REAL TIME IMPLEMENTATION OF IMAGE RECOGNITION AND TEXT TO SPEECH CONVERSION

K.Kalaivani, R.Praveena,V.Anjalipriya,R.Srineena
Department of Electronics and Instrumentation Engineering, Easwari Engineering College, Bharathi Salai, Ramapuram Chennai-89

Abstract
This paper introduces an innovative, efficient and real-time cost beneficial technique that enables user to hear the contents of documents/text images instead of reading through them. It combines the concept of Optical Character Recognition (OCR) and Text to Speech Synthesiser (TTS) in MATLAB R2011b. This kind of system enables visually impaired people to interact with computers effectively through vocal interface. It also enables users to listen the audio created from the contents of web pages, e-books, documents and almost all electronically available data thereby enhancing the human-computer interactions. It is used to develop a system which extracts text data from images containing horizontally aligned text using Optical Character Recognition (OCR) and speaks out the extracted text data using Text to Speech synthesizer (TTS).

1. Introduction
Image acquisition, recognition and speech conversion using Optical Character Recognition (OCR) and Text to Speech synthesizer (TTS) by MATLAB is an Image Processing Technology used to convert the image containing horizontal text into text documents and the extracted text is converted into speech.

2. Basic working of the system
When the text book is placed under the mechanical set up which consists of web camera, captures the images from the text book. The captured image is placed in the created GUI in MATLAB and various image processing techniques like conversion of rgb to gray scale image, contrast adjustment, adaptive threshold can be carried out and recognizes the segmented characters. These segmented characters are given as input to the Optical Character Recognition (OCR) to obtain the converted text. The text document is converted into speech using Text to Speech synthesizer (TTS).

3. Existing methodology
As for now, the old method to perform text to speech conversion is followed. It requires a text document mandatory to convert it into speech. A person has to type the text from the images of the books. Next, the converted text is sent to the Text to Speech synthesizer (TTS) for speech conversion. This is the only existing methodology in our country.

4. Proposed methodology
Our methodology involves two main modules. The first stage involves acquiring image from the web camera and converting it into text document using Optical Character Recognition (OCR). The second stage involves natural language processing and digital signal processing for converting the text into speech using Text to Speech synthesizer (TTS).

4.1 Steps involved in our methodology
1. Image acquisition by the web camera
2. Loading the image into the axial panel of the created Graphical User Interface (GUI) in MATLAB R2011b.
3. Pre-processing of the image (RGB to gray image, contrast adjustment, adaptive threshold)
4. Converting pre-processed image into text document using OCR
5. Converting text document into speech using TTS.

5. Image analysis
Image analysis is the extraction of meaningful information from images; mainly from digital images by means of digital image processing techniques. Many important image analysis tools such as edge detectors and neural networks are inspired by human visual perception models. Computer image analysis largely contains the fields of computer or machine vision, and medical imaging, and makes heavy use of pattern recognition, digital geometry, and signal processing.

5.1 Lossy Image Compression
Lossy compression provides higher levels of data reduction but results in a less than perfect reproduction of the original image. It provides high compression ratio. Lossy image compression is useful in applications such as broadcast television, video conferencing and facsimile transmission in which a certain amount of error is an
acceptable trade-off for increased compression performance.

5.2 Discrete Cosine Transformation
A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio and images to spectral methods for the numerical solutions of partial differential equations. The use of cosine rather than sine function is critical in these applications: for compression, it turns out that cosine functions are much more efficient, whereas differential equations the cosines express a particular choice of boundary conditions.

In particular, a DCT is Fourier related transform similar to the Discrete Fourier Transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry, wherein some variants the input and output data are shifted by half a sample. There are eight standard DCT variants, of which four are common. The most common variant of Discrete Cosine Transform is the type-II DCT, which is often called simply “the DCT”; its inverse, the type-III DCT, is correspondingly often called simply “the inverse DCT” or “the IDCT”.

5.3 Histogram
In statistics, a histogram is a graphical representation showing a visual impression of the distribution of the data. It is an estimate of the probability distribution of a continuous variable and was first introduced by Karl Pearson. A histogram consists of tabular frequencies, shown as adjacent rectangles, erected over discrete intervals (bins), with an area equal to the frequencies of the observation in the interval.

