DOI: http://dx.doi.org/10.26483/ijarcs.v12i2.6690

Volume 12, No. 2, March-April 2021

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SMART APPLICATION FOR THE VISUALLY- IMPAIRED

Olumide Olayinka Obe And Akinwonmi A. E.
Department of Computer Science,
Federal University of Technology,
Akure, Nigeria

Abstract: In this work, an application that would allow recognizing objects from images recorded by the camera of a mobile device was developed. An android phone camera was used to take images of some objects and then store them in the android database and the name of each object was stored in an audio mode. The SIFT (Scale-Invariant Feature Transform) was applied for the development of the application. To improve the performance of the application, one of the fastest corner detection algorithms, the Features from Accelerated Segment Test (FAST) algorithm was implemented. Since the algorithm was implemented on a smartphone, OpenCV for Android SDK was used. The cascaded filters approach was used by SIFT to detect scale-invariant characteristic points, where the difference of Gaussians (DoG) was calculated on rescaled images progressively. A blob detector based on the Hessian matrix to find points of interest was used by SURF. To measure local change around the points, the determinant of the Hessian matrix was used, and points were chosen based on where this determinant is maximal. The determinant of the Hessian was used by SURF to select scale. The auditory presentation of object recognition results to the blind user was done through a pre-recorded message. 97% accuracy is recorded in the performance of the system.

Keywords: Visual impaired, Android, character recognition, image binarisation, Hessian matrix

I. INTRODUCTION

From the statistics of the World Health Organization (WHO) as of April 2018, a sum of 285 million visually impaired individual lives across the world [1]. These visually impaired and blind people face numerous challenges in routine tasks such as selecting objects for their personal use or identifying obstacles on their way while walking down the road. Identification of objects and movement in the surroundings is a primary challenge for blind people which normally sighted people take for granted. As far as outdoor activities are concerned the blind indicate difficulties in safe and independent mobility depriving them of normal professional and social life [2]. The ability of people who are blind or have significant visual impairments to identify objects on their own will certainly enhance independent living and foster economic and social self-sufficiency. Finally, the common problem experienced by the blind is the so-called activities of daily living.

Today, there is an increasing interest in developing technologies that attempt to help visually impaired people in their daily lives. However, it is shown that the object identification task is still the major difficulty for blind people. Although many applications can be used for this task, there are still obvious limitations that require more improvement. Among these available technologies, some dedicated devices can be found for navigation and object recognition. These wearable devices have the disadvantage that they are expensive in comparison to the software. Also, blind users are required to carry many gadgets and devices, each for a different purpose such as object identifiers, navigators, and mobile phones. Software that will run on the blind user's smartphones will help to eliminate the need for carrying devices for identifying objects.

The other solutions available at present include the Low-tech labeling systems in which labels are attached to objects, e.g. with tactile signs or text messages in Braille [3],[4] and the High-tech systems that employ 1-D and 2-D barcodes, talking labels, or radio-frequency identification devices (RFID). RFID can be used to search for objects at short distances to which RFID tag was applied using an acoustic signal, but a big limitation to its usage is that it is very hard for blind users to find the position of the bar code and to correctly point the bar code reader at the bar code. Some reading-assistive systems such as pen scanners might be employed in these and similar situations. Such systems integrate optical character recognition (OCR) software to offer the function of scanning and recognition of text and some have integrated voice output [5]. Both the Low-tech and the High-tech systems, however, require attaching special tags or visual signs to the objects. Consequently, they can be costly, since such systems need to be regularly maintained to keep them up to date.

This work hereby aims to provide some assistance to the blind via a software application based on the Android platform which will help ease the difficulty of performing the task of object identification. In this work, the focus is on the group of daily living tasks related to personal care systems, in particular, systems that enable the visually impaired to identify objects, e.g. food packs, medicine containers, and other items. The software is based on an image processing system running on the Android platform. The object recognition notifications are communicated to the blind user utilizing pre-recorded verbal messages.

1.1 Background of the Study

Before the advent of technologies that aid object identification for the blind, blind people have been living dependently on other people who are not visually impaired for their daily activities. Nowadays, with the rapid development in the field of mobile technology, different IT-based assistive technologies have been developed to provide a better quality of life for these people who have special needs such as visual impairment. These technologies have contributed mainly to helping blind people to interact efficiently with social activities and increasing their ability for having independent lives. This is can be seen through different applications used for path guiding, obstacle detection, searching, and identifying objects [6].

