to find the HTK tools.

For example assume that the working folder you choose is

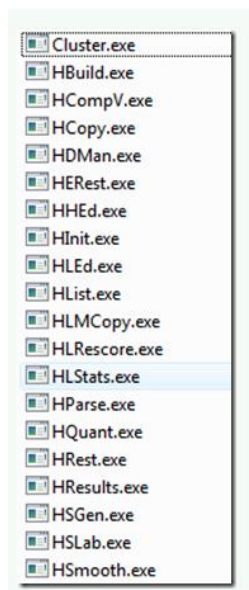
G:\Temp

It is expected that HTK tools is stored into a subfolder named HTK into

G:\temp

G:\Temp\HTK

Below is part of HTK tools .



Fig(3.23):snapshot 19

You will need to add the following code into Class Story constructor .

Note that InitializeComponent(); is automatically inserted by the IDE (Integrated Development Environment for C#)

```
public Audio()
{
    InitializeComponent();
    m_rec.WorkingFolder=@"G:\TEMP";
    m_rec.OnRecognize +=new
SPLib.OnRecognizeEventHandler(rec_OnRecognize);
}
```

- Speak Button is used to capture the Microphone input speech

Add the following code tomouseDown and Up events handlers of Speak button

```
private void SpeakButton_MouseDown(object sender, MouseEventArgs
e)
{
    m_rec.StartCapture();
}
```

```
private void SpeakButton_MouseUp(object sender, MouseEventArgs e)
{
    m_rec.EndCapture();
}
```

- Handle recognition results into the event handler

```
OnRecognizeEventHandler(rec_OnRecognize);
```

```
void rec_OnRecognize(object sender, string [] words)
```

```
{
    RecognitionResultsListBox.Items.Clear();
    RecognitionResultsListBox.Items.AddRange(words);
}
```

- Training the system .

Before starting use this system, you should initialize it with some words . Using SPLib this is a simple process . You just have to add speech strings to the system.

Add the following code to the Button record Mouse down and up events . This will be used to store the input speech string into a temp wav file into the working folder . This wav file plus the associated word list into the text box will consists speech string.

```
private void RecordSpeechStringButton_MouseDown(object sender,
MouseEventArgs e)
```

```
{
    m_rec.WAVBuffer.Open();
    m_rec.WAVBuffer.Record();
}
```

```
private void RecordSpeechStringButton_MouseUp(object sender,
MouseEventArgs e)
{
    m_rec.WAVBuffer.Store();
}
```

Add the following code into click event handler of AddSpeechString Button

```
private void AddSpeechStringButton_Click(object sender, EventArgs e)
{
    string []splitter ={" ",",",",",",":"};
    string []wordlist =textBox1 .Text .Split
(splitter,StringSplitOptions.RemoveEmptyEntries);
    m_rec.AddString(m_rec.WAVBuffer.FileName, wordlist);
}
```

- Testing the System

Now you can press Speak Button , then start speaking in isolated speech style, then release the button .The results will be listed into the list box.

At any time you can press the button play to listen to the last recorded speech

you will need to add the following code into Click events of the Play Button

```
private void PlayButton_Click(object sender, EventArgs e)
{
    m_rec.Play();
}
```

3.3 C# programming for Multimedia applications

This article will focus on playing a WAVE .wavsound file, not MPEG-3 .mp3sound file. You can play mp3 files using a Windows Media Player Windows Forms control; however, this only makes sense if you want to load the actual windows media player control in your windows form GUI could hurt performance.

3.3.1. How To Use Windows Media Player To Play Audio And Video

This how-to shows you how to use the Windows Media Player component in your C# project to play back various forms of audio and video .

1-Adding WMP to the Toolbox

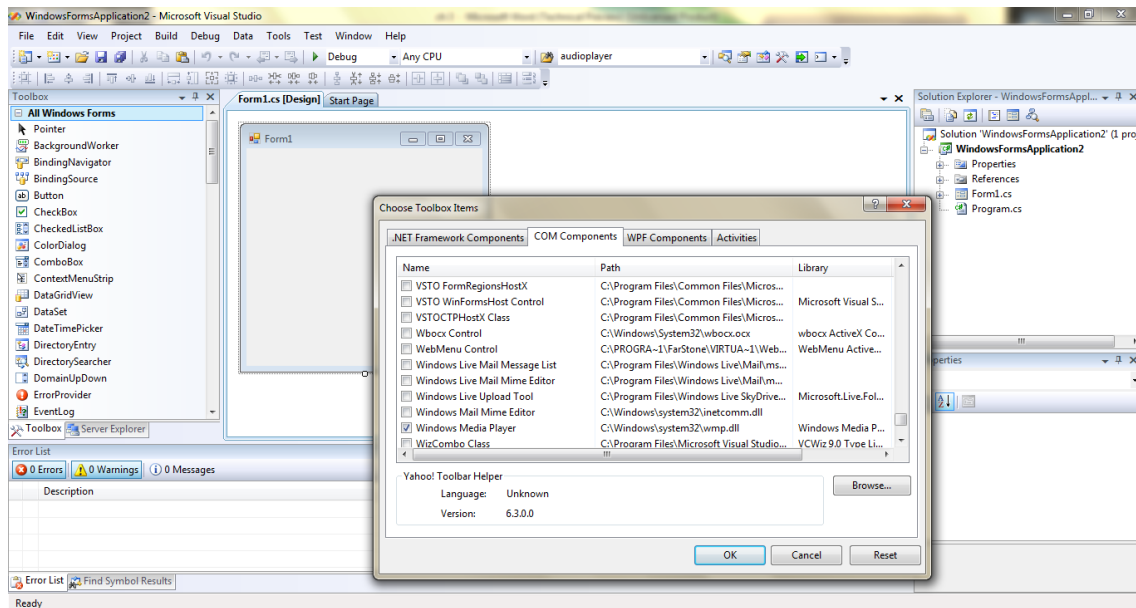
1-Make a new Windows Application (Windows Forms app)solution in Visual .

2- Double click on the form(Form1 .cs) and look in to your (view >Toolbox if its not already open .

3-Expand the Common tab in the toolbox.

4- If Windows Media Player(WMP) is not in the list , right click

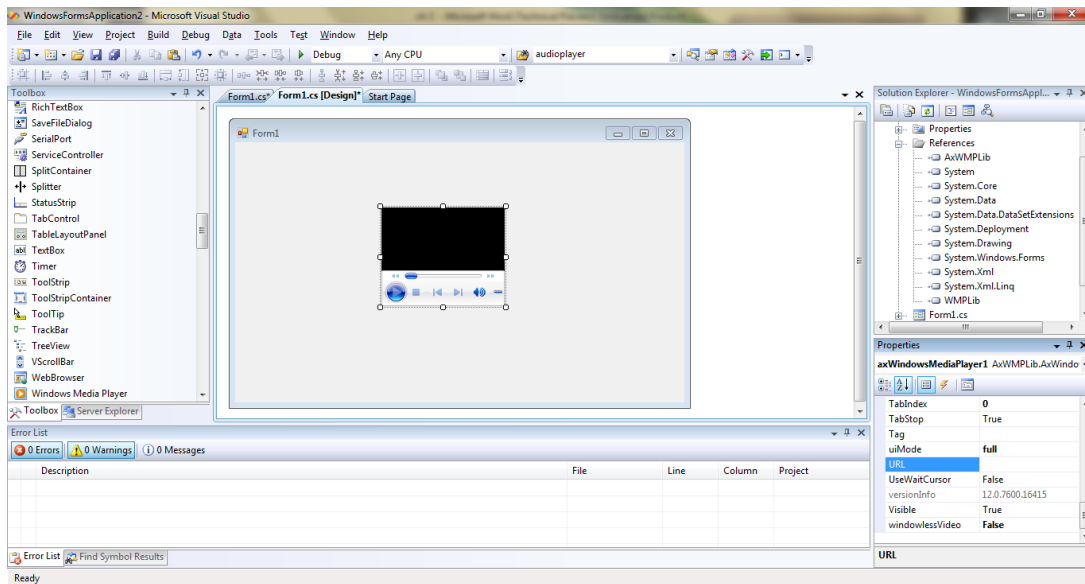
In the toolbox and click “Choose Item”...in the open dialog click the Com Components tab and select “Windows Media Player”.



Fig(3.24) snapshot20

5- Now, drag the WMP component onto your form. (You can toggle its visibility in case you don't want it to show)

6-In your code behind file for the form, you can play the sound by assigning a URL to the WMP control and calling play (you have to set the "URL" property of the media player control to the location of the file (media file)you can specify a local file)



Fig(3.25) snapshot21

7- Click on the Media player control in form Designer. In the properties windows, which over to the Events list. Double click on the play state changed event to create a new event handler

Place the following code in the new Event Handler:

(In our code we use this code)story form

```
private void videoPlayer_PlayStateChange(object sender,
AxWMPLib_WMPOCXEvents_PlayStateChangeEvent e)
```

```
{
```

```
    If(e.newState==1)
```

```
        playButton.Visible=true;
```

```
    if (e.newState==1)
```

```
        SpeakButton.Visible=true;}
}
```

To make play button and speak button both are visible after specified video is ended .Because there properties are invisible and we want them to be visible after showing video.

