CHAPTER I

INTRODUCTION

1.1 Introduction to OCR Based Speech Synthesis System using LabVIEW

Machine replication of human functions, like reading, is an ancient dream. However, over the last five decades, machine reading has grown from a dream to reality. Character recognition or optical character recognition (OCR), is the process of converting scanned images of machine printed or handwritten text (numerals, letters, and symbols), into a computer format text (such as ASCII). Optical character recognition has become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence. Many commercial systems for performing OCR exist for a variety of applications. Speech is probably the most efficient medium for communication between humans. A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read any text aloud, whether it was directly introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system.

1.1.1 OCR

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. Optical character recognition belongs to the family of techniques performing automatic identification. These different techniques are discussed below and define OCR's position among them.

1.1.2 Automatic Identification

The traditional way of entering data into a computer is through the keyboard. However this is neither always the best nor the most efficient solution. In many cases automatic identification may be an alternative. Various technologies for automatic identification exist, and they cover needs for different areas of application. Below a brief overview of the different technologies and their applications is given.

1.1.2.1 Speech recognition

In systems for speech recognition, spoken inputs from a predefined library of words are recognized. Such systems should be speaker-independent and may be used for instance for reservations or ordering of goods by telephone. Another kind of such systems are those used to recognize the speaker, rather than the words, for identification.

1.1.2.2 Vision systems

By the use of a TV-camera objects may be identified by their shape or size. This approach may for instance be used in automatons for recirculation of bottles. The type of bottle must be recognized, as the amount reimbursed for a bottle depends on its type.

1.1.2.3 Magnetic stripe

Information contained in magnetic stripe is widely used on credit cards etc. Quite a large amount of information can be stored on the magnetic stripe, but specially designed readers are required and the information cannot be read by humans.

1.1.2.4 Barcode Recognition

A barcode is a machine-readable representation of information. Barcodes can be read by optical scanners called barcode readers or scanned from an image using software. A 2D barcode is similar to a linear, one-dimensional barcode, but has more data representation capability. Fig of 1-D barcode and 2-D barcode are given bellow.

Fig 1.1 1-D barcode The Gettysburg Address (UPC)

Fig 1.2 2-D barcode Universal Product Code

1.1.2.5 Magnetic Ink Character Recognition (MICR)

Printing in magnetic ink is mainly used within bank applications. The characters are written in ink that contains finely ground magnetic material and they are written in stylized fonts which are specifically designed for the application. Before the characters are read, the ink is exposed to a magnetic field. This process accentuates each character and helps simplify the detection. The characters are read by interpreting the waveform obtained when scanning the characters horizontally. Each character is designed to have its own specific waveform. Although designed for machine reading, the characters are still readable to humans. However, the reading is dependent on the characters being printed with magnetic ink.

1.1.2.6 Optical Mark Recognition (OMR)

OMR technology detects the existence of a mark, not its shape. OMR forms usually contain small ovals, referred to as 'bubbles,' or check boxes that the respondent fills in. OMR cannot recognize alphabetic or numeric characters. OMR is the fastest and most accurate of the data collection technologies. It is also relatively user-friendly. The accuracy of OMR is a result of precise measurement of the darkness of a mark, and the sophisticated mark discrimination algorithms for determining whether what is detected is an erasure or a mark.

Fig 1.3 The College Board SAT uses OMR technology

1.1.2.7 Optical Character Recognition

Optical character recognition is needed when the information should be readable both to humans and to a machine and alternative inputs cannot be predefined. In comparison with the other techniques for automatic identification, optical character recognition is unique in that it does not require control of the process that produces the information.

1.1.3 Speech Synthesis

Speech synthesis is the artificial production of human speech. Synthesizing is the very effective process of generating speech waveforms using machines based on the phonetical transcription of the message. Recent progress in speech synthesis has produced synthesizers with very high intelligibility but the sound quality and naturalness still remains a major problem.

1.1.3.1 Phonetics and Theory of Speech Production

Speech processing and language technology contains lots of special concepts and terminology. To understand how different speech synthesis and analysis methods work one must have some knowledge of speech production, articulatory phonetics, and some other related terminology. The basic theories related to these topics are described below.

1.1.3.1.1 Representation and Analysis of Speech Signals

Continuous speech is a set of complicated audio signals which makes producing them artificially difficult. Speech signals are usually considered as voiced or unvoiced, but in some cases they are something between these two. Voiced sounds consist of fundamental frequency (F0) and its harmonic components produced by vocal cords (vocal folds). The vocal tract modifies this excitation signal causing formant (pole) and sometimes antiformant (zero) frequencies. Each formant frequency has also amplitude and bandwidth and it may be sometimes difficult to define some of these parameters correctly. The fundamental frequency and formant frequencies are probably the most important concepts in speech synthesis and also in speech processing in general. With purely unvoiced sounds, there is no fundamental frequency in excitation signal and therefore no harmonic structure either and the excitation can be considered as white noise. The airflow is forced through a vocal tract constriction which can occur in several places between glottis and mouth. Some sounds are produced with complete stoppage of airflow followed by a sudden release, producing an impulsive turbulent excitation often followed by a more protracted turbulent excitation. Unvoiced sounds are also usually more silent and less steady than voiced ones. Speech signals of the three vowels (/a/ /i/ /u/) are presented in time- and frequency domain in Fig: 1.4. The fundamental frequency is about 100 Hz in all cases and the formant frequencies F1, F2, and F3 with vowel /a/ are approximately 600 Hz, 1000 Hz, and 2500 Hz respectively. With vowel /i/ the first three formants are 200 Hz, 2300 Hz, and 3000 Hz, and with /u/ 300 Hz, 600 Hz, and 2300 Hz. The harmonic structure of the excitation is also easy to perceive from frequency domain presentation.

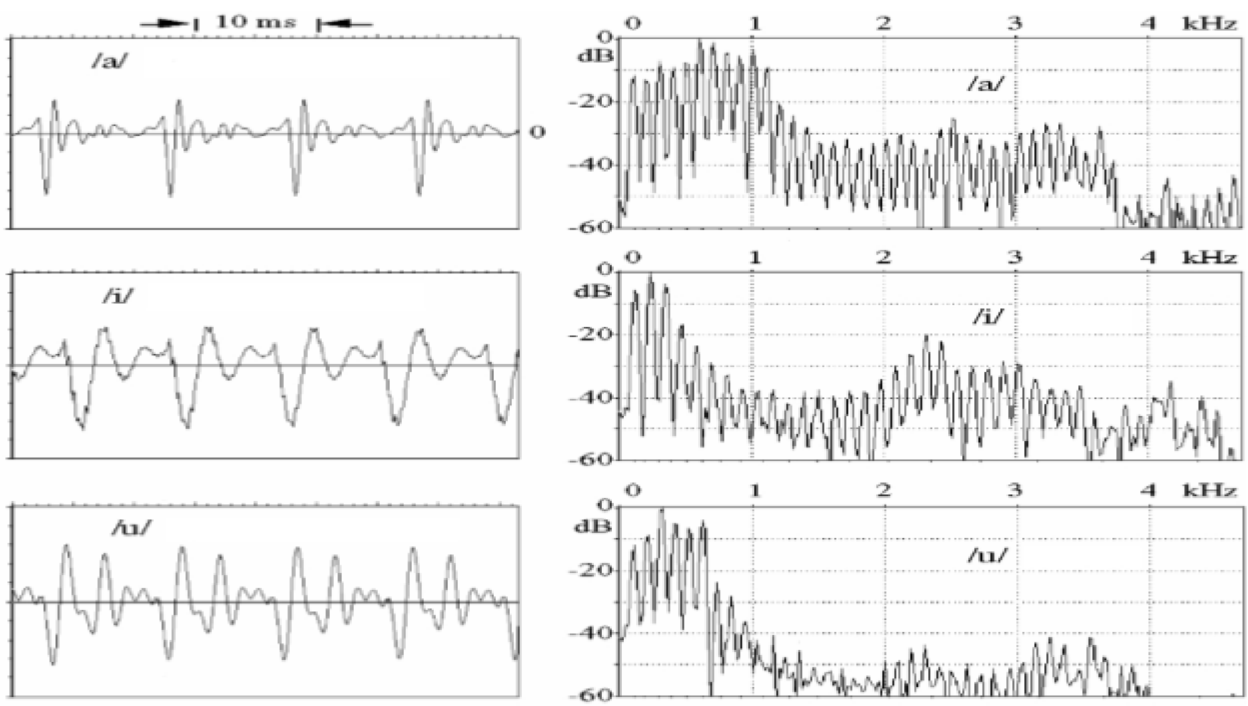


Fig 1.4 The time and frequency domain presentation of vowels /a/, /i/, and /u/.

For determining the fundamental frequency or pitch of speech, for example a method called cepstral analysis may be used. Cepstrum is obtained by first windowing and making Discrete Fourier Transform (DFT) for the signal and then logaritmizing power spectrum and finally transforming it back to the time-domain by Inverse Discrete Fourier Transform (IDFT). The procedure is shown in Fig 1.5.

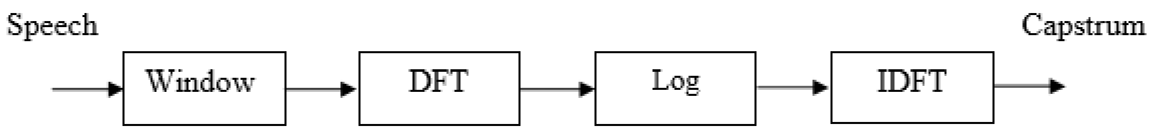


Fig 1.5 Cepstral analyses.

1.2 Problem Forulation

Voice output of printed or hand written text produced by OCR system with Speech synthesis gives very effective medium of communication. For some application speech (voice) communication is more useful than text. So, we have chosen to develop OCR based text to speech system using LabVIEW. The main objective of this report is to:

- Study Optical character recognition technology
- Study the speech synthesis technology
- Develop Optical character recognition using LabVIEW software
- Develop text to speech module using LabVIEW software
- Combine OCR and Text to speech module to obtain the desired result.