There are a lot of challenges that face visually impaired people in performing their daily tasks especially accessing information about surrounding objects. For these reasons, there is an increasing interest in developing effective solutions that can help them in recognizing any objects. These solutions were designed using different technologies such as image processing, which includes optical character recognition (OCR), color identifiers, brightness identifiers, and object recognition algorithms. Furthermore, barcode, RFID (radiofrequency identification devices), tactile signs, and Braille have been used for a wide range of applications. However, there are still some limitations of these techniques which present the need for designing a solution that attempts to address these issues.

To solve this visual problem faced by the blind, this project focuses on the design and implementation of an android application for visual product identification for the blind which will detect an object and analyze it from a built-in database and give vocal information about the object.

II. REVIEW OF RELATED LITERATURE

Over the past several years, much work has been done on various assistive applications and devices for people with special needs. Further, there are many new and general mobile image recognition applications. Google Goggles [7] which identifies landmarks and logos, to recognize the book and CD covers, SnapTell (www.snaptell.com), and other applications are in use. Over the past few decades also, many currency recognition applications were developed but with limitations and having specific environments. An object recognition tool for android smartphones was developed in [8]. This tool has three image processing modules: a color detection module, a light detector module, and an image recognition module. The Image recognition module is a composite task that includes many steps such as key points detection, calculation of the description, and comparing the resulted description with other image descriptions in the database to recognize the object. [9] design a system for helping blind people to navigate through undefined places and to identify objects. Usually, blind people try to identify objects by tangible contact. However, RFID technology is used for the presented systems to help blind people in identifying objects in addition to the navigation. In this system, the RFID electronic tags are placed under specific objects and the database is installed in a remote computer that maintains information of electronic tags. The RFID reader that embedded in the user cane will generate radio Frequency for transfer data from the tag chip, and the chip will send back the unique tag's id, and this tag's id is transferred and processed in a remote computer by taking the related info based on the tag's identity code. The blind people have a wireless headset and it is connected with the computer and RFID reader. Therefore, the computer will provide the information in a voice representation to the blind person. design a system for helping blind people to navigate through undefined places and to identify objects. Usually, blind people try to identify objects by tangible contact. However, RFID technology is used for the presented systems to help blind people in identifying objects in addition to the navigation. In this system, the RFID electronic tags are placed under specific objects and the database is installed in a remote computer that maintains information of electronic tags. The RFID reader that embedded in the user cane will generate radio Frequency for transfer data from the tag chip, and the chip will send back the unique tag's id, and this tag's id is transferred and processed in the remote computer by taking the related info based on the tag's identity code. The blind people have a wireless headset and it is connected with the computer and RFID reader. Therefore, the computer will provide the information in a voice representation to the blind person. In [10], a shopping facilitator for visually impaired people was designed. They used RFID technology and PIC microcontroller in the supermarkets. The system using two Tarang-F4 versions of ZigBee transceiver modules, which is the protocol for communication on wireless networks used to transfer the information between devices. One of the Zigbee modules is to interface with the microcontroller and the other one is connected with a PC for transfer data. The goods are organized on shelves with passive RFID tags which have unique waves of radiofrequency that are generated by an RFID reader and the RFID reader reads and sends unique code to the microcontroller. [11] is probably the best known assistive application for object recognition. It uses a combination of object recognition, machine translation, photo-OCR, and human annotators to extract information from an image. Originally, the application would only allow for the limited number of recognition attempts and then require a user to buy credits. As of 2017, this limitation seems to be lifted and the application can be used free of charge, without any explicitly stated limits (there is probably still some kind of fair-user-policy on the number of recognitions performed). TapTapSee is an on-line solution and cannot work without an Internet connection. The main drawback of this application is that the recognition is very slow. In some cases, it may take more than 30 seconds.

III. METHOD

The goal of this work is to design an application that would allow recognizing objects from images recorded by the camera of a mobile device. The SIFT (Scale-Invariant Feature Transform) was applied for the development of the application. The SIFT is considered a very powerful computer vision algorithm for detecting and describing local image features. SIFT allows computing feature descriptors strongly independent of the image registration conditions. To improve the performance of the application the Features from Accelerated Segment Test (FAST) algorithm was implemented. It is one of the fastest corner detection algorithm.

An android phone camera was used to take images of some objects and then store them in the android database and the name of each object was stored in an audio mode. To make sure the object is visible in the camera view, a mobile camera with a sufficiently large optical sensor and good clarity was used. Since the algorithm was implemented on a smartphone, OpenCV for Android SDK was used.