2-MediaPlayer methods and properties:

Media player provides methods and properties for playing songs in the media library

Name	Description
IsDisposed	.Gets a value that indicates whether the object is disposed
IsLooped	.Gets a value that indicates whether the player is playing video in a loop
IsMuted	.Gets or sets the muted setting for the video player
PlayPosition	.Gets the play position within the currently playing video
State	.Gets the media playback state, MediaState
Video	.Gets the Video that is currently playing
Volume	.Gets or sets the video player volume

Table 3.1

3.4 Analysis of Data

Analysis of data is a process of inspecting, cleaning, transforming, and modeling data with the goal of highlighting useful information, suggesting conclusions, and supporting decision making .Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, in different business, science, and social science domains.

The latest development is voice recognition analysis) VRA) (in our project.(This is a computer based system that records the user voice from a Suitable Microphone to a computer and uses the user's voice pattern to determine if they are telling the truth .Some will tell the user that the voice is being recorded, others may even say that the voice recorded is going through a VRA, but some will just say nothing .The VRA has no human involvement, the computer software decides if you are telling nothing .It works during a say some little minute record to report or discuss a claim.

For our project we use some buttons to do this tasks:

Interactive mode:

- Speak Button is used to capture the Microphone input speech
- Microphone to receive the speech stream

- In mouseDown event of speakbutton which is titled "Speak "the start capture is called to start receiving the speech stream by the Microphone connected to the computer .

```
private void SpeakButton_MouseUp(object sender, MouseEventArgs e)
{
    m_rec.EndCapture();
}
```

- In mouse up event of the button the endcapture function is called to end the recognition session .At this instant the SpeechLib will send the recognition results to the event's handler .
- The event handler that receives the recognition results .This event handler is in the test application not in the library .It is automatically receive the results when endcapture is called as mentioned shortly.

```
void rec_OnRecognize(object sender, string[] words)
{
    .
```

The recognition results are stored into the words Array .It sends all the results. We only compare this result with correct answer which is stored in our database.

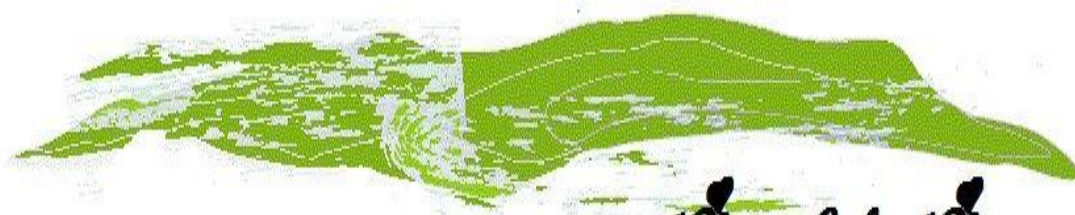
Asinour application we will use the string array directly to check if it contains the expected results or not.If it contains the expected results in this case the user can continue this level and the next levels.

CHAPTER 4

QUICK USER GUIDE



Smart Interactive E-Teacher



Pet Life Project



4.1. Use cases

4.1.1 Use cases scenario:-

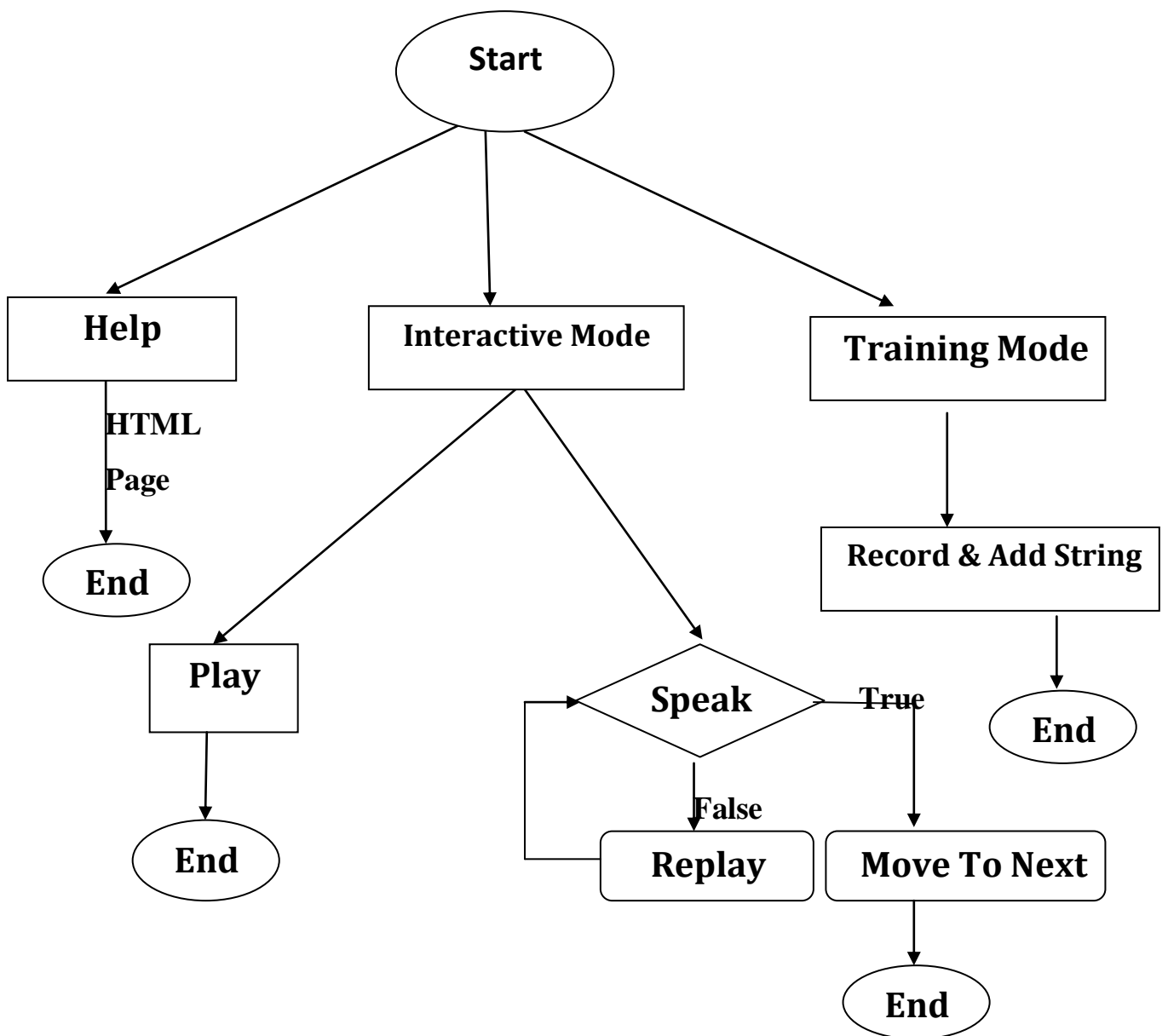
You will have three option after he enter the home page screen of our program

1-The help button will enable the user to read the total explaining for our program

2-The training button will enable you to train program of the data and that may be increase the quality of the program if he use it correctly you will say it after you press mouse down at record button and when you finish saying its name press the mouse up then you will type its name on the text box and then click at add string button

3-The interactive button will enable you to test the system recognition quality after you enter this mode you will hear the story as aparts, if your answer is right you will move to the next part Then you will hear a question,you have to answer it after you press mouse down at speak button and when you finish press the mouse up. If your answer is false you will hear this part again to help you correcting answer If you want to hear the last recorded speech just click on the play button

4.1.2 Use cases Diagram:-



Fig(4.1)

4.2 Basic Settings

4.2.1 Sound Cards:

Because speech requires a relatively low bandwidth, just about any medium-high quality 16 bit sound card will get the job done. You must have sound enabled in your kernel, and you must have correct drivers installed. Sound card quality often starts a heated discussion about their impact on accuracy and noise.

Sound cards with the 'cleanest' A/D (analog to digital) conversions are recommended, but most often the clarity of the digital sample is more dependent on the microphone quality and even more dependent on the environmental noise. Electrical "noise" from monitors, pci slots, hard-drives, etc. are usually nothing compared to audible noise from the computer fans, squeaking chairs, or heavy breathing.

Some ASR software packages may require a specific sound card. It's usually a good idea to stay away from specific hardware requirements, because it limits many of your possible future options and decisions. You'll have to weigh the benefits and costs if you are considering packages that require specific hardware to function properly.

4.2.2 Microphones:

A quality microphone is key when utilizing ASR. In most cases, a desktop microphone just won't do the job. They tend to pick up more ambient noise that gives ASR programs a hard time.

Hand held microphones are also not the best choice as they can be cumbersome to pick up all the time. While they do limit the amount of ambient noise

The best choice, and by far the most common is the headset style. It allows the ambient noise to be minimized, while allowing you to have the microphone at the tip of your tongue all the time. Headsets are available without earphones and with earphones (mono or stereo). I recommend the stereo headphones, but it's just a matter of personal taste.

A quick note about levels: Don't forget to turn up your microphone volume. This can be done with a program such as XMixer or OSS Mixer and care should be used to avoid feedback noise.

4.2.3 Computers/Processors:

ASR applications can be heavily dependent on processing speed. This is because a large amount of digital filtering and signal processing can take place in ASR.

As with just about any cpu intensive software, the faster the better. Also, the more memory the better. It's possible to do some SR with 100MHz and 16M RAM, but for fast processing (large dictionaries, complex recognition schemes, or high sample rates), you should shoot for a minimum of a 400MHz and 128M RAM. Because of the processing required, most software packages list their minimum requirements.