6. Image pre-processing steps

6.1 Conversion of rgb image to gray scale image
RGB image is an image which is made up of three different colors namely red, green and blue respectively. Gray scale image is an image which is a combination of black and white colors respectively.

rgb2gray is the command used in MATLAB to convert RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance. I = rgb2gray (RGB) converts the true color image RGB to the grayscale intensity image I. rgb2gray converts RGB values to grayscale values by forming a weighted sum of the R, G, and B components: 0.2989 * R + 0.5870 * G + 0.1140 * B

6.2 Contrast adjustments
Contrast adjustment is the process of separating the black and white colors in an image. Imadjust - Adjust image intensity values or colormap.

J = imadjust(I) maps the values in intensity image I to new values in J such that 1% of data is saturated at low and high intensities of I. This increases the contrast of the output image J.

6.3 Adaptive Threshold
Adaptive threshold is the process of choosing the threshold to minimize the intra class variance of the thresholded black and white pixels. Graythresh - Global image threshold using Otsu's method. LEVEL = graythresh(I) computes a global threshold (LEVEL) that can be used to convert an intensity image to a binary image with IM2BW. LEVEL is a normalized intensity value that lies in the range [0, 1].

im2bw - Convert image to binary image by thresholding.

im2bw produces binary images from indexed, intensity, or RGB images. To do this, it converts the input image to grayscale format (if it is not already an intensity image), and then converts this grayscale image to binary by thresholding. The output binary image BW has values of 1 (white) for all pixels in the input image with luminance greater than LEVEL and 0 (black) for all other pixels.

7. Optical Character Recognition
OCR is the acronym for Optical Character Recognition. This technology allows a machine to automatically recognize characters through an optical mechanism.

Most of the designs in OCR follow a modification of this architecture. Given a page for recognition, first it is preprocessed. The aim of the preprocessing module is to prepare the image for recognition. Preprocessing involves binarization, skew correction and normalization. It undergoes some image enhancements such as filtering out noise and increasing the contrast. Segmentation occurs at two levels. On the first level, text, graphics and other parts are separated. On the
second level, text lines, words and characters in the
image are located.
Segmentation is followed by feature extraction, which is
concerned with the representation of the object. Feature
extraction and classification are the heart of OCR. Feature
extraction is expected to make the image invariant to rotation, translation, scaling, line-thickness, etc. It could also remove redundant information to compress the data amount and a lot of other things. The classifier is then trained with the extracted features for classification task. The classification stage identifies each input character image by considering the detected features. A range of classifiers are in use for these purpose. For improving the recognition result, post processing module is incorporated. The post-processor is typically intended to improve accuracy by detection and correction of OCR errors.

7.1 An Overview of the Tesseract OCR Engine
Tesseract is an open-source OCR engine that was
developed at HP between 1984 and 1994. Tesseract had
a significant lead in accuracy over the commercial
engines, but did not become a product. The next stage of
its development was back in HP Labs Bristol as an
investigation of OCR for compression. In late 2005, HP
released Tesseract for open source.
The first step is a connected component analysis in
which outlines of the components are stored. Tesseract
was probably the first OCR engine able to handle white-
on-black text so trivially. At this stage, outlines are
gathered together, purely by nesting, into Blobs. Blobs
are organized into text lines, and the lines and regions
are analyzed for fixed pitch or proportional text. Text
lines are broken into words differently according to the
kind of character spacing. Fixed pitch text is chopped
immediately by character cells. Proportional text is
broken into words using definite spaces and fuzzy
spaces.
Recognition then proceeds as a two-pass process. In the
first pass, an attempt is made to recognize each word in
turn. Each word that is satisfactory is passed to an
adaptive classifier as training data. The adaptive
classifier then gets a chance to more accurately
recognize text lower down the page.

7.1.2 Baseline Fitting
Once the text lines have been found, the baselines are
fitted more precisely using a quadratic spline. This was
another step for an OCR system, and enabled Tesseract
to handle pages with curved baselines, which are a
common artifact in scanning, and not just at book
bindings.