The cascaded filters approach was used by SIFT to detect scale-invariant characteristic points, where the difference of Gaussians (DoG) was calculated on rescaled images progressively.

A blob detector based on the Hessian matrix (equation 1) to find points of interest was used by SURF. To measure local change around the points, the determinant of the Hessian matrix was used, and points were chosen based on where this determinant is maximal. The determinant of the Hessian was used by SURF to select scale.

$$H(x,\sigma) = \begin{bmatrix} Lxx(x,\sigma) & Lxy(x,\sigma) \\ Lxy(x,\sigma) & Lyy(x,\sigma) \end{bmatrix} \quad (1)$$

This was used at a given point p = (x, y) and scale σ.

Where $L_{xx}(p, \sigma)$, etc. is the convolution of the second-order derivative of Gaussian with the image at the point x.

The auditory presentation of object recognition results to a blind user was done through a pre-recorded message.

3.1 System Flowchart and Architecture

The flowchart for the proposed methodology is shown in the following figure.

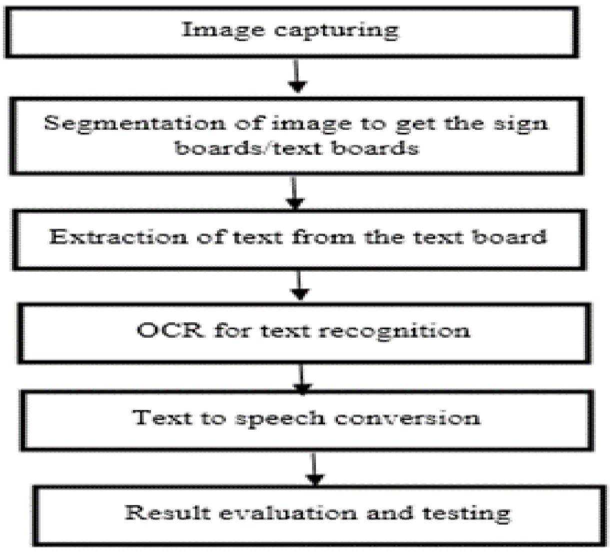

Figure 1. Flowchart for the proposed system

From figure 1, an input image is captured from the camera. Then a segmentation algorithm is applied to perform segmentation of the desired part of the image from signboards/text boards. Image segmentation is an essential process for most image analysis techniques. Then Extraction of text from the text board is performed by using the image processing technique. Text recognition is proposed to be done by optical character recognition. Optical character recognition (OCR), is the electronic conversion of photographed images of typewritten or in print text into computer-readable text. Then the obtained text is converted into a speech which is output and finally results in evaluation and testing.

3.2 Phases of Design

Optical character recognition (OCR) is the mechanical or electronic conversion of images of typed, handwritten, or printed text into machine-encoded text. It is widely used as a form of data entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as machine translation, text-to-speech, key data, and text mining.

A. Pre-processing: OCR software often "pre-processes" images to improve the chances of successful recognition. Techniques include:

De-skew – If the document was not aligned properly when scanned, it may need to be tilted a few degrees clockwise or counterclockwise to make lines of text perfectly horizontal or vertical.

Despeckle – remove positive and negative spots, smoothing edges

Binarisation – Convert an image from color or greyscale to black-and-white. The task of binarisation is performed as a simple way of separating the text from the background.

Line removal – Cleans up non-glyph boxes and lines

Layout analysis or "zoning" – Identifies columns, paragraphs, captions, etc. as distinct blocks. Especially important in multi-column layouts and tables.

Line and word detection – Establishes a baseline for word and character shapes, separates words if necessary.

Script recognition – In multilingual documents, the script may change at the level of the words, and hence, identification of the script is necessary before the right OCR can be invoked to handle the specific script.

Character isolation or "segmentation" – For per-character OCR, multiple characters that are connected due to image artifacts must be separated; single characters that are broken into multiple pieces due to artifacts must be connected. Normalize aspect ratio and scale-segmentation of fixed-pitch fonts is accomplished relatively simply by aligning the image to a uniform grid based on where vertical grid lines will least often intersect black areas. For proportional fonts, more sophisticated techniques are needed because whitespace between letters can sometimes be greater than that between words, and vertical lines can intersect more than one character.