CHAPTER II

LITERATURE REVIEW

2.1 Literature Review

”High Quality Text to Speech Synthesis”, Rajiv Kumar Yadav, This paper describes the Methodically, character recognition is a subset of the pattern recognition area. However, it was character recognition that gave the incentives for making pattern recognition and image analysis matured fields of science. After character recognition these character are converted into speech. Speech is the vocalization form of human communication. Speech communication is more effective medium than text communication medium in many real world applications.

2.2 History of OCR

To replicate the human functions by machines, making the machine able to perform tasks like reading is an ancient dream. The origins of character recognition can actually be found back in 1870. This was the year that C.R.Carey of Boston Massachusetts invented the retina scanner which was an image transmission system using a mosaic of photocells. Two decades later the Polish P.Nipkow invented the sequential scanner which was a major breakthrough both for modern television and reading machines.

During the first decades of the 19’th century several attempts were made to develop devices to aid the blind through experiments with OCR. However, the modern version of OCR did not appear until the middle of the 1940’s with the development of the digital computer. The motivation for development from then on, was the possible applications within the business world.

2.2.1 The Start of OCR

By 1950 the technological revolution was moving forward at a high speed, and electronic data processing was becoming an important field. Data entry was performed through punched cards and a cost-effective way of handling the increasing amount of data was needed. At the same time the technology for machine reading was becoming sufficiently mature for application, and by the middle of the 1950's OCR machines became commercially available.

The first true OCR reading machine was installed at Reader's Digest in 1954. This equipment was used to convert typewritten sales reports into punched cards for input to the computer.

2.2.2 First Generation OCR

The commercial OCR systems appearing in the period from 1960 to 1965 may be called the first generation of OCR. This generation of OCR machines were mainly characterized by the constrained letter shapes read. The symbols were specially designed for machine reading, and the first ones did not even look very natural. With time multifont machines started to appear, which could read up to ten different fonts. The number of fonts were limited by the pattern recognition method applied, template matching, which compares the character image with a library of prototype images for each character of each font.

2.2.3 Second Generation OCR

The reading machines of the second generation appeared in the middle of the 1960's and early 1970's. These systems were able to recognize regular machine printed characters and also had hand-printed character recognition capabilities. When hand-printed characters were considered, the character set was constrained to numerals and a few letters and symbols.

The first and famous system of this kind was the IBM 1287, which was exhibited at the World Fair in New York in 1965. Also, in this period Toshiba developed the first automatic letter sorting machine for postal code numbers and Hitachi made the first OCR machine for high performance and low cost. In this period significant work was done in the area of standardization. In 1966, a thorough study of OCR requirements was completed and an American standard OCR character set was defined; OCR-A. This font was highly stylized and designed to facilitate optical recognition, although still readable to humans. A European font was also designed OCR-B, which had more natural fonts than the American standard. Some attempts were made to merge the two fonts into one standard, but instead machines being able to read both standards appeared.

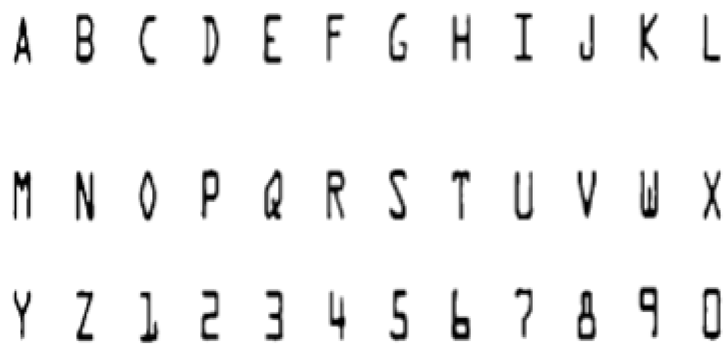


Fig 2.1 OCR-A.

Fig 2.2 OCR-B.

2.2.4 Third Generation OCR

For the third generation of OCR systems, appearing in the middle of the 1970's, the challenge was documents of poor quality and large printed and hand-written character sets. Low cost and high performance were also important objectives, which were helped by the dramatic advances in hardware technology.

Although more sophisticated OCR-machines started to appear at the market simple OCR devices were still very useful. In the period before the personal computers and laser printers started to dominate the area of text production, typing was a special niche for OCR The uniform print spacing and small number of fonts made simply designed OCR devices very useful. Rough drafts could be created on ordinary typewriters and fed into the computer through an OCR device for final editing. In these way word processors, which were an expensive resource at this time, could support several people and the costs for equipment could be cut.

2.2.5 OCR Today

Although, OCR machines became commercially available already in the 1950's, only a few thousand systems had been sold worldwide up to 1986. The main reason for this was the cost of the systems. However, as hardware was getting cheaper, and OCR systems started to become available as software packages, the sale increased considerably. Today a few thousand is the number of systems sold every week, and the cost of an omnifont OCR has dropped with a factor of ten every other year for the last 6 years.

2.3 Components of an OCR System

A typical OCR system consists of several components. In Fig 2.3 a common setup is illustrated. The first step in the process is to digitize the analog document using a digital scanner. Then extracted text will be pre-processed (binarization or thresholding), when the regions containing text are located, each symbol is extracted through a segmentation process.

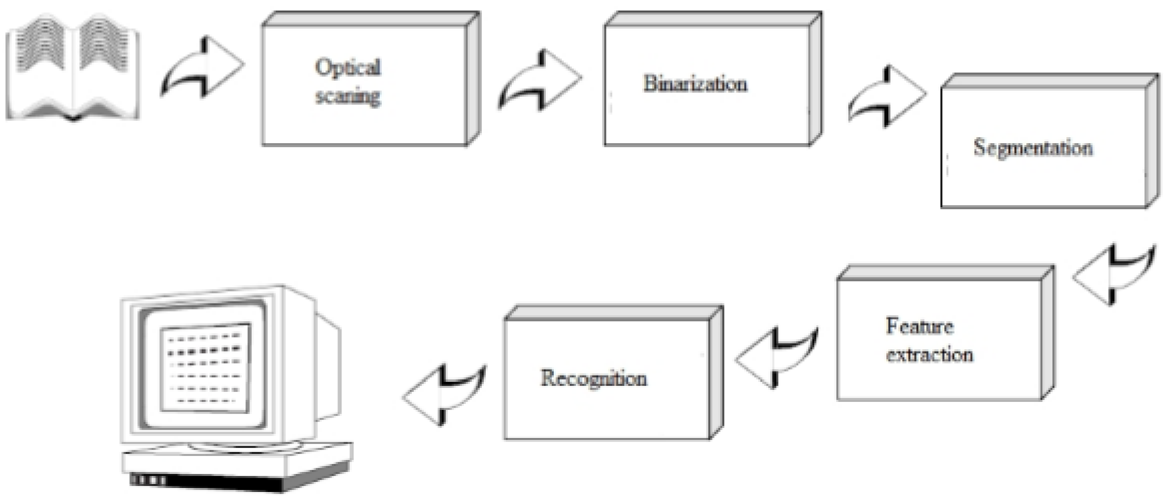


Fig 2.3 Components of an OCR-system

The identity of each symbol is found by comparing the extracted features with descriptions of the symbol classes obtained through a previous learning phase. Finally contextual information is used to reconstruct the words and numbers of the original text. In the next sections these steps and some of the methods involved are described in more detail.

2.3.1 Image Scanning

In computing, a scanner is a device that optically scans images, printed text, handwriting, or an object, and converts it to a digital image. Common examples found in offices are variations of the desktop (or flatbed) scanner where the document is placed on a glass window for scanning. Hand-held scanners, where the device is moved by hand, have evolved from text scanning "wands" to 3D scanners used for industrial design, reverse engineering, test and measurement, orthotics, gaming and other applications. Mechanically driven scanners that move the document are typically used for large-format documents, where a flatbed design would be impractical.

Modern scanners typically use a charge-coupled device (CCD) or a Contact Image Sensor (CIS) as the image sensor, whereas older drum scanners use a photomultiplier tube as the image sensor. A rotary scanner, used for high-speed document scanning, is another type of drum scanner, using a CCD array instead of a photomultiplier. Other types of scanners are planetary scanners, which take photographs of books and documents, and 3D scanners, for producing three dimensional models of objects.

Another category of scanner is digital camera scanners, which are based on the concept of reprographic cameras. Due to increasing resolution and new features such as anti-shake, digital cameras have become an attractive alternative to regular scanners. While they still having disadvantages compared to traditional scanners (such as distortion, reflections, shadows, low contrast), digital cameras offer advantages such as speed, portability, gentle digitizing of thick documents without damaging the book spine. New scanning technologies are combining 3D scanners with digital cameras to create full-color, photo-realistic 3D models of objects.

2.3.2 Binarization

With the advancement of technology and widespread use of colour and grayscale scanners, most images scanned now are grayscale. The reasons for not using colour images are the non-colour nature of some texts such as books, the long time needed for scanning, the large volume needed for storing color images and lack of appropriate methods for segmentation of colour images.

On the contrary, because of the complexity of the OCR operation, the input of the character recognition phase in most methods is binary images. Therefore, in the preprocessing phase, grayscale images are to be converted to binary images. The most common method is using a threshold. In this method, the pixels lighter than the threshold are turned to white and the remainder to black pixels. An important point to notice in here is to determine the threshold. In some methods in which the used pictures are very similar to each other, a fixed threshold is used.

So binarization is the process of converting a grayscale image (0 to 255 pixel values) into binary image (0 to1 pixel values) by thresholding. The binary document image allows the use of fast binary arithmetic during processing, and also requires less space to store.

2.3.3 Segmentation Process

Segmentation of text is a process by which the text is partitioned into its coherent parts. The text image contains a number of text lines. Each line again contains a number of words. Each word may contain a number of characters.

The following segmentation scheme is proposed where lines are segmented then words and finally characters. These are then put together to the effect of recognition of individual characters. The individual characters in a word are isolated. Spacing between the characters can be used for segmentation.