4.2.4 How to Use a Speech-Recognition Headset

Not all headsets are created equal, and speech-recognition software requires a high-quality headset with some special features in order to ensure accuracy. Anyone who uses speech-recognition software knows that accuracy is the most important thing or the software loses its convenience. It's therefore just as important to buy a headset designed specifically for speech recognition as it is to use it properly

Things You'll Need:

- Speech-recognition headset
- Computer
- Speech-recognition software

Choose the Right Speech-Recognition Headset

[Step 1](#)

Buy only speech-recognition headsets that are specifically designed for use with speech- recognition software. These headsets have been tested for accuracy.

[Step 2](#)

Look for speech-recognition headsets that use noise-canceling technology to filter out background noise. Particularly important when attempting speech recognition in a busy work environment, noise-canceling technology can significantly improve speech-recognition accuracy.

 [Step 3](#)

Choose a headset with a long, flexible microphone that can be positioned directly in front of your mouth. Rigid microphones may not fit close enough to pick up your words properly. A quality microphone is crucial for speech recognition.

 [Step 4](#)

Consider buying a wireless Bluetooth speech-recognition headset if you need increased mobility. If you find yourself constantly putting on and taking off your headset to move about your office, this is a good option.

 [Step 5](#)

Shop for speech-recognition headsets that have volume controls and a mute button for added convenience.

Use Your Speech-Recognition Headset Effectively

 [Step 1](#)

Plug your headset directly into your computer or into the soundcard using the 2.5mm connector, the 3.5mm connector or the USB cable.

 [Step 2](#)

Adjust the headband on your headset so that it fits comfortably and stably.

 [Step 3](#)

Position the microphone so it is directly in front of your lips. This will help the microphone pick up your words as accurately as possible.

 **Step 4**

Limit the background noise as much as possible when using your speech recognition software. This includes turning off any televisions or radios, closing the windows and doors, turning off the air conditioning if possible and moving to a quiet location. Noise cancellation will help with background noise, but it will not eliminate the problem.

 **Step 5**

Speak as clearly as possible and enunciate your words without sounding stilted. You want to talk naturally, but even the best headset can't help you if you mumble.

4.2.5 How to Configure a Computer for a Bluetooth Headset

When you configure your Bluetooth headset to work with your computer, you can control all of the incoming and outgoing sounds wirelessly. This is useful when using voice-over-Internet protocols, video and audio conferencing, playing interactive video games, using speech-recognition software, recording audio dictation and many other applications.

Things You'll Need:

- Bluetooth headset
- Computer with Bluetooth capability

Setup a Bluetooth Headset for Use with Windows XP



Step 1

Go into the Bluetooth Devices item in the Control Panel on your computer to access configuration options in order to pair your Bluetooth headset with your computer.



Step 2

Click on the Options tab in the Bluetooth Devices menu and select the "Turn discovery on" option. This will allow your computer to discover Bluetooth devices within range. This option is turned off by default for security and will automatically turn off when the Bluetooth connection you establish disconnects.



Step 3

Check the box for "Allow Bluetooth devices to connect to this computer." If this box is left unchecked, your headset will not have permission to connect.



Step 4

Turn on Bluetooth headset so your computer can discover it.



Step 5

Open Bluetooth settings and click Add. This will open the Add Bluetooth Device Wizard.



Step 6

Search for your Bluetooth headset by selecting "My device is set up and ready to be found" in the wizard and clicking on the next button. An icon for your headset will appear in the window when it has been found.



Step 7

Click on the icon for your device and the Next button. Enter the passcode for your headset. Your headset should then appear in the Bluetooth Devices list.



Step 8

Configure your computer to use your Bluetooth headset for audio functions by opening My Bluetooth Places.



Step 9

Right-click on the icon for your headset and select the Connect Headset option from the menu.



Step 10

Accept the connection on your headset according to manufacturer's instruction when you hear a beep. Often you are not required to take action to accept the connection. The icon should light up when the connection is complete.

4.2.6 How to Set Up Speech Recognition Software:

There are a number of speech recognition software packages available for both Windows and Macintosh computers. Though each speech recognition software package is different and offers different features, they all work similarly.

For the most part, they can be set up according to the same guidelines and using the same types of equipment. In fact, there are some universal (and basic) steps to take when setting up voice recognition software.



Step 1

Get a quality microphone. A microphone of good audio quality gives the speech recognition software every opportunity to identify what you're saying.



Step 2

Set up and configure the microphone. Making the volume too quiet can mean your microphone will be missing some parts of your speech. Making it too loud can mean the sound will become fuzzy and distorted, making speech impossible to comprehend.



Step 3

Select the correct language. depending on the speech recognition software. There will often be a number of language selections from which to choose.



Step 4

Train the software often. The more you train your speech recognition software, the more effective and accurate it will be. Without training, the software will make more mistakes and become less effective.



Step 5

Explore the feature set of your voice recognition software. Use every feature of your software at least once to see what's there. Though you probably won't use many of the available features, you may find that some of them are extremely useful to you.

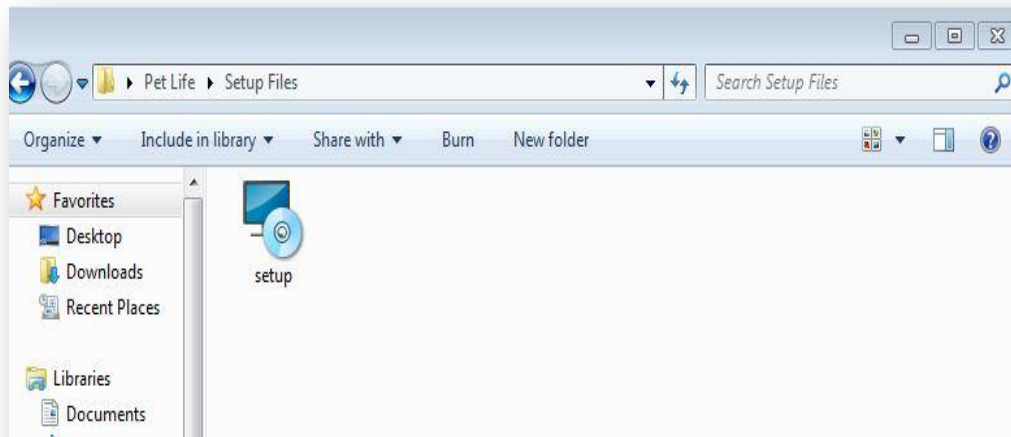


Step 6

Speak clearly. If you're used to speaking quickly, you may have to slow down at first. As the speech recognition software is trained, you can begin to speak faster.

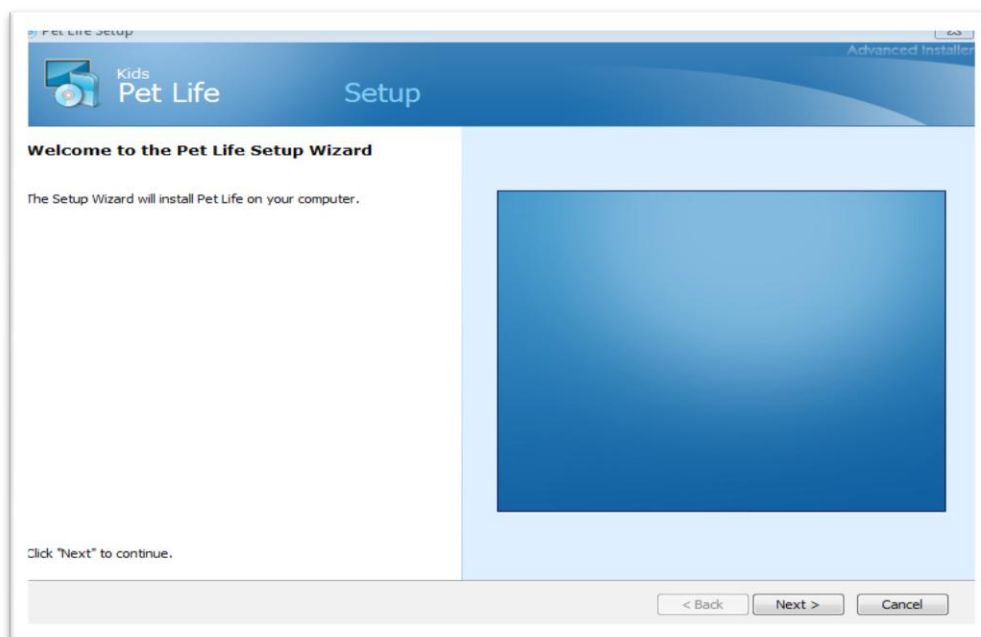
4.1.7 How to Set Up Smart Interactive E-Teacher Software:

open the setup files and click on the setup icon



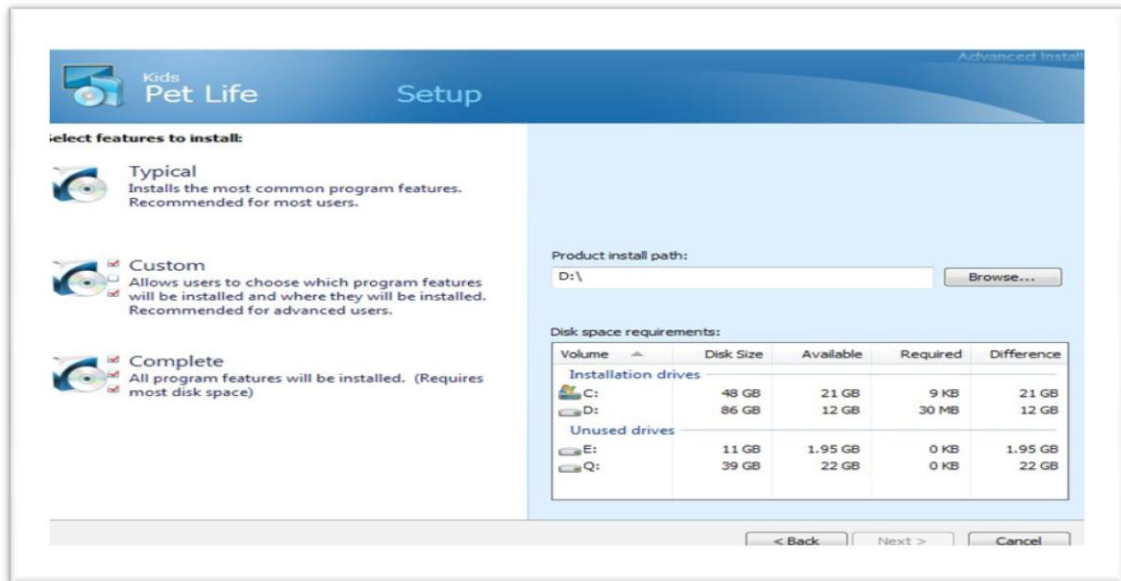
Fig(4.2) snapshot1

After double clicking the setup EXE file you will be prompted with a welcome screen **Click Next** to begin an Express Install:



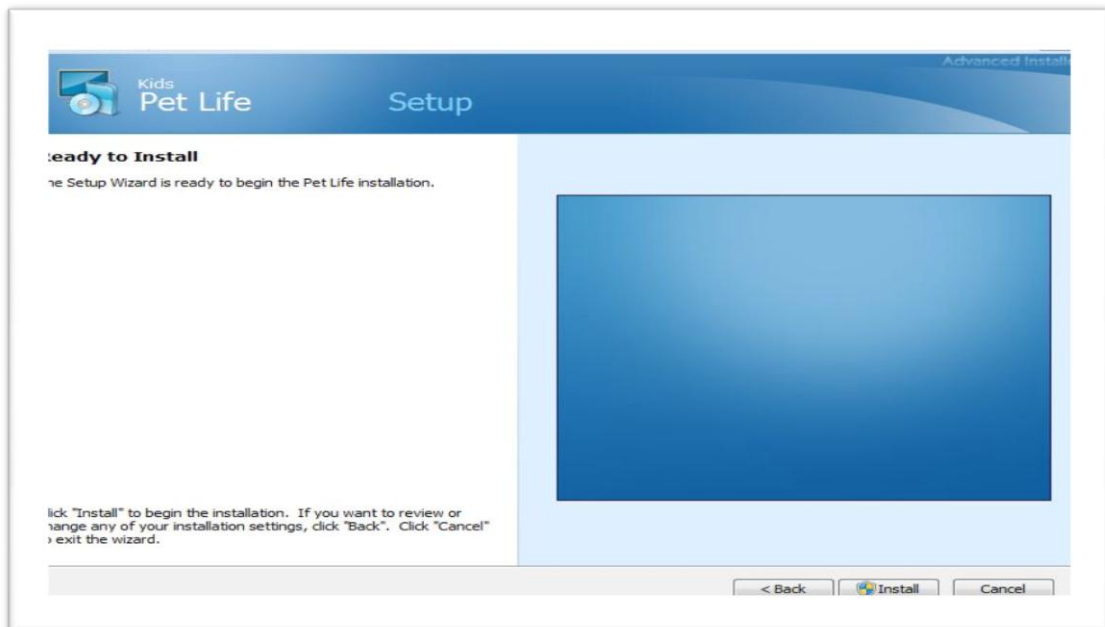
Fig(4.3) snapshot2

If you wish to do a Custom Install check Custom install but this is not needed for most users



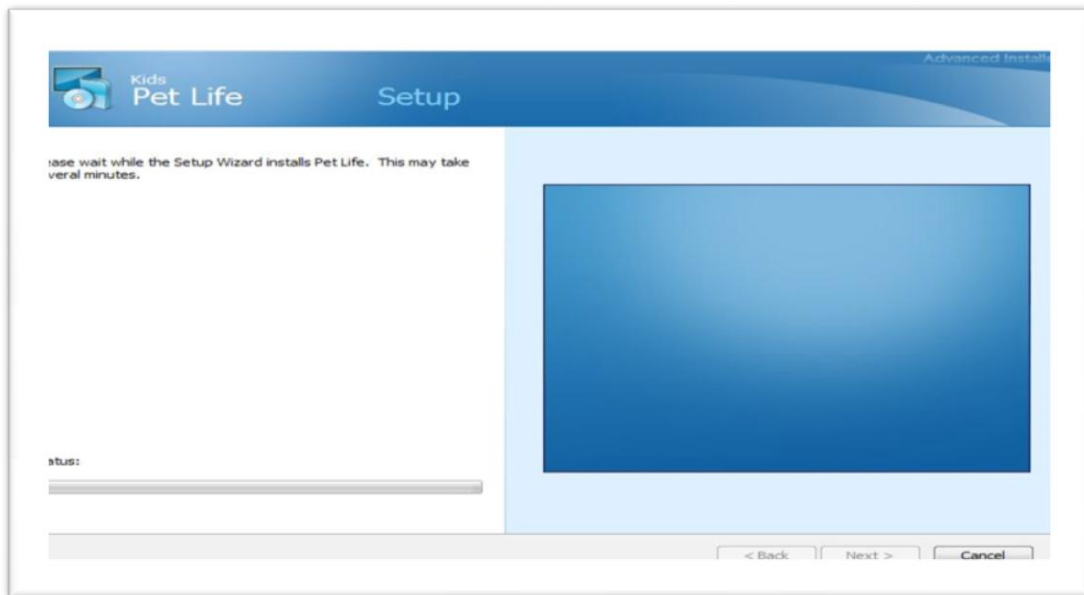
Fig(4.4) snapshot3

After pressing next, you will have the option to choose the path where to install to. If at all possible, please keep the default location and press next



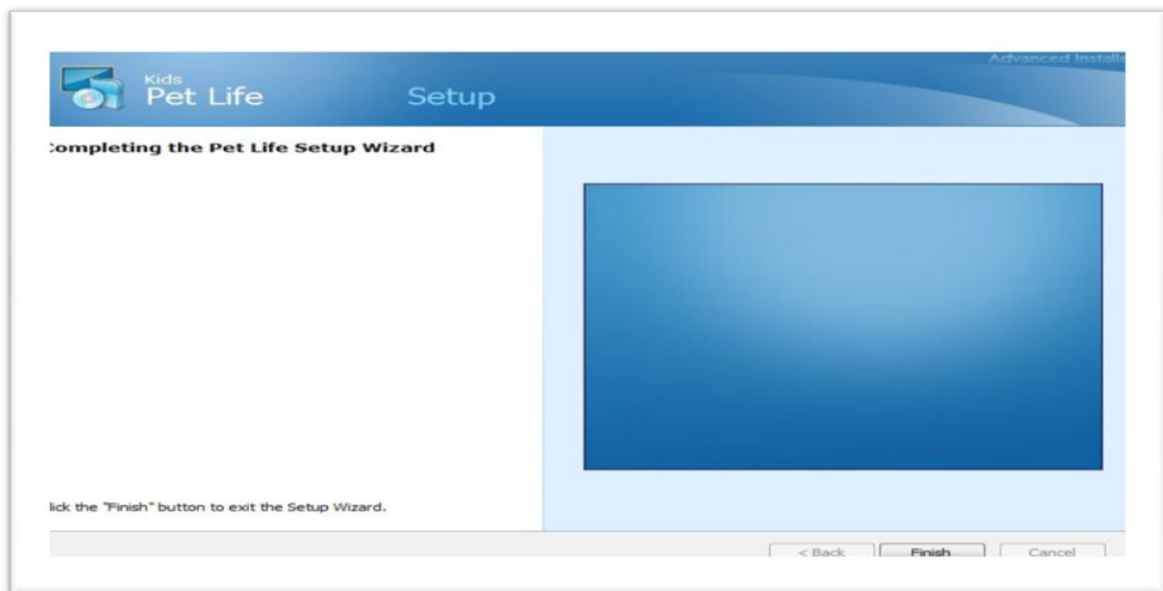
Fig(4.5) snapshot4

Then click install button your setup will start



Fig(4.6) snapshot5

Then click finish button



Fig(4.7) snapshot6

After that you can access our program from desktop shortcut

4.3 Training Mode:

Proper training is critical. A solid training foundation is the key to on-going success with speech recognition for all users regardless of skill or age. There are in fact two aspects of training with speech recognition.

First, the speech recognition system itself must be properly trained to recognize the student's words.

The software gets accustomed to the user's voice by building an individual model that is modified with every utterance. This model helps the software predict what word to display from the active dictionary with every subsequent user utterance. The better the model, the better the prediction, so that if the software is used correctly, prediction improves with increased usage.