B. Character recognition: There are two basic types of core OCR algorithms, which may produce a ranked list of candidate characters. Matrix matching involves comparing an image to a stored glyph on a pixel-by-pixel basis; it is also known as "pattern matching", "pattern recognition", or "image correlation". This relies on the input glyph being correctly isolated from the rest of the image, and on the stored glyph being in a similar font and at the same scale. This technique works best with typewritten text and does not work well when new fonts are encountered. This is the technique the early physical photocell-based OCR implemented, rather directly. Feature extraction decomposes glyphs into "features" like lines, closed loops, line direction, and line intersections. These are compared with an abstract vector-like representation of a character, which might reduce to one or more glyph prototypes. General techniques of feature detection in computer vision applied to this type of OCR, which is commonly seen in "intelligent" handwriting recognition and indeed most modern OCR software. Nearest neighbor classifiers such as the k-nearest neighbor algorithm are used to compare image features with stored glyph features and choose the nearest match.

C. Post-processing: OCR accuracy can be increased if the output is constrained by a lexicon – a list of words that are allowed to occur in a document. This might be, for example, all the words in the English language, or a more technical lexicon for a specific field. This technique can be problematic if the document contains words not in the lexicon, like proper nouns. Tesseract uses its dictionary to influence the character segmentation step, for improved accuracy. The output stream may be a plain text stream or file of characters, but more sophisticated OCR systems can preserve the original layout of the page and produce, for example, an annotated PDF that includes both the original image of the page and a searchable textual representation.

D. Character detection: The work adopted the use of OpenCV (open source computer vision) library to process the images so that features for each letter could be extracted. First, the frames were obtained continuously from the camera and send to the process. Once the object of interest is extracted from the camera image using a cascade classifier, a subsequent process can be done using the following steps.
Conversion to grayscale: The first stage in OCR involves generating a black-and-white version of the color or grayscale scanned page. The colour of the image is detected at the colour detection phase to convert the RGB colour images into the greyscale colour images. OCR is essentially a binary process: it recognizes things that are either there or not. If the original scanned image is perfect, any black it contains was part of a character that needs to be recognized while any white was part of the background. Reducing the image to black and white is hence the first stage in figuring out the text that is to be processed. This conversion may also introduce some errors.
OCR: All OCR programs are slightly different, but generally they process the image of each page by recognizing the text character by character, word by word, and line by line
Basic error correction: OCR programs have error-checking features to help you spot mistakes. For example, it uses a method called near-neighbor analysis to find words that are likely to occur nearby, so text incorrectly recognized as "the barking bog" might be automatically changed to "the barking dog".
Layout analysis: Good OCR programs automatically detect complex page layouts, such as multiple columns of text, tables, images, and so on. Images are automatically turned into graphics, tables are turned into tables, and columns are split up correctly, so the text from the first line of the first column isn't automatically joined to the text from the first line of the second column.
Proofreading: Even the best OCR programs aren't that perfect, especially when they're working from very old documents or poor-quality printed text. That's why the final stage in OCR should always be a good, old-fashioned human proofread.
Finite State Automaton: This work employed direct finite state automata in analyzing and capturing the set of recognized characters or alphabets and group them so that the resultant group represents a valid consonant-vowel combination. This is achieved with the help of a Finite State Automaton. A finite automaton is a finite-state machine that accepts or rejects strings of symbols or characters and only produces a unique computation of the automaton for each input string.

Mathematically, a *deterministic finite automaton* is a quintuple (Five-Tuple) consisting of

$M\,(K, \Sigma, \delta, sF)$ where,
K is a finite set of *states,*
Σ is an alphabet,
$s \in K$ is the *initial state*
$F \subseteq K$ is the set of *final states,* and
δ, the *transition function, is* a function from (K x E) to K

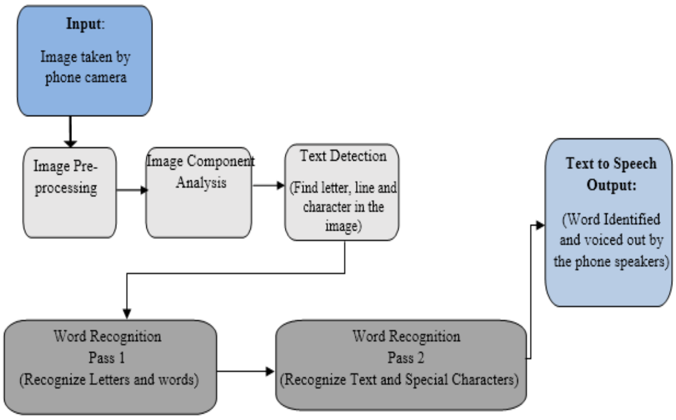

Figure 2. Design Architecture

From the system design architecture in figure 2, the process of detecting and recognizing text is divided into text detection stage and recognition stage. Text detection deals with finding text area from an input image, whereas recognition deals with converting obtained text into characters and words. The stepwise method is used for this purpose. Stepwise methods have separate stages of detection and recognition and they proceed through detection, classification, segmentation, and recognition. The image of the text is captured with the camera of the phone. This image is then taken through various processing stages.