2.3.3.1 Line Segmentation

Line segmentation is the process of identifying lines in a given image. Steps for the line Segmentation is as follows

- Scan the BMP image horizontally to find first ON pixel and remember that y coordinate as y1.
- Continue scanning the BMP image then we would find lots of ON pixel since the characters would have started.
- Finally, we get the first OFF pixel and remember that y coordinate as y2.
- y1 to y2 is the line.

- Repeat the above steps till the end of the image.

2.3.3.2 Word Segmentation

As it's known that there is a distance between one word to another word. This concept will be use here for word segmentation. After the line segmentation scan the image vertically for word segmentation. Steps for the word Segmentation is as follows

Scan the BMP image vertically for the recognized line segment, to find first ON pixel and remember that x coordinate as x1. Treat this as starting coordinate for the word.

- Continue scanning the BMP image then we would find lots of ON pixel since the word would have started.
- Finally, we get the successive five (this is assumed word distance) OFF pixel column and remember that x coordinate as x2.
- x1 to x2 is the word.
- Repeat the above steps till the end of the line segment.
- Repeat the above steps for all the recognized line segments.

2.3.3.3 Character Segmentation

Character segmentation is the process of separation of characters word. Steps for the line Segmentation is as follows

- Scan the BMP image vertically for the recognized word segment, to find first ON pixel and remember that x coordinate as x1. Treat this as starting coordinate for the character.
- Continue scanning the BMP image then we would find lots of ON pixel since the characters would have started.
- Finally, we get the OFF pixel column and remember that x coordinate as x2.

- x1 to x2 is the character.
- Repeat the above steps till the end of the word segment, line segment.
- Repeat the above steps for all the recognized line segments [2, 25].

2.3.4 Feature Extraction

The objective of feature extraction is to capture the essential characteristics of the symbols, and it is generally accepted that this is one of the most difficult problems of pattern recognition. The most straight forward way of describing a character is by the actual raster image. Another approach is to extract certain features that still characterize the symbols, but leaves out the unimportant attributes. The techniques for extraction of such features are often divided into three main groups, where the features are found from:

- The distribution of points.
- Transformations and series expansions.
- Structural analysis.

The different groups of features may be evaluated according to their sensitivity to noise and deformation and the ease of implementation and use. The results of such a comparison are shown in table 2.1. The criteria used in this evaluation are the following:

I. Robustness.
    1. Noise.
       Sensitivity to disconnected line segments, bumps, gaps, filled loops etc.
    2. Distortions.
       Sensitivity to local variations like rounded corners, improper protrusions dilations and shrinkage.

3. Style variation.

   Sensitivity to variation in style like the use of different shapes to represent the same character or the use of serifs slants etc.

4. Translation.

   Sensitivity to movement of the whole character or its components.

5. Rotation.

   Sensitivity to change in orientation of the characters.

II. Practical use.

1. Speed of recognition.
2. Complexity of implementation.
3. Independence.

Each of the techniques evaluated in table 2.1 are described in the next sections.

Feature extraction technique	**Robustness**					**Practical use**		
	1	2	3	4	5	1	2	3
Template matching	◍	◍	○	○	○	○	●	○
Transformations	○	●	●	●	●	○	○	◍
Distribution of points: Zoning	○	◍	○	○	◍	●	●	○
Moments	◍	◍	○	●	●	○	◍	○
n-tuple	◍	○	◍	○	◍	●	●	◍
Characteristic loci	○	●	●	●	◍	●	●	○
Crossings	○	●	●	●	◍	●	●	○
Structural features	○	●	●	●	◍	●	○	●

● High or easy ◍ Medium ○ Low or difficult

Table 2.1 Evaluation of feature extraction techniques

2.3.4.1 Template-Matching and Correlation Techniques

These techniques are different from the others in that no features are actually extracted. Instead the matrix containing the image of the input character is directly matched with a set of prototype characters representing each possible class. The distance between the pattern and each prototype is computed, and the class of the prototype giving the best match is assigned to the pattern.

The technique is simple and easy to implement in hardware and has been used in many commercial OCR machines. However, this technique is sensitive to noise and style variations and has no way of handling rotated characters.

2.3.4.2 Feature Based Techniques

In these methods, significant measurements are calculated and extracted from a character and compared to descriptions of the character classes obtained during a training phase. The description that matches most closely provides recognition. The features are given as numbers in a feature vector, and this feature vector is used to represent the symbol.

2.3.4.3 Distribution of Points

This category covers techniques that extract features based on the statistical distribution of points. These features are usually tolerant to distortions and style variations. Some of the typical techniques within this area are listed below.

2.3.4.3.1 Zoning

The rectangle circumscribing the character is divided into several overlapping, or no overlapping, regions and the densities of black points within these regions are computed and used as features.

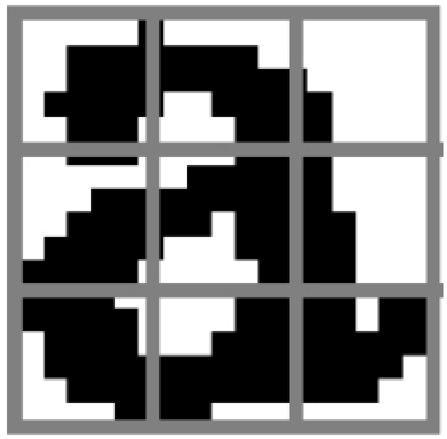


Fig 2.4 Zoning

2.3.4.3.2 Moments

The moments of black points about a chosen centre, for example the centre of gravity, or a chosen coordinate system, are used as features.

2.3.4.3.3 Crossings and Distances

In the crossing technique features are found from the number of times the character shape is crossed by vectors along certain directions. This technique is often used by commercial systems because it can be performed at high speed and requires low complexity. When using the distance technique certain lengths along the vectors crossing the character shapes are measured, for instance the length of the vectors within the boundary of the character.

2.3.4.3.4 N-tuples

The relative joint occurrence of black and white points (foreground and background) in certain specified orderings, are used as features.

2.3.4.3.5 Characteristic Loci

For each point in the background of the character, vertical and horizontal vectors are generated.

The numbers of times the line segments describing the character are intersected by these vectors are used as features.

2.3.5 Recognition

After we got the character by character segmentation we store the character image in a structure. This character as to be identified for the predefined character set.

There will be preliminary data will be stored for all characters for a identified font and size. This data contains the following information

- **a.** Character ASCII value
- **b.** Character name
- **c.** Character BMP image
- **d.** Character width and length
- **e.** Total number of ON pixel in the image.

For every recognized Character above mentioned information will be captured. The recognized character information will be compared with the predefined data which we have stored in the system.

As we are using the same font and size for the recognition there will be exact one unique match for the character. This will identify us the name of the character.

If the size of the character varies it will be scaled to the known standard and then recognizing process will be done.

2.4 Text to Speech Conversion System

Traditionally, Text-to-Speech (TTS) systems convert input text into voice by using a set of manually derived rules for prosody generation and/or voice synthesis. While these systems can achieve a high level of intelligibility, they typically sound unnatural. The process of deriving these rules is not only labour intensive but also difficult to8 generalize to a new language, a new voice, or a new speech style.

TTS can "read" text from a document, Web page or e-Book, generating synthesized speech. TTS programs can be useful for a variety of applications. For example, proofreading with TTS allows the author to catch awkward phrases, missing words or pacing problems.

TTS can also convert text files into audio MP3 files that can then be transferred to a portable MP3 player or CD-ROM. This can save time by allowing the user to listen to reports or background materials in bed, en route to a meeting, or while performing other tasks.

Even top screenwriting software includes TTS functionality so that a writer can assign different voices to characters in his or her script. The writer can then listen to the dialog to weed out stilted sentences. In the area of education, TTS programs provide a valuable edge, particularly for learning new languages. Speech engines are available in a variety of languages, including English, Spanish, German, French, and dozens more.

Fig 2.7 is a simple functional diagram of a general TTS synthesizer. A TTS system is composed of two main parts, the Natural Language Processing (NLP) module and the Digital Signal Processing (DSP) module.

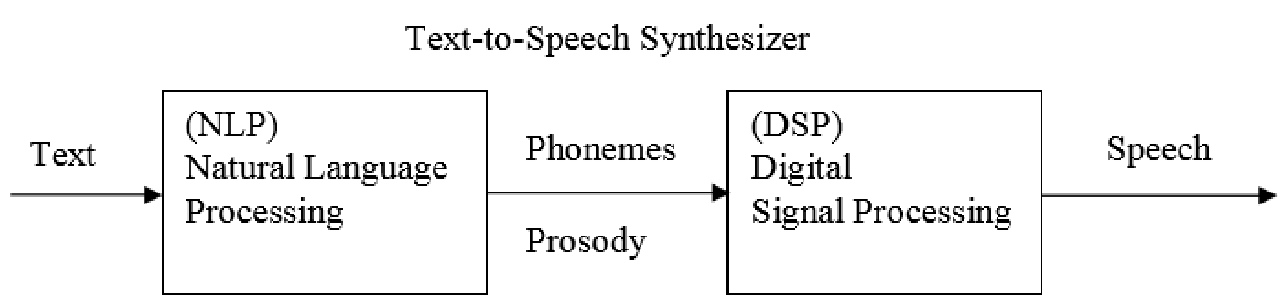


Fig 2.5 General TTS Synthesizer

The NLP module takes a series of text input and produces a phonetic transcription together with the desired intonation and prosody (rhythm) that is ready to pass on the DSP module. There are three major components within the NLP module, the letter-to-sound component, the prosody generation component, and the morpho syntactic analyser component.

The DSP module takes the phonemes and prosody that were generated by the NLP module and transforms them into speech. There are two main approaches used by DSP module: rule-based-synthesis approach and concatenative-synthesis approach.