Therefore, the trainer should help the student gain a general understanding of how the speech recognition software works, so that he or she understands the importance of proper usage.

This brings us to the second aspect of training--the student must be trained in all aspects of the system that they need to know. All users, and especially younger users, must be properly trained in the process of saying and selecting the words. Additionally, users must learn how to correct any mismatches between the user's spoken word and the software's predictions. Beyond this, some students may also want or need to learn how to spell by voice.

For training acoustic models is necessary a set of feature files computed from the audio training data, one each for every recording in the training corpus. Each recording is transformed into a sequence of feature vectors consisting of the Mel-Frequency Cepstral Coefficients (MFCCs). The training was performed using utterances of speech data collected from speakers

Table 1 Phonemes symbols used in the training of HMM:

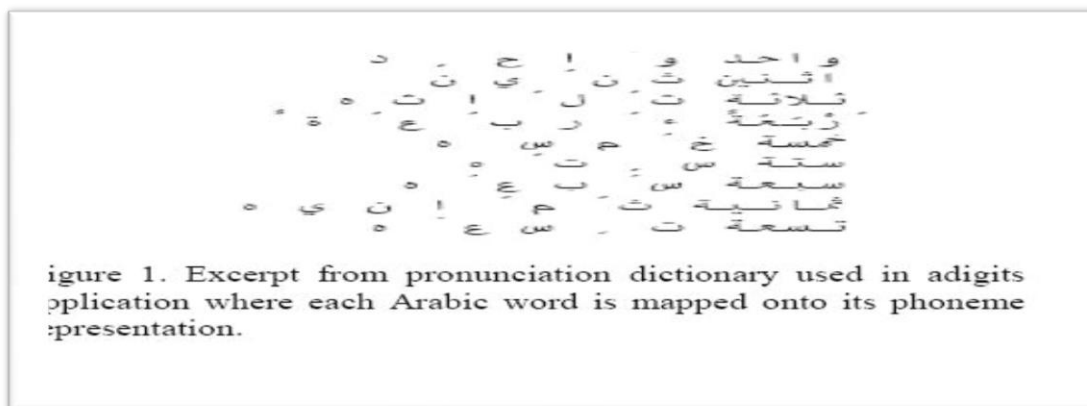
Transliteration	Alphabet
Alef	ء
Ba'	ب
Ta'	ت
Tha'	ث
Ha'	ح
Emphatic Kha'	خ
Dal	د
Ra'	ر
Ayn	ع
Sin	س
Emphatic Sad	ص
Lam	ل
Mim	م
Ha'	ه
Waw	و

Table 4.1

The training process consists of: convert the audio data to a stream of feature vectors, convert the text into a sequence of linear triphone HMMs as shown in

Table 1 using the pronunciation dictionary, and find the best state sequence or state alignment through the sentence HMM for the corresponding feature vector sequence.

For each senone, gather all the frames in the training corpus that mapped to that senone in the above step and build a suitable statistical model for the corresponding collection of feature vectors. The circularity in this training process is resolved using the iterative Baum-Welch or forward-backward training algorithm .



Fig(4.8)

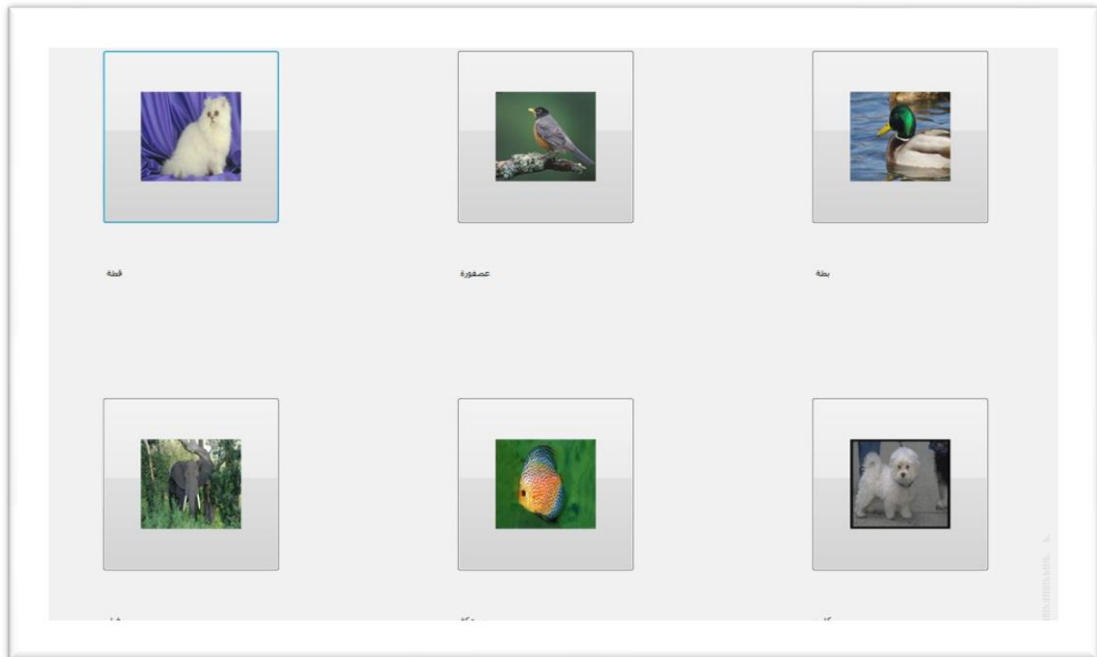
Two training modes are defined as

- Training on **clean** data only and as
- Training on clean and noisy (**multi-condition**) data.

The advantage of training on clean data only is the modeling of speech without distortion by any type of noise. Such models should be suited best to represent all available speech information. The highest performance can be obtained with this type of training in case of testing on clean data only. But these models contain no information about possible distortions.

This aspect can be considered as advantage of multi-condition training where distorted speech signals are taken as training data. This leads usually to the highest recognition performance when training and testing are done in the same noise condition.

This is our Training Mode form



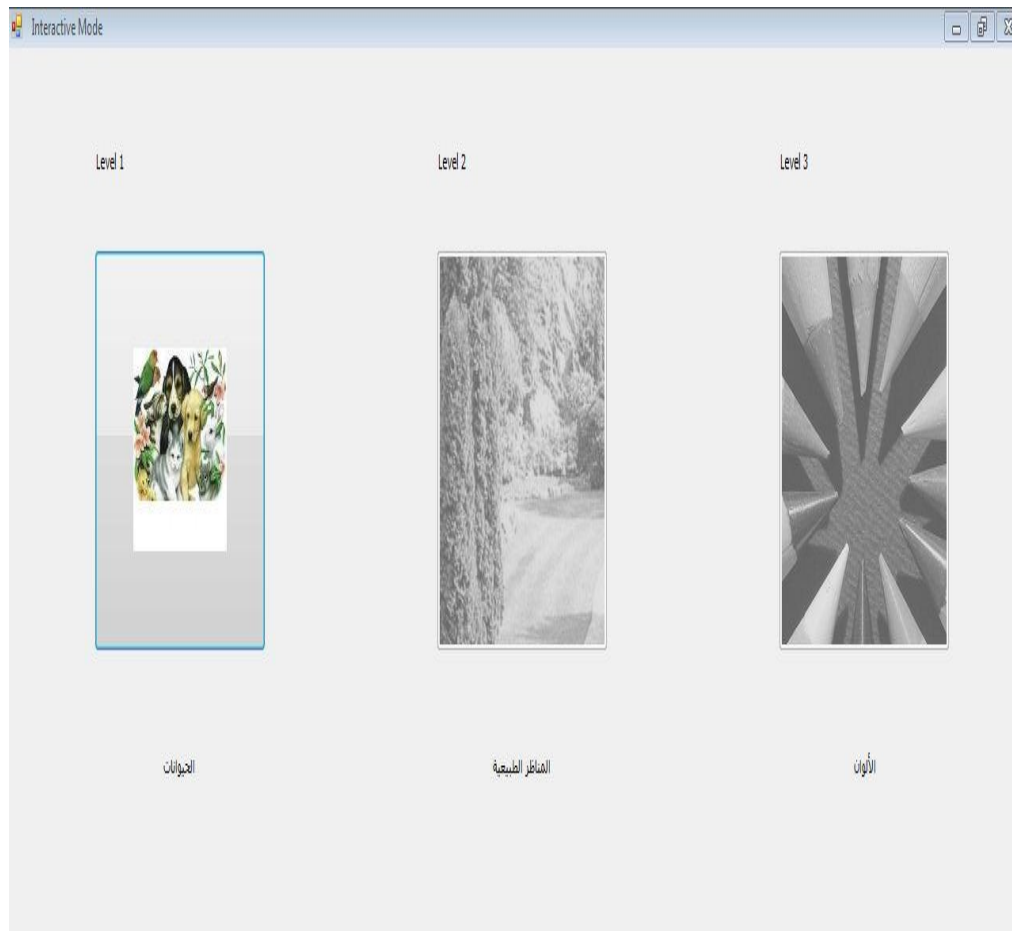
Fig(4.9) snapshot7

By using this mode, we can train our program to have better recognition even the user can train the program but he must be careful as he may make the system more better or damage it.