Text detection and localization: Text detection deals with detecting the presence of the text in the input image whereas text localization localizes the position of the text and forms groups of text regions by eliminating the maximum of the background. Text detection and localization process are performed using connected component analysis or region-based methods.

The connected component (cc) analysis method forms a graph of connected points based on color or edge features from the binarized image.

During the preprocessing stage, the text image is divided into small regions using windows and search these regions for the presence of text using texture or morphological operations since text and non-text regions have different textual properties.

After the text detection and localization stage output may contain non-text regions along with text regions as false positives. The classification stage verifies text regions and eliminates non-text regions using classification algorithms. This stage can also be called verification. A supervised classification that knows properties of text such as color, size, texture, etc. before classification is applied. Unsupervised algorithms do not have prior knowledge about text features.

The segmentation process is used to separate text from the background and to extract bounded text from the image. The stepwise method used here undergoes segmentation to obtain precisely extracted characters that are fed to the recognition stage.

Character segmentation: Character segmentation is the process of converting text into multiple sets of single characters. It is suitable in the case of degraded text of connected characters. Gradient vector flow-based method is used which is applied directly on grayscale images. It initially identifies candidate cut pixels from the characters and then uses a two-pass pathfinding process that finds out potential cuts in forwarding pass and verify true cuts and removes false cuts in the backward pass.

Text Recognition: The text recognition stage converts images of text into a string of characters or words. It is important to convert images of text into words as the word is an elementary entity used by human for his visual recognition.

The character recognition approach used in this design is achieved using the Optical Character Recognition module (OCR) where initially images are segmented into k classes followed by binary text image hypothesis generation which passes through connected components analysis and grayscale consistency constraint module before getting fed to OCR.

Word recognition: Word recognition uses character recognition outputs along with language models or lexicons to recognize words from text images.

E. Tesseract: Tesseract is Open source OCR engine (figure 3). Tesseract works with independently developed Page Layout Analysis Technology. Hence, Tesseract accepts the input image as a binary image. Tesseract can handle both, the traditional- Black on the White text and also inverse-white on the Black text. Outlines of the component are stored on connected Component Analysis. Nesting of outlines is done which gathers the outlines together to form a Blob. Such Blobs are organized into text lines. Text lines are analyzed for fixed pitch and proportional text. Then the lines are broken into words by analysis according to the character spacing. Fixed pitch is chopped in character cells and proportional text is broken into words by definite spaces and fussy spaces.

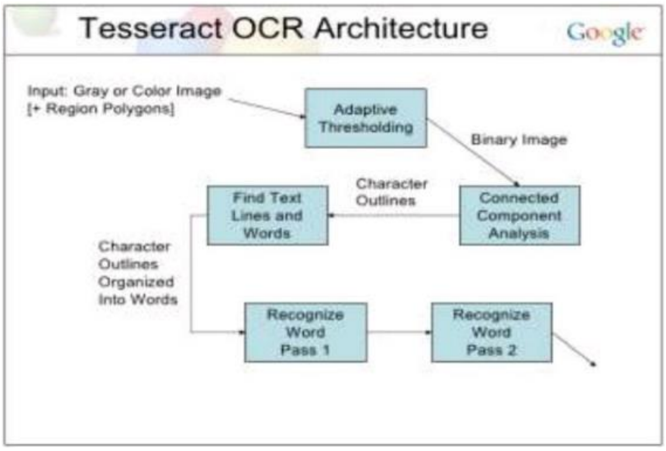

Figure 3. Tesseract Architecture (Athira et al, 2016, Adopted)

Tesseract performs activities to recognize words. This recognition activity mainly consists of two passes. The first pass tries to recognize the words. Then satisfactory word is passed to the Adaptive Classifier as training data, which recognizes the text more accurately. During the second pass, the words which were not recognized well in the first pass are recognized again through run over the page. Finally, Tesseract resolves fuzzy spaces. To locate small and capital text Tesseract checks alternative hypotheses for x-height.