2.4 Hardware Requirements

The system comprises of mostly software portion but had some hardware involved too. The hardware that we used was:

- P.C
- Speaker

2.5.1 Computers/Processors

OCR based Speech Synthesis System applications require a high processing speed computer system to perform specified task. It's possible to do 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. It requires the operating system and sound must be installed in PC.

2.5.2 Speaker

OCR based Speech Synthesis System applications requires a good quality, low cost speaker to produce a good quality of sound.

2.6 Software Platform

The software platform used here is LabVIEW (Laboratory Virtual Instrument Engineering Workbench).

2.6.1 LabVIEW

LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine the order of program execution, LabVIEW uses dataflow programming, where the flow of data through the nodes on the block diagram determines the execution order of the VIs and functions.VIs, or virtual instruments, are LabVIEW programs that imitate physical instruments.

In LabVIEW, user builds a user interface by using a set of tools and objects. The user interface is known as the front panel. User then adds code using graphical representations of functions to control the front panel objects. This graphical source code is also known as G code or block diagram code. The block diagram contains this code. In some ways, the block diagram resembles a flowchart.

2.6.2 Virtual Instruments

LabVIEW works on a data flow model in which information within a LabVIEW program, called a virtual instrument (VI), flows from data sources to data sinks connected by wires. The data can be modified as it is passed from source to sink by other VIs LabVIEW supports two types of VIs--internal VIs and user created VIs Internal VIs are packaged with LabVIEW and perform simple functions like adding numbers or opening files. User created VIs consists of both a graphical user interface called the front panel and a code pipeline called the block diagram. These VIs tend to be much more complex considering that they can contain any number of internal or user created VIs in an endless number of configurations.

2.6.3 LabVIEW Program Structure

A LabVIEW program is similar to a text-based program with functions and subroutines; however, in appearance it functions like a virtual instrument (VI). A real instrument may accept an input, process on it and then output a result. Similarly, a LabVIEW VI behaves in the same manner.

A LabVIEW VI has 3 main parts:

2.6.3.1 Front Panel window

Every user created VI has a front panel that contains the graphical interface with which a user interacts. The front panel can house various graphical objects ranging from simple buttons to complex graphs. Various options are available for changing the look and feel of the objects on the front panel to match the needs of any application.

2.6.3.2 Block Diagram window

Nearly every VI has a block diagram containing some kind of program logic that serves to modify data as it flows from sources to sinks. The block diagram houses a pipeline structure of sources, sinks, VIs, and structures wired together in order to define this program logic. Most importantly, every data source and sink from the front panel has its analog source and sink on the block diagram. This representation allows the input values from the user to be accessed from the block diagram. Likewise, new output values can be shown on the front panel by code executed in the block diagram.

2.6.3.3 Controls, Functions and Tools Palette

Windows, which contain icons associated with extensive libraries of software functions, subroutines, etc.

2.7 Software Implementation

LabVIEW software of OCR base speech synthesis system includes two steps:

- Optical character recognition
- Text to speech synthesis

2.7.1 Optical Character Recognition

In optical character recognition process image of printed text is used as input for OCR system.

In optical character recognition process five steps are involve:

- Image Acquisition
- Image Pre-processing (Binarization)
- Image Segmentation
- Template matching
- Recognition

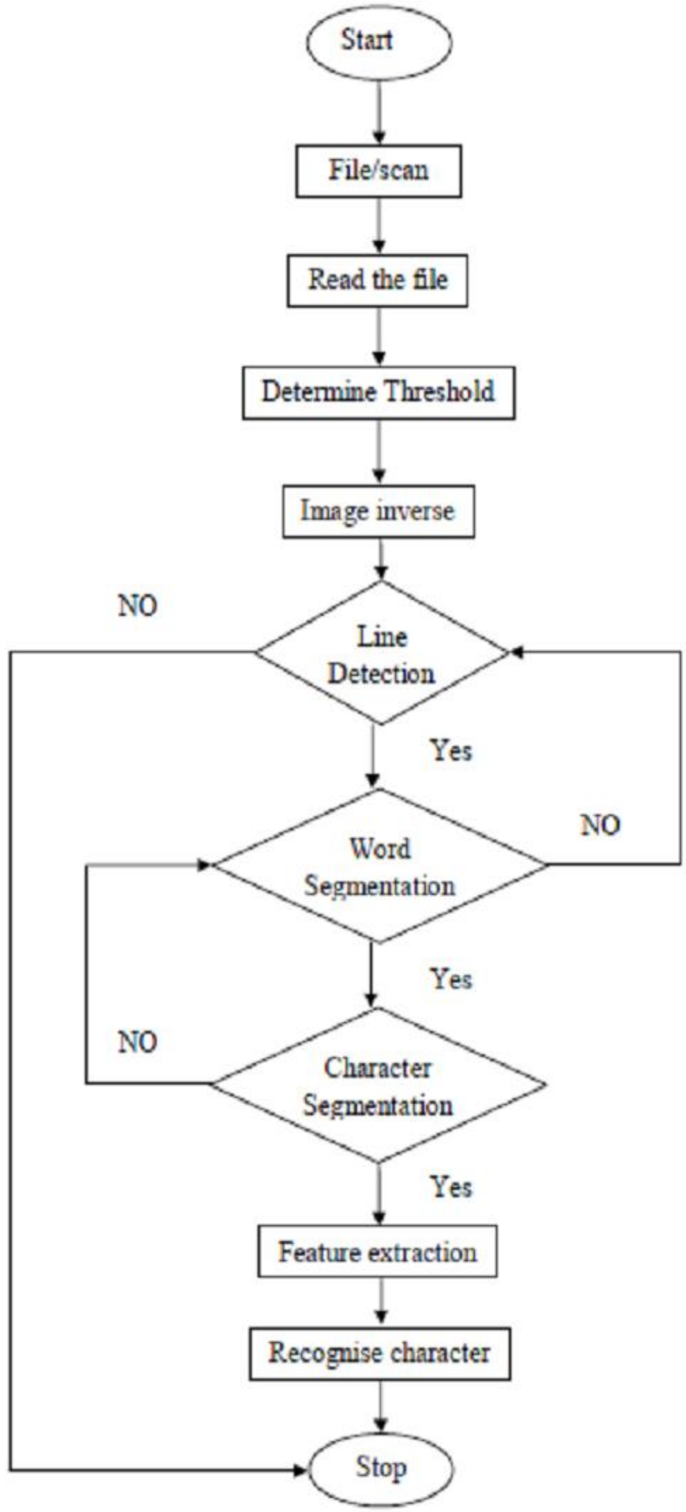


Fig 2.6 Flow chart OCR system

2.7.1.1 Image Acquisition

The image has been captured using a digital HP scanner. The flap of the scanner had been kept open during the acquisition process in order to obtain a uniform black background.

Image configuration has been done with the help of Imaq create subvi of LabVIEW. The configuration of the image means selecting the image type and border size (default is 3) of the image as per the requirement. Then Imaq file read subvi is use to read the file as shown in Fig 2.7

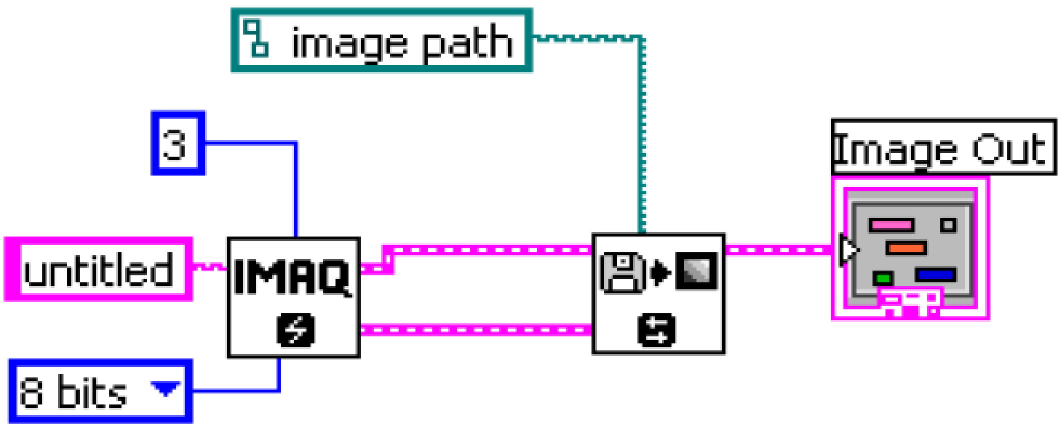


Fig 2.7 Image Configuration

2.7.1.2 Image Pre-processing (Binarization, Thresholding)

Binarization is the process of converting a grayscale image (0 to 255 pixel values) into binary image (0 to1 pixel values) by a threshold value of 175. the pixels lighter than the threshold are turned to white and the remainder to black pixels.

2.7.1.3 Image Segmentation

The input of this step is obtained thresholded image from above step. Three steps are involved in segmentation process are described below.

2.7.1.4 Template matching

Template matching is process in which correlation between stored templates and Segmented character will be finding in LabView by using correlation vi, which is described below.

2.7.1.4.1 Correlation

The correlation sub vi find best correlation between segmented character and stored templates of each character. Here two inputs first one is segmented character image and second one stored template image. Output of this is correlations between segmented character and every stored template.