4.4 Interactive Mode

In this mode we can do our recognition

This is our interactive mode form



Fig(4.10) snapshot8

Here, we are activating only level one and after the child answer all its question right he will be allowed to start the next level

Child must speak the answer clearly and he must say just one word.

4.5. Step By Step Tutorial

In this stage we will explain our program in details

After you finishing the setup of the program this screen shot will appear



Fig(4.11) snapshot9

This is our homepage as we see you will have three choice

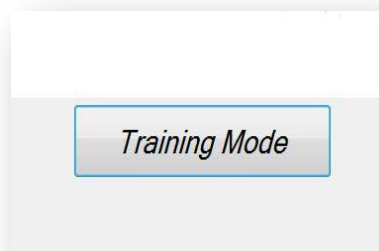
1-First: The Help Button



Fig(4.12) snapshot10

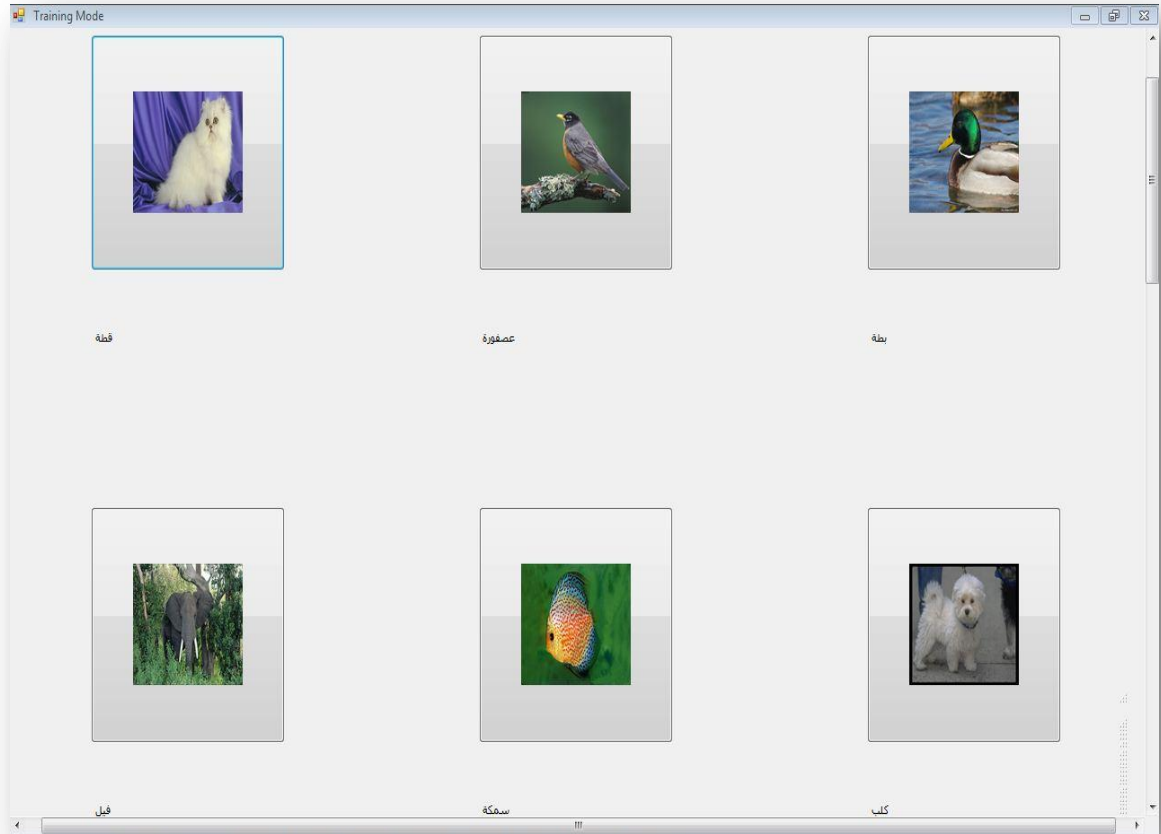
When you click at this button, the html help will be opened to explain our program

2-Second: Training Mode Button

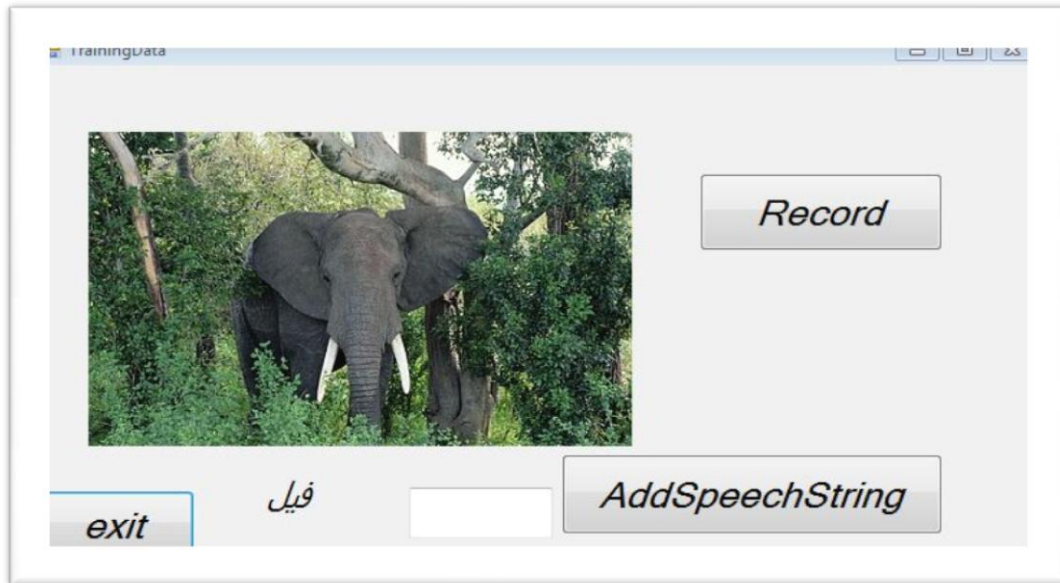


Fig(4.13) snapshot11

When you click at this button you will log in to training mode which enables you to train your system to have a better recognition but you must be careful to train your system in the right way otherwise your recognition quality will be decreased and the program will be damaged



Fig(4.14) snapshot12



When you click at any Image you log in to the trainind data

Fig(4.15) snapshot13

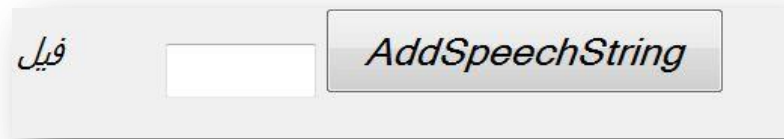
To train the system of the data

#If you know what is in the picture you will say it after you press mouse down at record button and when you finish saying its name press the mouse up



Fig(4.16) snapshot14

Then you will type its name on the text box and then click at add string button



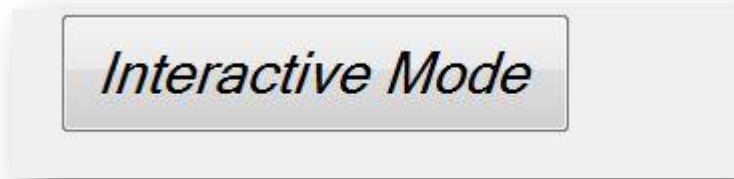
Fig(4.17) snapshot15

#If you donot know what is in the picture it is preferable not guess, as if you say wrong answer this will make the quality of the program decrease so it will be better to press exit button.



Fig(4.18) snapshot16

3-Interactive Mode Button :



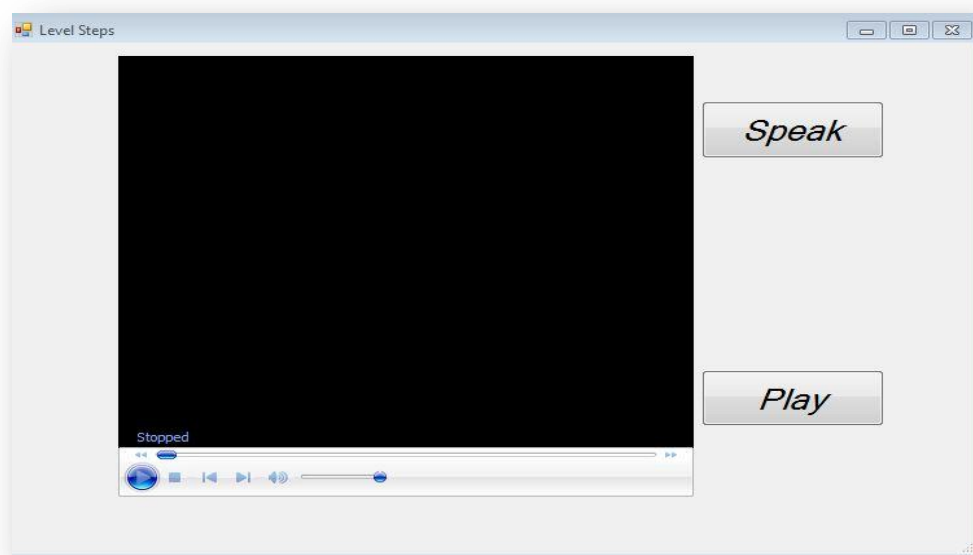
Fig(4.19) snapshot17

When you click at this button you will log in to interactive mode which enables you to make recognition and test your system quality



Fig(4.20) snapshot18

You must click on the button of level 1 that will be activated then you will log in to the level steps screen



Fig(4.21) snapshot19

At this screen you will hear the story as aparts, if your answer is right you will move to the next part Then you will hear a question,you have to answer it after you press mouse down at speak button and when you finish press the mouse up.