F. Text to Speech: A Text-To-Speech (TTS) synthesizer is a computer-based algorithm that can 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. There is a fundamental difference between this work and any other talking machine (as a cassette-player for example) in the sense that we are interested in the automatic production of new sentences. Systems that just concatenate isolated words or parts of sentences, denoted as Voice Response Systems, are only applicable when a limited vocabulary is required, and when the sentences to be pronounced have a very restricted structure. In the case of TTS synthesis, it is impossible to record and store all the words of the language. It is thus more suitable to define Text-To-Speech as the automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter. TTS can be a help to a handicapped person who is unable to speak. With the help of an especially designed keyboard and a fast sentence assembling the program, synthetic speech can be produced in a few seconds. Astro-physician Stephen Hawking gives all his lectures in this way. Blind people also widely benefit from TTS systems, when coupled with Optical Recognition Systems (OCR), which give them access to written information. The market for speech synthesis for blind users of personal computers will soon be invaded by mass-market

A Text-To-Speech (TTS) synthesizer is a computer-based algorithm that can 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. There is a fundamental difference between this work and any other talking machine (as a cassette-player for example) in the sense that the work is interested in the automatic production of new sentences. Systems that just concatenate isolated words or parts of sentences, denoted as Voice Response Systems, are only applicable when a limited vocabulary is required, and when the sentences to be pronounced have a very restricted structure. In the case of TTS (figure 4) synthesis, it is impossible to record and store all the words of the language. It is thus more suitable to define Text-To-Speech as the automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter. TTS can be a help to a handicapped person who is unable to speak. With the help of an especially designed keyboard and a fast sentence assembling the program, synthetic speech can be produced in a few seconds. Astro-physician Stephen Hawking gives all his lectures in this way. Blind people also widely benefit from TTS systems, when coupled with Optical Recognition Systems (OCR), which give them access to written information. The market for speech synthesis for blind users of personal computers will soon be invaded by mass-market synthesizers bundled with sound cards. DEC talk (TM) is already available with the latest SoundBlaster (TM) cards now, although not yet in a form useful for blind people.

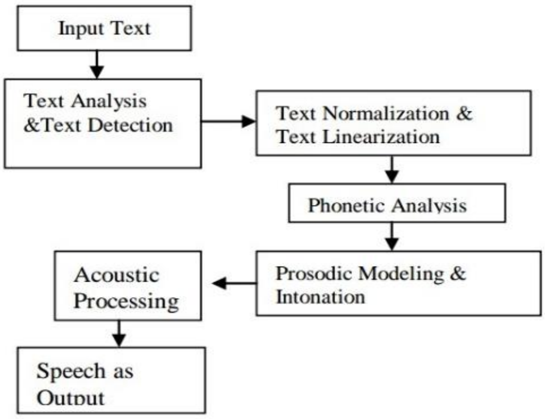

Figure 4. Block diagram of TTS

3.3 Algorithm of Design

Algorithm for OCR:
1. Start
2. Take input image.
3. Check if RGB image and convert to grayscale image.
4. Create black & white images using “graythresh” & “im2bw”.
5. Filter image and remove pixels/objects less than 30 pixels.
6. Load template character.
7. Separate lines using the “line” function.
8. Separate letters using “bwlabel”.
9. Recognize letter using read letter & write into text.
10. Check if a line is finished then move to the next line.
11. Check if an image is finished.
12 .Exit.

IV. SYSTEM IMPLEMENTATION AND TESTING

4.1 Output Specification and Design

Text recognition is performed by off-the-shelf OCR before the output of informative words from the localized text regions. A text region labels the minimum rectangular area for the accommodation of characters inside it, so the border of the text region contacts the edge boundary of the text character. However, our experiments show that OCR generates better performance if text regions are first assigned proper margin areas and binarized to segment text characters from the background. Thus each localized text region is enlarged by enhancing the height and width by 10 pixels respectively, and then Otsu’s method was used to perform binarization of text regions, where margin areas are always considered as background. We test both open- and closed-source solutions exist that have APIs that allow the final stage of conversion to letter codes (e.g. OmniPage, Tesseract, ABBYReader). The recognized text codes are recorded in script files. Then the method employed the Google text-to-speech to load these files and display the audio output of text information. Blind users can adjust speech rate, volume, and tone according to their preferences

4.2 System Implementation

This chapter captures the implementation of the proposed system. The implementation of the proposed system was accomplished using Android SDK and Open CV library. The reason for choosing this is because android is an open-source mobile operating system with a massive user base and a simplified mobile app development process.
During the implementation phase, the final system was tested to ensure it meets user requirements and the research objective as stated in chapter one.