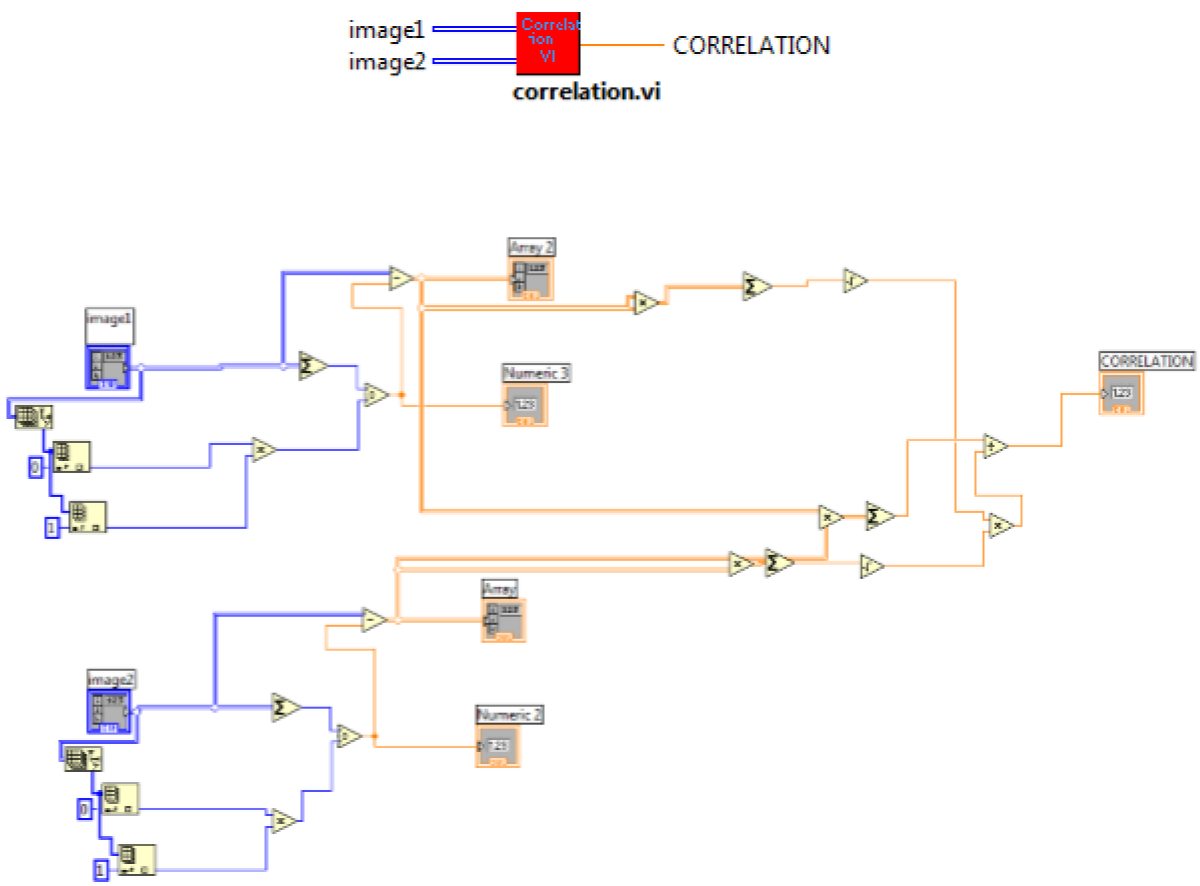


Fig 2.8 Block diagram of correlation vi

2.7.1.5 Recognition

After we got the character by character segmentation we store the character image in a structure. This character as to be identified for the pre-defined character set. There will be preliminary data will be stored for all characters for an identified font and size. This data contains the following information

- Character ASCII value
- Character name
- Character BMP image
- Character width and length
- Total number of ON pixel in the image.

For every recognized Character above mentioned information will be captured. The recognized character information will be compared with the pre-defined data which we have stored in the system.

As we are using the same font and size for the recognition there will be exact one unique match for the character. This will identify us the name of the character. If the size of the character varies it will be scaled to the known standard and then recognizing process will be done.

2.7.2 Text to speech synthesis

In text to speech module text recognized by OCR system will be the inputs of speech synthesis system which is to be converted into speech in .wav file format and creates a wave file named output .wav, which can be listen by using wave file player.

Two steps involved in text to speech synthesis

- Text to speech conversion
- Play speech in .wav file format

2.7.2.1 Text to speech conversion

In the text speech conversion input text is converted to speech (in LabVIEW) by using automation open, invoke node and property node will be described below in next section of this chapter. Flow chart text speech conversion is shown below in Fig 2.9.

In LabVIEW the ACTIVE X sub pallet in Communication pallet and its functions to exchange data between applications. ActiveX technology provides a standard model for inter application communication that different programming languages can implement on different platforms.

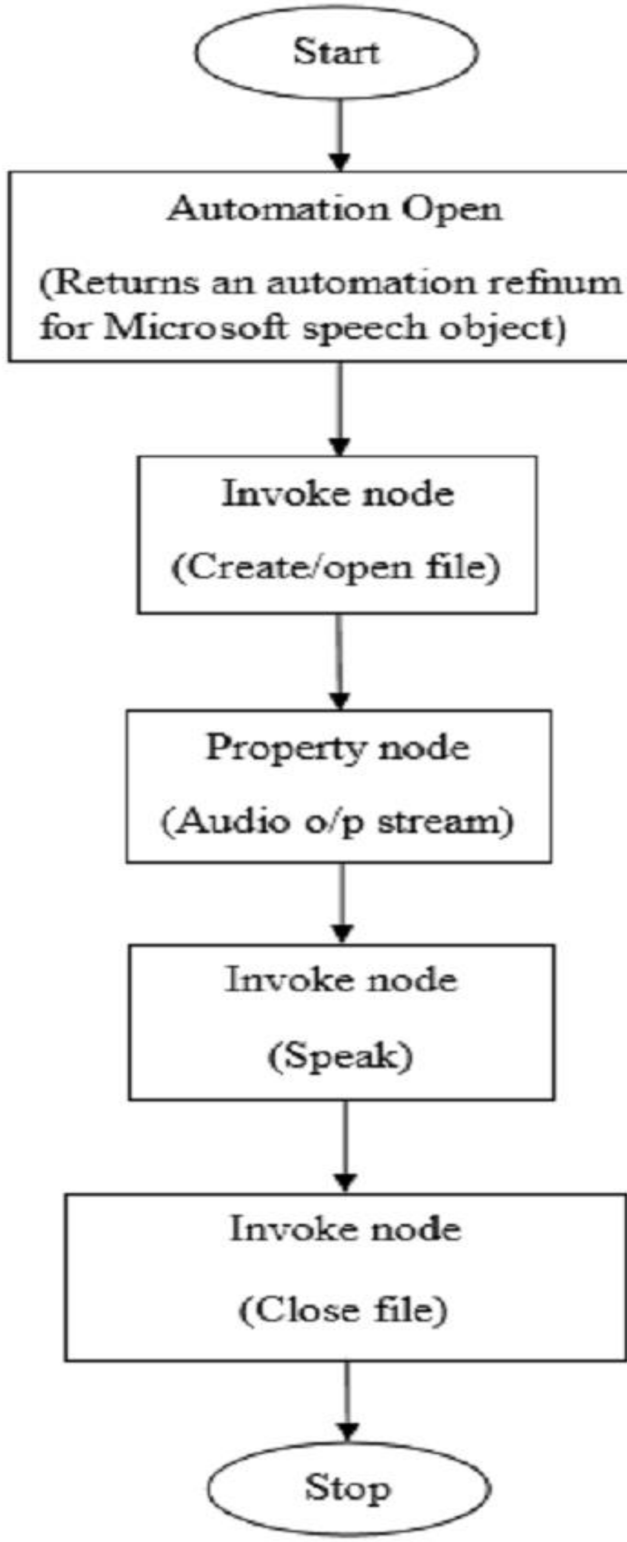


Fig 2.9 Flowchart for the text to speech wave file conversion

2.7.2.1.1 Overview of ActiveX

ActiveX is the general name for a set of Microsoft Technologies that allows you to reuse code and link individual programs together to suit your computing needs. Based on COM (Component Object Model) technologies, ActiveX is an extension of a previous technology called OLE (Object Linking and Embedding). Each program does not need to regenerate components, but rather, reuse components to give user the power to combine applications together. LabVIEW offers support for ActiveX automation as a server as well as support for ActiveX Containers, and ActiveX Events.

2.7.2.1.2 ActiveX **Automation**

ActiveX/COM refers to the process of controlling one program from another via ActiveX. Like networking, one program 2acts as the client and the other as the server. LabVIEW supports automation both as the client and the server. Both programs, client and server, exist independent of each other but are able to share information. The client communicates with the ActiveX objects that the server opens to allow the sharing of information. The automation client can access the object's properties and methods. Properties are attributes of an object. Another program can set or retrieve an object's attributes. Similarly, methods are functions that perform an operation on objects. Other applications can invoke methods. An example of an ActiveX property is the program name, height or width. An example of an ActiveX method is the save or print method.

2.7.2.1.3 ActiveX Automation with LabVIEW

LabVIEW as an ActiveX server or ActiveX client can interface with other programs from the LabVIEW programming interface. In this case, LabVIEW acts as the automation client and requests information of the automation server, or other program. Likewise, other ActiveX automation clients can interface with the LabVIEW ActiveX automation server.

2.7.2.1.4 LabVIEW as an Automation Client

LabVIEW provides functions in its API that allow LabVIEW to act as an automation client with any automation server. The diagram below shows LabVIEW's programming flow, and gives the associated functions with each block.