7.1.3 Word Recognition
Part of the recognition process for any character
recognition engine is to identify how a word should be
segmented into characters. The initial segmentation
output from line finding is classified first. The rest of the
word recognition step applies only to non-fixed pitch
text.

7.1.4 Chopping Joined Characters
While the result from a word is unsatisfactory, Tesseract
attempts to improve the result by chopping the blob with
worst confidence from the character classifier. Candidate
chop points are found from concave vertices of a
polygonal approximation of the outline, and may have
either another concave vertex opposite, or a line
segment. It may take up to 3 pairs of chop points to
successfully separate joined characters from the ASCII
set.

7.1.5 Static Character Classifier
An early version of Tesseract used topological features
developed from the work of Shillman. Though nicely
independent of font and size, these features are not
robust to the problems found in real life images, as
Bokser describes. An intermediate idea involved the use
of segments of the polygonal 14 approximation as
features, but this approach is also not robust to damaged
characters.

7.1.6 Classification
Classification proceeds as a two-step process. In the first
step, a class pruner creates a shortlist of character classes
that the unknown might match. Each feature fetches,
from a coarsely quantized 3-dimensional lookup table, a
bit-vector of classes that it might match, and the bit-
vectors are summed over all the features.

7.1.7 Adaptive Classifier
It has been suggested and demonstrated that OCR
ingines can benefit from the use of an adaptive
classifier. Since the static classifier has to be good at
generalizing to any kind of font, its ability to
discriminate between different characters or between
characters and non-characters is weakened. A more font-
sensitive adaptive classifier that is trained by the output
of the static classifier is therefore commonly used to
obtain greater discrimination within each document, where the number of fonts is limited.

8. Text To Speech Synthesis (TTS)
Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diaphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output.

The quality of a speech synthesizer is judged by its similarity to the human voice, and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer.

Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

8.1 Microsoft Speech Application Programming Interface (SAPI)
The Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications. Till date, a number of versions of the API have been released, which have shipped either as part of a Speech SDK, or as part of the Windows OS itself. Applications that use SAPI include Microsoft Office, Microsoft Agent and Microsoft Speech Server.

In general all versions of the API have been designed such that a software developer can write an application to perform speech recognition and synthesis by using a standard set of interfaces, accessible from a variety of programming languages.

Typically in SAPI 5 applications issue calls through the API (for example to load a recognition grammar, start recognition, or provide text to be synthesized). The sapi.dll runtime component interprets these commands and processes them, where necessary calling on the engine through the engine interfaces.

The recognition and synthesis engines also generate events while processing (for example, to indicate an utterance has been recognized or to indicate word boundaries in the synthesized speech). These pass in the reverse direction, from the engines, through the runtime dll, and on to an event sink in the application.

9. Graphical User Interface Design Environment (GUIDE)
GUIDE is both a directory and a function.
GUIDE - Open the GUI Design Environment. GUIDE - initiates the GUI Design Environment (guide) tools that allow GUls to be created or edited interactively from FIG-files or handle(s) to figure.

Calling guide by itself will open the guide Quick Start Dialog where you can choose to open a previously created GUI or create a new one from one of the provided GUI templates.

guide (filename) - opens the FIG-file named 'filename' for editing if it is on the MATLAB path. guide (full path) opens the FIG-file at 'full path even if it is not on the MATLAB path. guide (handle list) opens the content of each of the figures in handle list in a separate copy of the guide design environment.

10. Conclusion
In our project by using Optical Character Recognition, ASCII formatted text file can be made from the raw images of books and the converted text can be made as speech through Text to Speech Synthesizer. Through this method, we can make editing process of books or web pages easier. We can save the audio file for future use also.

Like any system in this world, this system also has few disadvantages. The MATLAB has to be worked upon by trained people only. For acquiring images during night, special lighting devices are necessary.

10.1 Future Scope
Since printed document images archived by many applications are more of historical and poor in quality,
there is a need to apply advanced image pre-processing techniques for document analysis.

Document image processing algorithms for document image collections need more progress. Schemes that learn from document image collections itself for better performance are needed. Recognition from poor quality documents results in a number of recognition errors. Retrieval of documents in this situation requires more functionality. Effective schemes for retrieval in presence of OCR errors there is need to develop Multilingual OCR system so that we can read more than one language documents for document analysis.

References