Android is an open-source and Linux-based operating system targeted for mobile devices such as smart-phones and tablet computers. Applications are generally developed in Java programming language using the Android software development kit (SDK). If used correctly, the SDK, together with Eclipse (the officially supported IDE) and JDK (Java Development Kit) is capable to deliver modern software for Android devices. The Android SDK (software development kit) provides the API (Application programming interface) libraries and developer tools essential to build, test, and debug apps for Android. This application is created in eclipse. Android Development Tools (ADT) is a plug-in for the Eclipse integrated development environment (IDE) that is designed to provide an integrated environment for Android applications.

4.3 System Requirement

Generally, the requirement of a system specifies the set of hardware and software that was expected at a minimum level for the application to function as expected. The proposed system will require the following phone requirement at a minimum to work.

4.4 Phone Requirement

i Android version 7.0 or higher
ii RAM 2GB or higher
iii A very good camera with high resolution

4.5 Experimental Result

Results are shown in the following Screenshots.

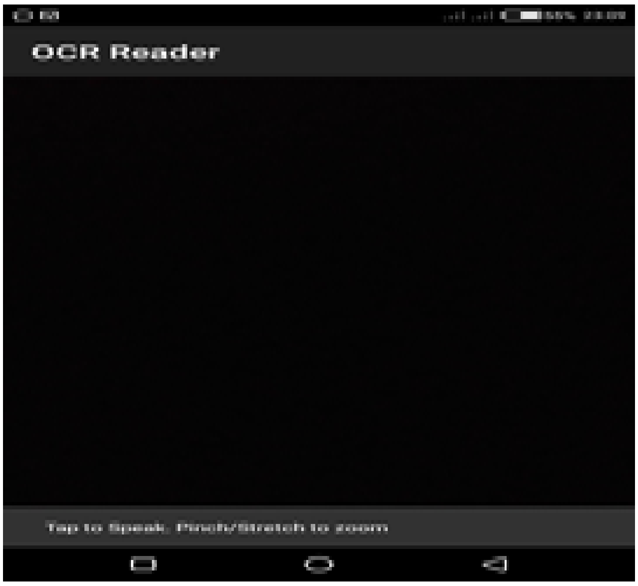

Figure 5. Main GUI Layout

Figure 5 shows the main screen of the application on a mobile phone.

Figure 6. Captured image from camera

The first step is capturing the image and double-clicking on the screen to produce the audio-output.

4.6 Technology Used

*Android***:** Android is an operating system (OS) designed basically for touchscreen mobile phones. It is based on the Linux kernel and is currently being developed by Google. Android's user interface allows direct manipulation, using touch gestures, swiping, tapping, and pinching, to manipulate objects on the screen, a virtual keyboard for textual input.

Applications that are more popularly known as "apps", extend the functionality of devices. They are written using the Android SDK (software development kit) and mostly use the Java programming language which provides complete access to the Android APIs.

*Eclipse***:** Eclipse is an integrated development environment (IDE) that contains a base workspace and an extensible plug-in system that is used for customizing the environment. Eclipse is mostly written in Java and hence its primary use is for developing Java applications. To develop applications in other programming languages plugins are required. The toolkit of Java, called SWT, has graphical control elements that are implemented by Eclipse. It is seen that most Java applications make use of the Java standard AWT (Abstract Window Toolkit) or Swing. To provide an integrated environment to build Android applications a Google-provided plugin called ADT (Android Development Tools) for the Eclipse IDE was used. It helps the developers to create user interfaces, add Android Framework API-based packages, debugging options using SDK tools, and in exporting signed or unsigned .apk files of applications to be used by users. ADT is freeware.

V. EVALUATION

The performance of the system was evaluated using a confusion matrix. Hundred (100) items were considered in the system. Three (3) blind volunteers were engaged in the use of the system (although, it can also be tested by non-visually impaired). In all cases, only three items were wrongly identified by the system as shown in table 1. This is due to wrong labeling.

Table I. Percentage Distribution of Observations

	TP	***FP***	***TN***	***FN***
%Distributions	97	3	97	3

The use of the system for the identification of objects becomes a very critical issue when it gives results of cases showing false positive (FP) and false-negative (FN). It is a sensitive issue to identify an object wrongly. The system is identified as True Positive (TP) if the system is true and it is true when tested, then the system is True Negative (TN) if the system is true but it is false when tested.

Thus, precision, recall, and the F-measure were considered for the performance evaluation metric. The above figure 6 shows the ca

i. Precision: The total number of correctly identified true object cases out of retrieved data when tested.