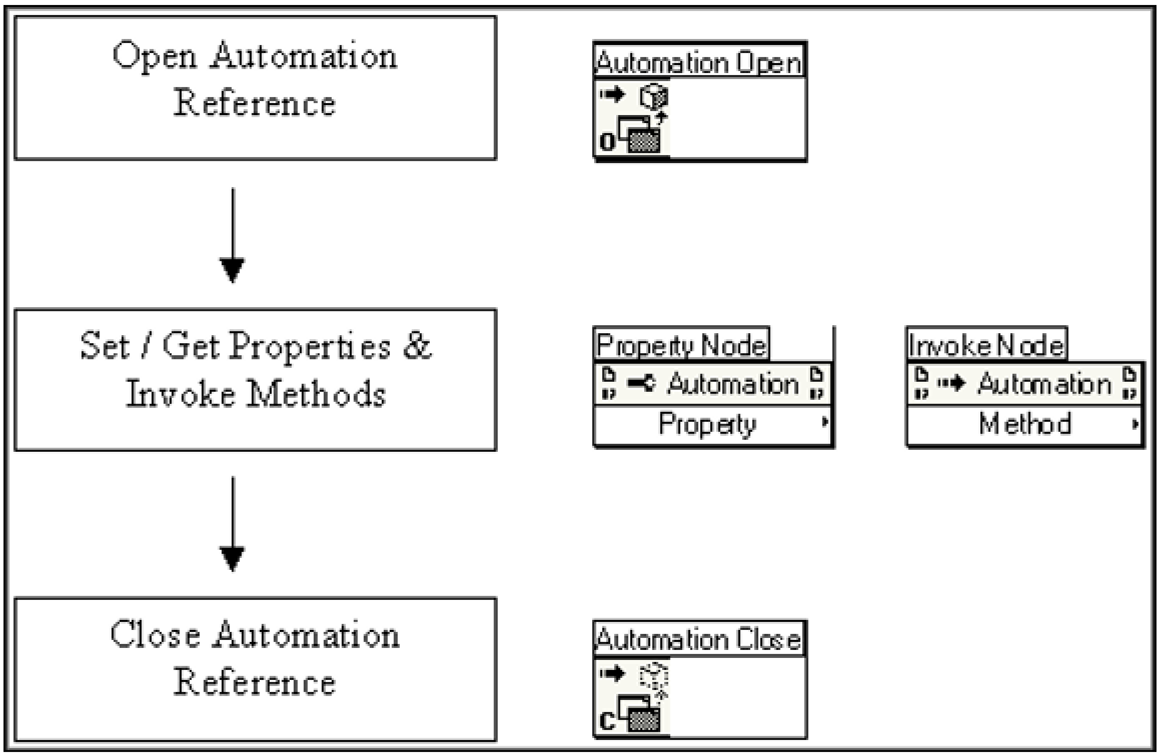
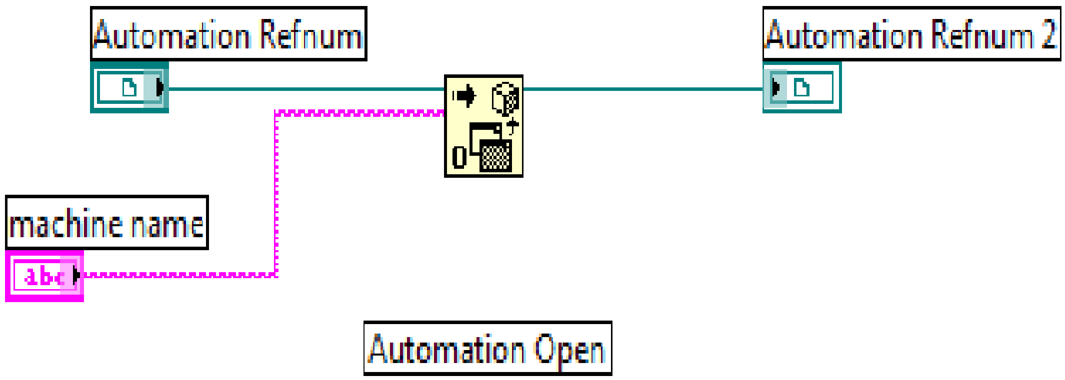


Fig 2.10 Programming flow of ActiveX used in LabVIEW

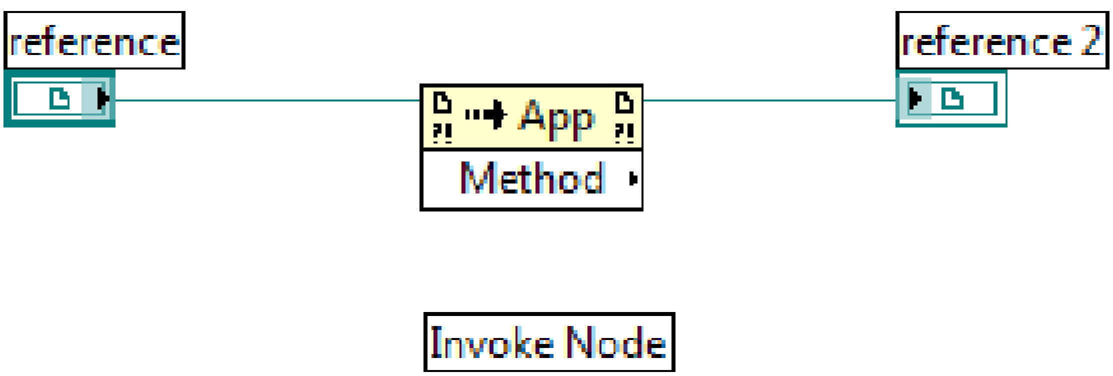
2.7.2.1.5 Automation Open (Windows)

Returns an automation refnum, which points to a specific ActiveX object. In Text to Speech VI, it gives refnum for Microsoft speech object library.



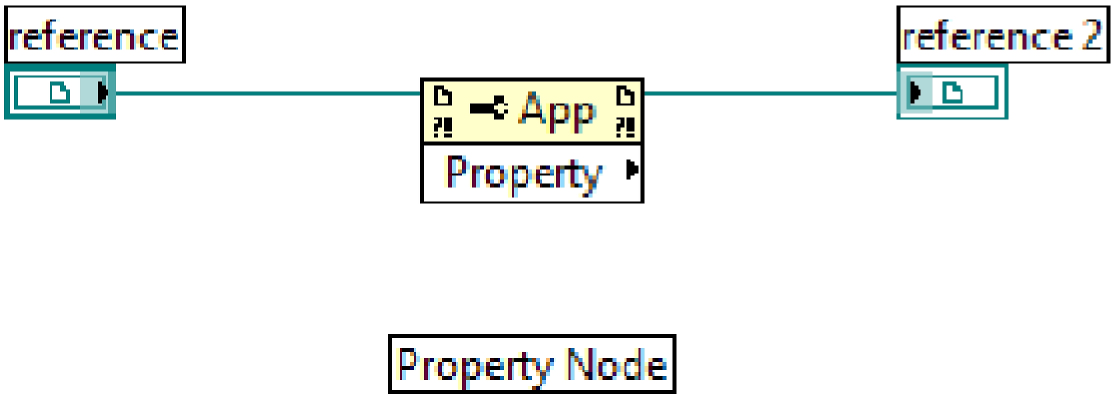
2.7.2.1.6 Invoke Node

Invokes a method or action on a reference. Most methods have associated parameters. If the node is conFigd for VI Server Application class or Virtual Instrument class and reference is unwired, reference defaults to the current Application or VI.



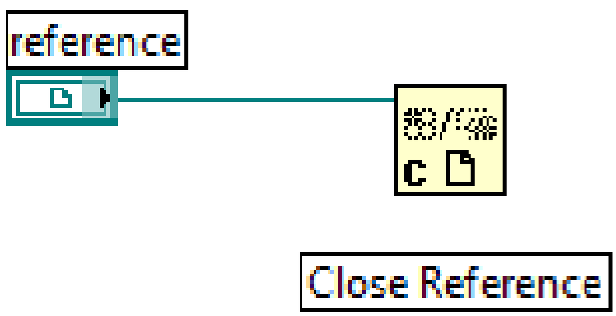
2.7.2.1.7 Property Node

Gets (reads) and/or sets (writes) properties of a reference. The Property Node automatically adapts to the class of the object that you reference. LabVIEW includes Property Nodes preconFigd to access VISA properties and ActiveX properties.



2.7.2.1.8 Close Reference

Closes a refnum associated with an open VI, VI object, an open instance of LabVIEW, or an ActiveX or .NET object.