If your answer is false you will hear this part again to help you correcting answer



Fig(4.22) snapshot20



Fig(4.23) snapshot21

If you want to hear the last recorded speech just click on the play button

This is the end of the quick user guide

GLOSSARY

Phoneme Is the smallest segmental unit of sound employed to form meaningful contrasts between utterances .Thus a phoneme is a group of slightly different sounds which are all perceived to have the same function by speakers of the language or dialect in question.

Acoustic Model An acoustic model is created by taking audio recordings of speech, and their text transcriptions, and using software to create statistical representations of the sounds that make up each word . It is used by a speech recognition engine to recognize speech .

Language model A statistical **language model** assigns a probability to a sequence of m words by means of a probability distribution .

Consonant In articulatory phonetics, a **consonant** is a speech sound that is articulated with complete or partial closure of the vocal tract .

Vowel In phonetics, a **vowel** is a sound in spoken language , pronounced with an open vocal tract so that there is no build - up of air pressure at any point above the glottis

Allophone In phonetics, an **allophone** is a conditioned realization (phones) of the same phoneme

Phone (phonetics) Within phonetics, a **phone** is the basic unit revealed via phonetic speech analysis

Speech recognition (also known as **automatic speech recognition** or **computer speech recognition**) converts spoken words to text

HTML: Hyper TextMarkup Language. It is the predominant markup language for web pages. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It allows images and objects to be embedded and can be used to create interactive forms.

-A Hidden Markov Model(HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved state

-HTK is a toolkit for building Hidden Markov Models (HMMs)

BIBLIOGRAPHY

References

- Ajami Y., “Investigating Spoken Arabic Digits in Speech Recognition Setting,” in Proceedings of Information's and Computer Science, UK, pp. 173-174, 2005.
- Al-Zabibi M., “An Acoustic Phonetic Approach in Automatic Arabic Speech Recognition,” Document with UMI, the British Library, UK, 1990.
- CMU <http://cmusphinx.sourceforge.net/html/cmusphinx.php>, 2003.
- Deller J., Proakis J., and Hansen J., Discrete Time Processing of Speech Signal, Macmillan, NY, 1993.
- Deshmukh N., Ganapathiraju A., Hamaker J., Picone J., and Ordowski M., “A Public Domain Speech to Text System,” in Proceedings of 6th European Conferences on Speech Communication and Technology, Hungary, pp. 2127-2130, 1999.
- El-Imam A., “An Unrestricted Vocabulary Arabic Speech Synthesis System”, Computer Journal of IEEE Transactions on Acoustic Speech and Signal Processing, vol. 37, no. 12, pp. 1829-1845, 1989.

- Elshafei M., "Toward an Arabic Text to Speech System," *Computer Journal of the Arabian Science and Engineering*, vol. 4, no. 16, pp. 565-583, 1991.
- Haton M., Cerisara C., Fohr D., Laprie Y., and Smaili K., *Reconnaissance Automatique de la Parole du Signal a Son Interpretation*, Monographies and Books, Oxford, 2006.
- Hiyassat H., Nedhal Y., and Asem S., "Automatic Speech Recognition System Requirement Using Z Notation," in *Proceedings of AMSE' 05*, France, pp. 514-523, 2005.
- Huang D., *Automatic Speech Recognition: The Development of the SPHINX System*, Kluwer Academic Publishers, 1989.
- Huang X., Acero A., and Hon H., *Spoken Language Processing: A Guide to Theory, Algorithm and System Design*, Prentice Hall, 2001.
- Huang X., Alleva F., Hon W., Hwang M., and Rosenfeld R., "The SPHINX-II Speech Recognition System: An Overview," *Computer Journal of Computer Speech and Language*, vol. 7, no. 2, pp. 137-148, 1993.
- Huang X., Ariki Y., and Jack M., "Hidden Markov Models for Speech Recognition," *Technical Report*, Edinburgh, UK, 1990.
- Kirchho K., Bilmes J., Henderson J., Schwartz

- R., Noamany M., Schone P., Ji G., Das S., Egan M., He F., Vergyri D., Liu D., and Duta N., “Novel Approaches to Arabic Speech Recognition,” Technical Report, Ohns-Hopkins University, 2002.
- Lee K., Hon H., and Reddy R., “An Overview of the SPHINX Speech Recognition System,” Computer Journal of IEEE Transactions on Acoustics Speech and Signal Processing, vol. 38, no. 1, pp. 35-45, 1990.
- Li X., Zhao Y., Pi X., Liang H., and Nefian V., “Audio Visual Continuous Speech Recognition Using a Coupled Hidden Markov Model,” in Proceedings of 7th International Conferences on Spoken Language Processing, Denver, pp. 213-216, 2002.
- Muhammad A., “AlaswaatAlaghawaiyah,” in Proceedings of International Conference on Signal Processing, Jordan, pp. 646-651, 1990.
- Pullum G. and Ladusaw W., Phonetic Symbol Guide, Near New, USA, 1996.
- [19] Ravishankar K., “Efficient Algorithms for Speech Recognition,” PhD Thesis, 1996.
- Satori H. and Chenfour N., “Arabic Speech Recognition System based on CMUSphinx,” in Proceedings of International Symposium on Computational Intelligence, Morocco, pp. 31-35, 2007.

- Satori H., Harti M., and Chenfour N.,
 “Introduction to Arabic Speech Recognition
 Using Cmusphinx System,” in Proceedings of
 Information and Communication Technologies
 Interantinal Symposium (ICTIS'07), Morocco,
 pp. 139-115, 2007.
- Vergyri D. and Kirchhoff K., Automatic
 Diacritization of Arabic for Acoustic Modelling
 in Speech Recognition, Editors, Coling, Geneva,
 2004.
- Vergyri D., Kirchhoff K., Duh K., and Stolcke
 A., “Morphology Based Language Modeling for
 Arabic Speech Recognition,” in Proceedings of
 Interspeech, Germany, pp. 2245-2248, 2004.
- Young S., “The HTK Hidden Markov Model
 Toolkit: Design and Philosophy,” Technical
 Report TR 152, 1994.

190e International Arab Jour

- L. Rabiner& B. Juang. 1993. "Fundamentals of Speech Recognition
- B. Balentine, D. Morgan, and W. Meisel. 1999. How to Build a Speech
 Recognition Application".
- C. Becchetti and L.P. Ricotti. 1999. "Speech Recognition : Theory and
 C++ Implementation
- A. Syrdal, R. Bennett, S. Greenspan. 1994. Applied Speech Technology
- P. Foster, T. Schalk. 1993. "Speech Recognition : The Complete
 Practical Reference Guide

- D. Jurafsky, J. Martin. 2000. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition
- J. Deller, J. Hansen, J. Proakis. Discrete-Time Processing of Speech Signals (IEEE Press Classic Reissue)
- " F. Jelinek. 1999. Statistical Methods for Speech Recognition (Language, Speech, and Communication)".
- "" L. Rabiner, R. Schafer. 1978. Digital Processing of Speech Signals
- . C. Manning, H. Schutze. 1999. "Foundations of Statistical Natural Language Processing

APPENDIX A

HTK TOOLS

A.1 The Fundamentals of HTK

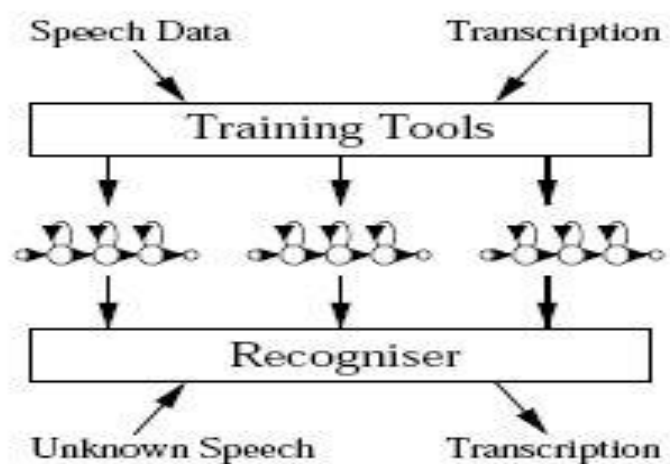


Fig. A.1 Message
Encoding/Decoding

HTK is a toolkit for building Hidden Markov Models (HMMs). HMMs can be used to model any time series and the core of HTK is similarly general-purpose. However, HTK is primarily designed for building HMM-based speech processing tools, in particular recognisers.

Thus, much of the infrastructure support in HTK is dedicated to this task. As shown in the picture above, there are two major processing stages involved. Firstly, the HTK training tools are used to estimate the parameters of a set of HMMs using training utterances and their associated transcriptions. Secondly, unknown utterances are transcribed using the HTK recognition tools.

It is necessary to understand some of the basic principles of HMMs. It is also helpful to have an overview of the toolkit and to have some appreciation of how training and recognition in HTK is organised.

A.2 General Principles of HMMs

A Hidden Markov Model (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved state. An HMM can be considered as the simplest dynamic Bayesian network. HMMs are used in speech recognition because a speech signal could be viewed as a piecewise stationary signal or a short time stationary signal. HMMs are popular because they can be trained automatically and are simple and computationally feasible to use.