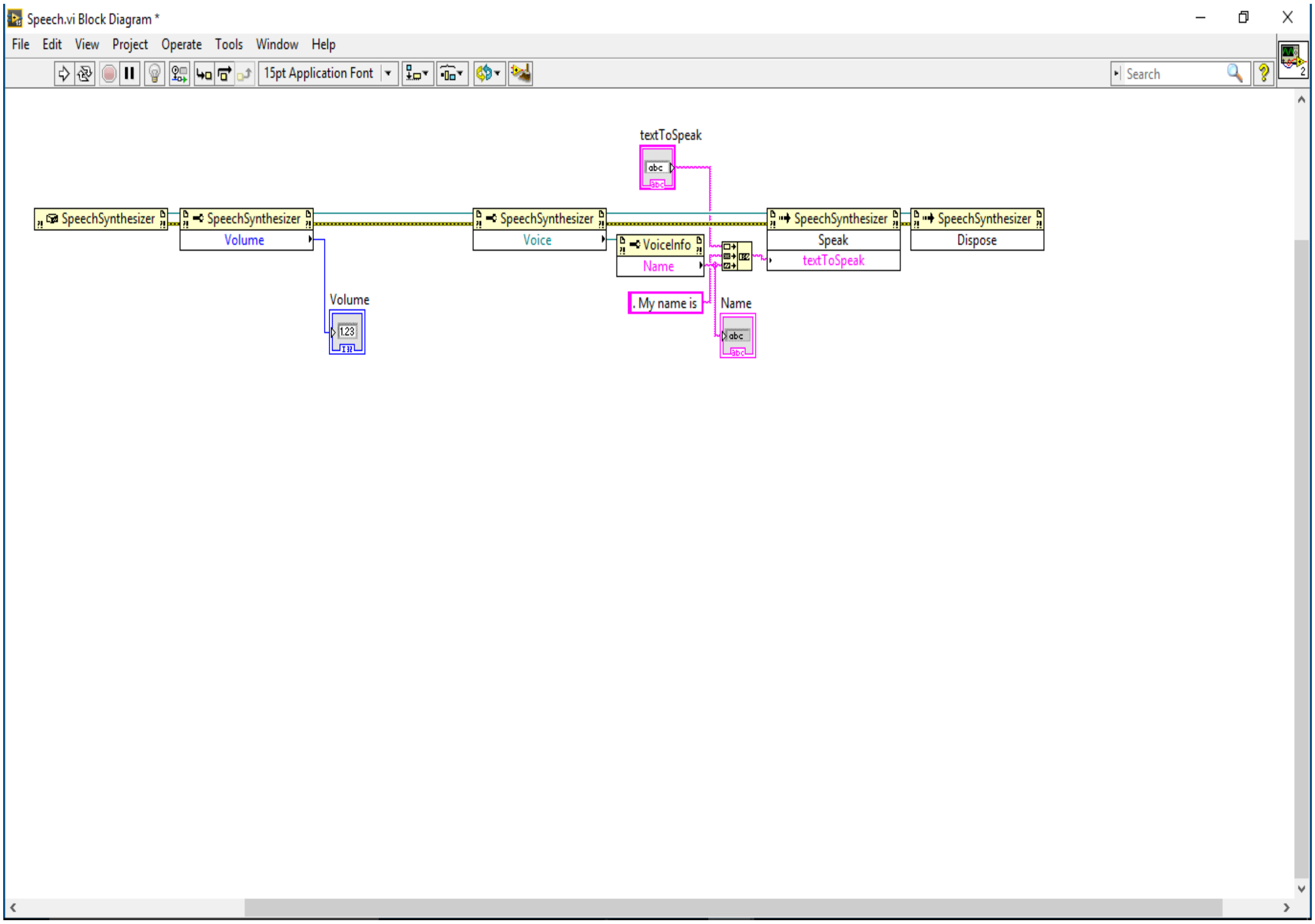


Fig 2.11 Block diagram of Text to Speech Synthesis

CHAPTER III

RESULT

3.1 Result and Discussion

3.1.1 LabVIEW Front Panel

Experiments have been performed to test the proposed system. Here whole system is implemented using LabVIEW 7.1 version. The front panel of OCR based speech recognition system is shown below in Fig 3.1.

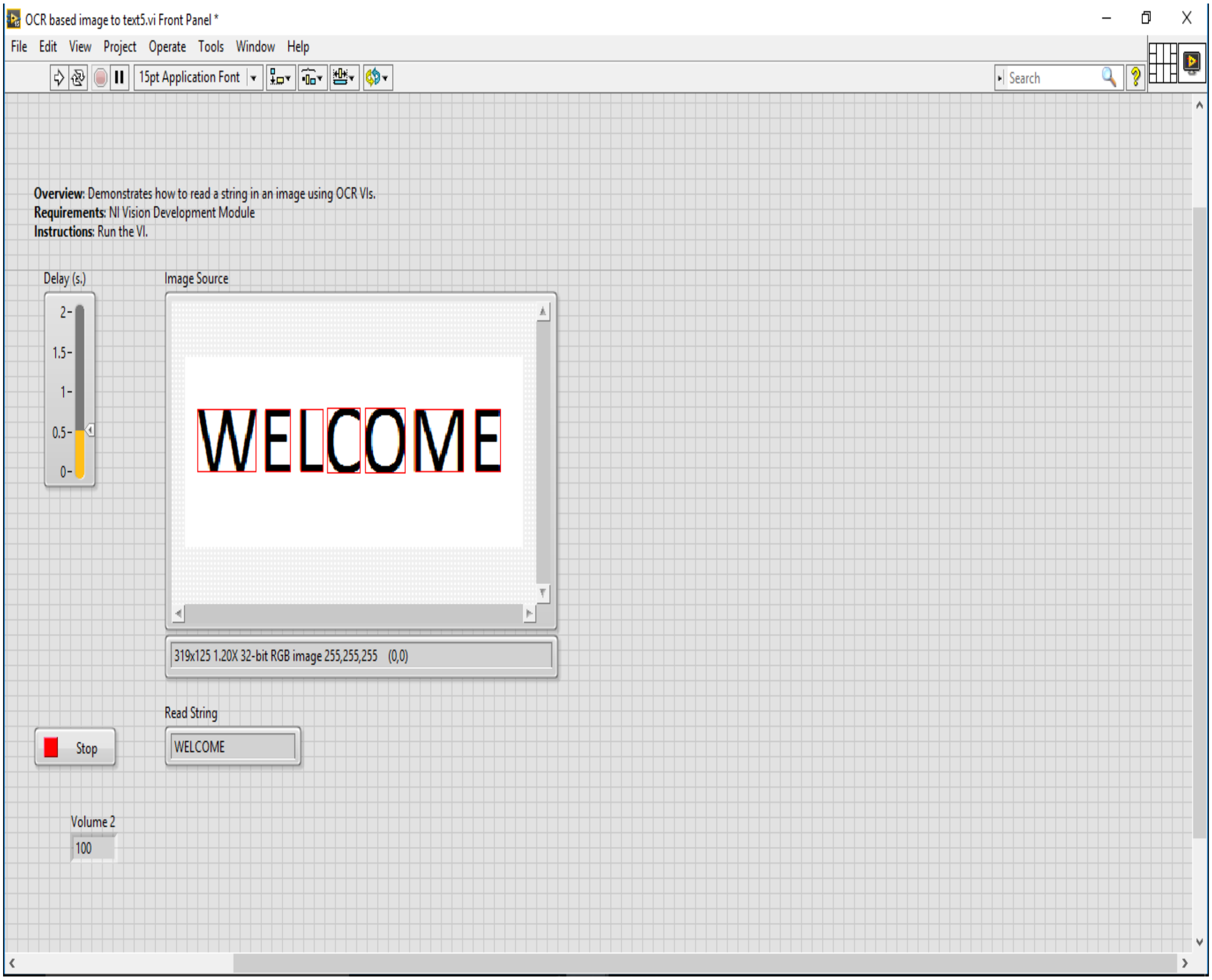


Fig 3.1 Front Panel of OCR based Speech Synthesis System

3.1.2 LabVIEW Block Diagram

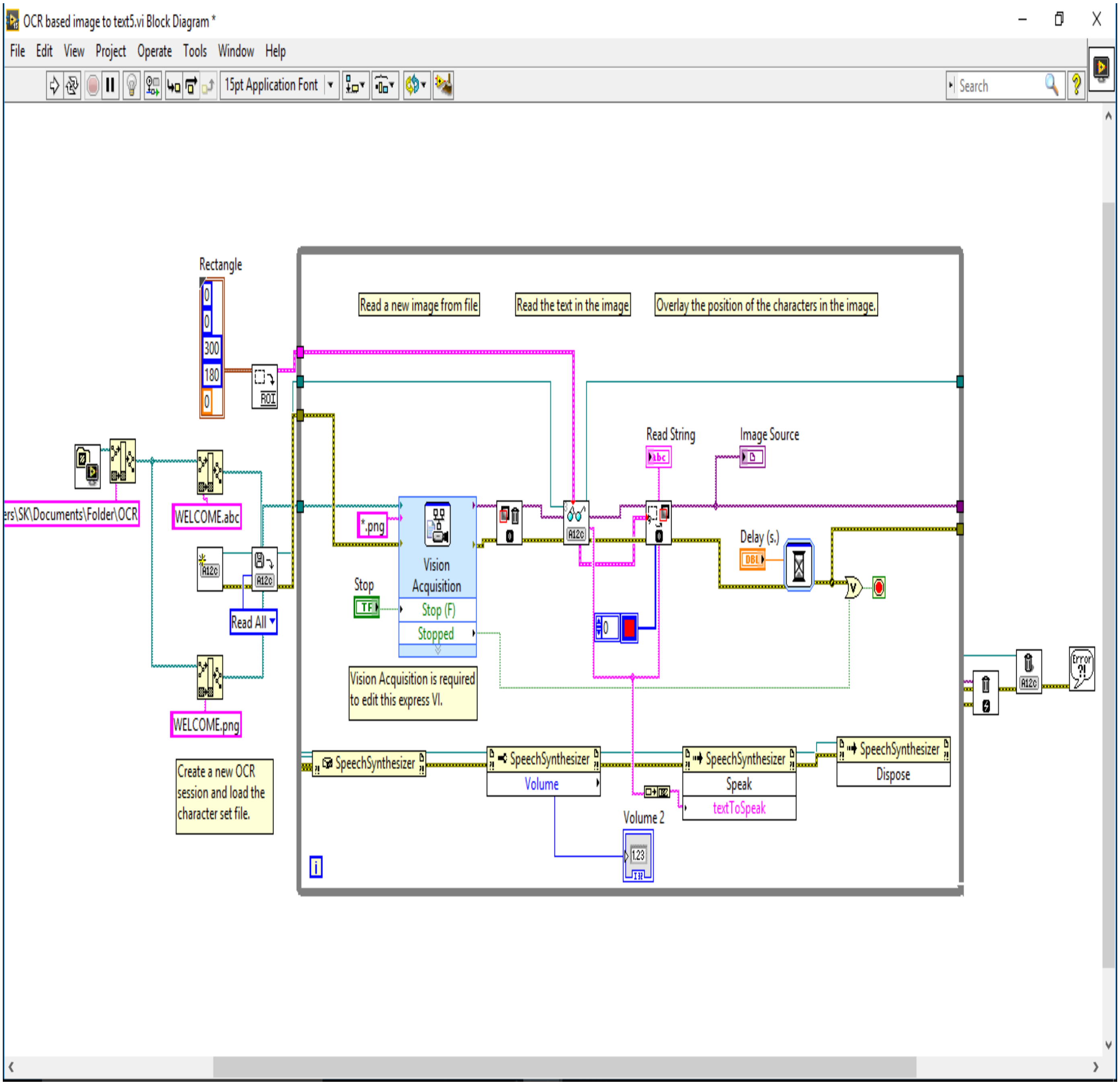


Fig 3.2 Block Diagram of OCR based Speech Synthesis System

The above diagram shows the Block Diagram of OCR based Speech Synthesis System. It contains Vision Acquisition, Text to Speech Synthesizer, and NI Vision Assistant.

3.1.3 NI OCR Training Interface

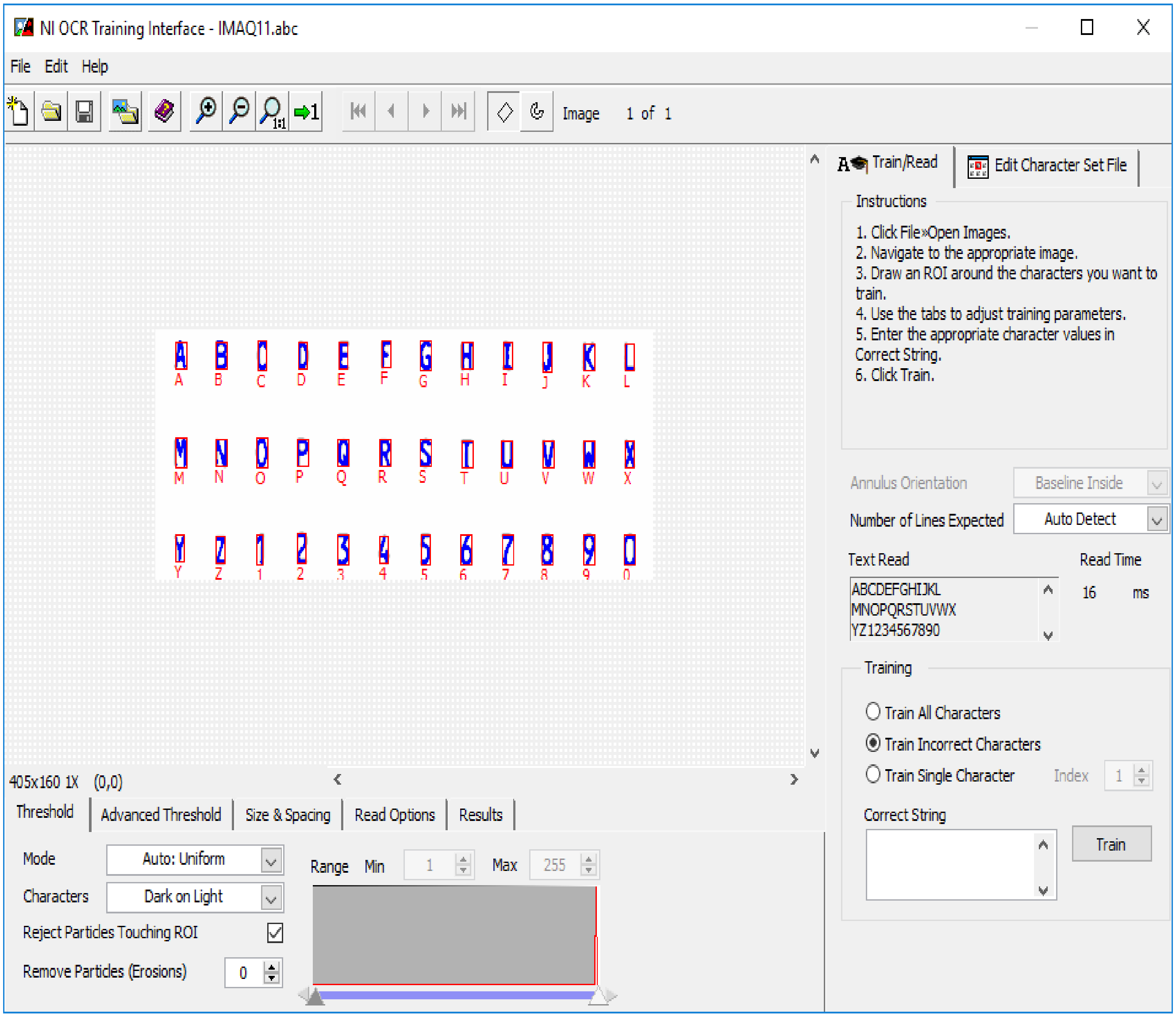


Fig 3.3 NI OCR Training Interface

The above diagram shows the NI Vision Assistant and Training Interface of OCR. In this diagram we have to train the Characters of the text in the image.

3.1.4 NI Vision Acquisition Software

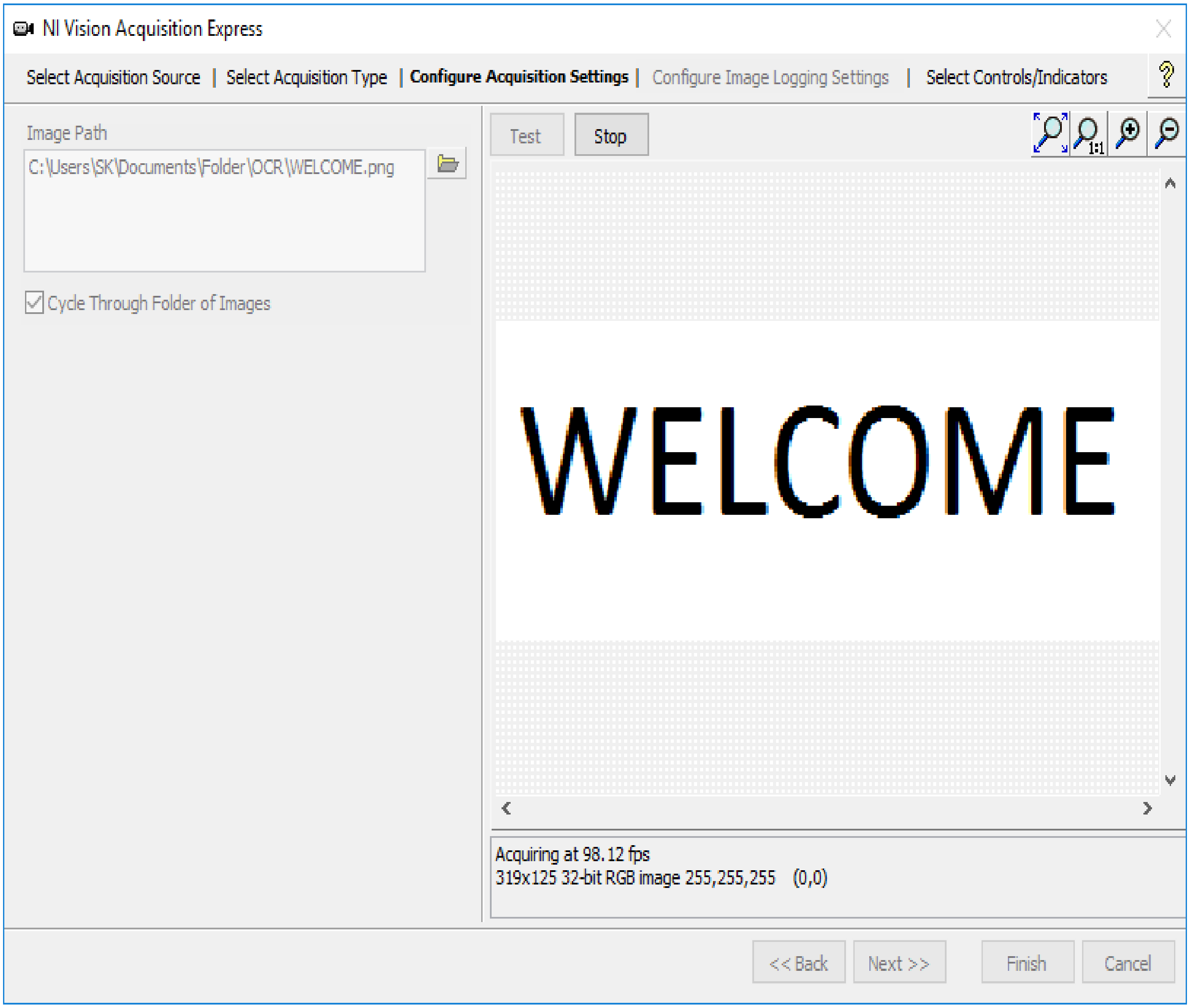


Fig 3.4 NI Vision Acquisition Software

The above diagram shows the NI Vision Acquisition Software and this software is used to Acquiring the images.

These are the output of the OCR Based Speech Synthesis System.

CHAPTER IV

CONCLUSION AND FUTURE WORKS

4.1 CONCLUSION

This thesis work describes OCR based Speech Synthesis System to produces a wave file output can be used as a good mode of communication between people. The system is implemented on LabVIEW 7.1 platform. There is two session of system first is OCR and second is Speech Synthesis. In OCR printed or written character documents are scanned and image is acquired by using IMAQ Vision for LabVIEW and then characters are recognized using segmentation and template matching methods developed in LabVIEW. In second section recognized text is converted into speech. The ACTIVE X sub pallet in Communication pallet is used to exchange data between applications. ActiveX technology provides a standard model for inter application communication that different programming languages can implement on different platforms. Microsoft Speech Object Library (Version 5.1) has been used to build speech-enabled applications, which retrieve the voice and audio output information available for computer. This library allows selecting the voice and audio device one would like to use, OCR recognized text to be read, and adjust the rate and volume of the selected voice. The application developed is user friendly, cost effective and gives the result in the real time. Moreover, the program has the required flexibility to be modified easily if the need arises.

4.2 FUTURE SCOPE

OCR base Speech recognition system using LabVIEW is an efficient program giving good results for specific fonts (equal to or above to 48 font size) still there are chances to improve it. The system can be improved by making it omnifont. Because there was no OCR base Speech recognition system implemented using Lab VIEW there is a good future scope to develop it using other methods more fast and efficient.

REFERENCES

1. T. Dutoit (1994) "High quality text-to-speech synthesis: a comparison of four candidate algorithms," Acoustics, Speech, and Signal Processing. ICASSP-94., 1994 IEEE International Conference on, vol.1, no.5, pp.I/565-I/568 vol.1, pp.19-22.
2. B.M. Sagar, Shobha G, R. P. Kumar (2008) “OCR for printed Kannada text to machine editable format using database approach” WSEAS Transactions on Computers Vol.7, pp.766-769.
3. G. Nagy (1992) "At the frontiers of OCR," Proceedings of the IEEE, vol.80, no.7, pp.1093-1100.
4. Landt, Jerry (2006) "Shrouds of Time: The history of RFID," AIM, Inc.
5. R.C. Palmer, “The Bar Code Book,” Helmers Publishing.