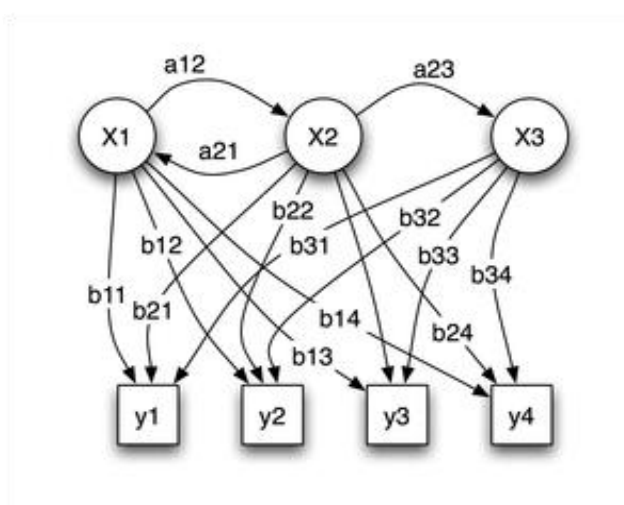


Fig A.2: HMM Model

A.3 Isolated Word Recognition

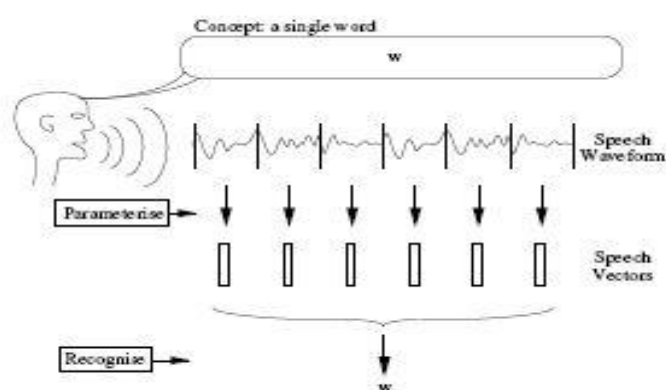


Fig. A.3 Isolated Word Problem

In HMM based speech recognition, it is assumed that the sequence of observed speech vectors corresponding to each word is generated by a Markov model as shown in Fig. A.4.

A Markov model is a finite state machine which changes state once every time unit and each time t that a state j is entered, a speech vector o_t is generated from the probability density $b_j(o_t)$. Furthermore, the transition from state i to state j is also probabilistic and is governed by the discrete probability a_{ij} . Fig. A.4 shows an example of this process where the six state model moves through the state sequence $X = 1; 2; 2; 3; 4; 4; 5; 6$ in order to generate the sequence o_1 to o_6 . Notice that in HTK, the entry and exit states of a HMM are non-emitting. This is to facilitate the construction of composite models

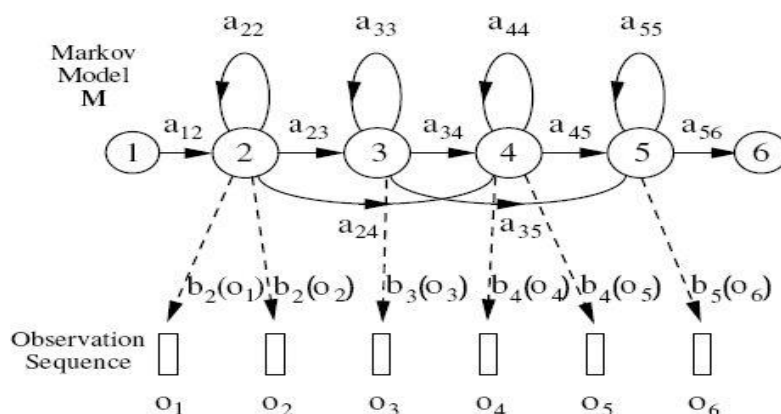
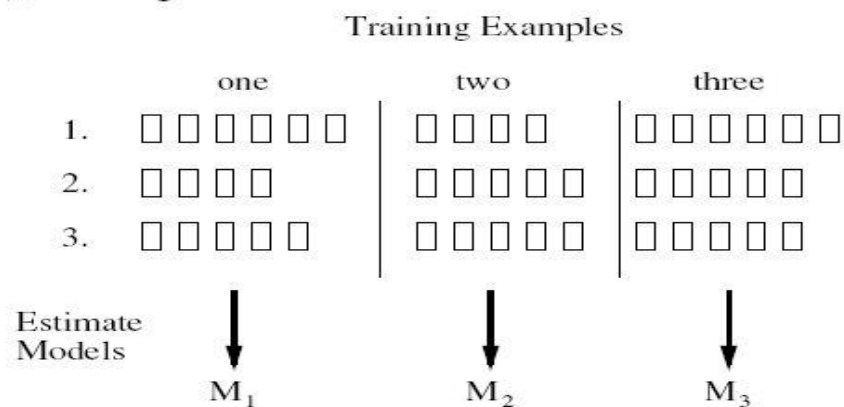


Fig. A.4 The Markov Generation Mode

All this, of course, assumes that the parameters f_{ij} and $b_j(o_j)$ are known for each model M_i . Herein lies the elegance and power of the HMM framework. Given a set of training examples corresponding to a particular model, the parameters of that model can be determined automatically by a robust and efficient re-estimation procedure. Thus, provided that a sufficient number of representative examples of each word can be collected then a HMM can be constructed which implicitly models all of the many sources of variability inherent in real speech.

Fig. A.5 summarises the use of HMMs for isolated word recognition. Firstly, a HMM is trained for each vocabulary word using a number of examples of that word. In this case, the vocabulary consists of just three words: "one", "two" and "three". Secondly, to recognise some unknown word, the likelihood of each model generating that word is calculated and the most likely model identifies the word.

(a) Training



(b) Recognition

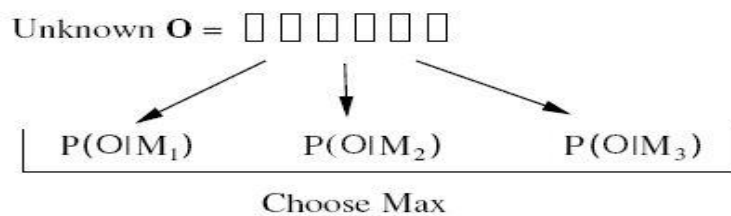


Fig. A.5 Using HMMs for Isolated Word Recognition

APPENDIX B

ARABIC PHONEMES INTERNATIONAL PHONETIC ALPHABETS (IPA)

IPA for Arabic

The charts below show the way in which the International Phonetic Alphabet (IPA) represents the Modern Standard form of the Arabic language in Wikipedia articles.

Notice that the pronunciations in the tables may differ, affected with the native variety of Arabic of the speaker.

1. IPA: Arabic <u>Consonants</u>			
IPA	Letter(s)	nearest English equivalent	<u>Trans.</u>
b	ب(<u>Bā'</u>)	But	B
t	ت(<u>Tā'</u>)	Sting	T
t^{s[1]}	ط(<u>Tā'</u>)	tall	t, T
d	د(<u>Dāl</u>)	deed	D
d^{s[1]}	ض(<u>Dād</u>)	dawn	ḍ, D
dʒ, g^[2]	ج(<u>Gīm</u>)	joy	ǧ, j
k	ك(<u>Kāf</u>)	skin	K

f	ف(Fā')	fool	F
q ^[1]	ق(Qāf)	No equivalent	q, k
θ	ث(Tā')	thing	t, th
ð	ذ(Dāl)	this	d, dh
ð ^[1] , z ^[1]	ظ(Zā')	No equivalent	z, Z
s	س(Sīn)	see	S
s ^[1]	ص(Sād)	No equivalent	ʃ, S
z	ز(Zayn)	zoo	Z
ʃ	ش(Shīn)	she	ʃ, sh
h	ه(Hā')	hen	H
m	م(Mīm)	man	M
n	ن(Nūn)	no	N
l	ل(Lām)	leaf	L
l ^[1]	الله	allah	L
r	ر(Rā')	trilledrun, like in Italian	R
w	و(Wāw)	we	W
j	ي(Yā')	yes	Y
x	خ(Hā')	loch	ħ, kh
ɣ	غ(Ġain)	between a light go and ahold	ġ, gh
ħ ^[1]	ح(Hā')	No equivalent	ħ, H, 7
ʕ ^[1]	ع('ayn)	No equivalent	ʕ, ' , 3
ʔ	ء (Hamza)	uh-(ʔ)oh	ʔ, ' ,

2. IPA: Arabic Vowels

IPA	Letter(s)	English Examples	<u>Trans.</u>
i:	ي	see	ī
i		sit	i

æ:, a: ^{[1][3]} (ʿAlif)	fan, fawn	ā
æ, a ^{[1][3]}	fat, fought	a
u: و	soon	ū
u	soot	u

3. IPA: Marginal Sounds

IPA	Letter(s)	English Examples	<u>Trans.</u>
p	پ (Pe)	spin	p
v	ف (Ve)	vine	v
g	ج ^[4] , گ (Gāf)	gut	g
ʒ	چ ^[5] , ژ (Zhe)	beige	ž, zh
tʃ	چ (Che) or تش church	church	č, ch
e:	ي		e
o:	و		o