Precision, P = TP/(TP + P)=97/(97+3)=97/100=0.97=97%

ii. Recall: The number of correctly identified object cases from the total number of true object identification cases

Recall, R = TP/(TP + FN)=97/(97+3)=97/100=0.97=97%

iii. F-measure: The equally weighted harmonic mean of precision and recall. The model presents the identification performance on the average value

F-measure = 2/(1/R + 1/P)=2/(1/0.97 + 1/0.97)=0.97=97%

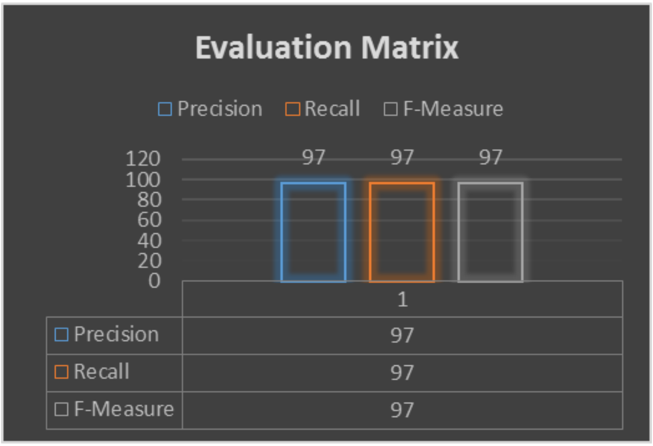

Figure 7. Evaluation Matrics

The evaluation metrics graph is shown in figure 7.

VI. DISCUSSION

In this paper, a prototype system that read printed text on hand-held objects for assisting blind persons was developed. To solve the common problem for blind users, the work proposed a motion-based technique to detect the object of interest. This method can effectively distinguish the object of interest from the background or other objects in the camera vision. To extract text regions from complex backgrounds, an "OCR Algorithm" text localization algorithm the corresponding feature maps estimate the global structural feature of text at every pixel was proposed. Block patterns project the proposed feature maps of an image patch into a feature vector. Adjacent character grouping is performed to calculate candidates of text patches prepared for text classification. An Adaboost learning model is applied to localize text in camera-based images. Off-the-shelf OCR is used to perform word recognition on the localized text regions and transform it into audio output for blind users.

VII. CONCLUSION

A system for assisting the blind via a software application based on the Android platform has been developed. It will help ease the difficulty of performing the task of object identification. For images without definite edges, the program may not work properly. But it will work perfectly for image texts which have a prominent edge. For the product with a fancy font, transparent text, text that is too small, blurred text, and for the non-planar surface it will not work properly. The labeling algorithm needs to be improved. A better labeling method of components could improve the detection of characters. This could get better results for circular text, which tends to be dismissed as the noise due to the grouping of the letters. To solve the common aiming problem for blind users, a have proposed a camera-based product information reading framework to help blind persons read product information from hand-held objects in their daily lives. In this project, the output is in the form of audio. The system records 97% accuracy.

VIII. REFERENCES

1. WHO (2020): Blindness and Visual Impairment. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment
2. Strumillo, Pawel. (2012). Electronic Systems Aiding Spatial Orientation and Mobility of the Visually Impaired. 98. 373-386. 10.1007/978-3-642-23187-2_24.
3. Gill, J., & Jolliff, L. (2008). People with Visual Disabilities. The Engineering Handbook of Smart Technology for Aging, Disability, and Independence, 143.
4. Onishi, J., & Ono, T. (2011). Contour pattern recognition through auditory labels of Freeman chain codes for people with visual impairments. In 2011 IEEE International Conference on Systems, Man, and Cybernetics (pp. 1088-1093). IEEE.
5. Kornsingha, T., & Punyathep, P. (2011, May). A voice system, reading medicament label for visually impaired people. In RFID SysTech 2011 7th European Workshop on Smart Objects: Systems, Technologies, and Applications (pp. 1-6). VDE.
6. Hakobyan, L., Lumsden, J., O'Sullivan, D., & Bartlett, H. (2013). Mobile assistive technologies for the visually impaired. Survey of ophthalmology, 58(6), 513-528.
7. Google Goggles (www.google.com/mobile/goggles)
8. Matusiak, K., Skulimowski, P., & Strurniłło, P. (2013, June). Object recognition in a mobile phone application for visually impaired users. In 2013 6th International Conference on Human System Interactions (HSI) (pp. 479-484). IEEE.
9. Varpe, K. M., & Wankhade, M. P. (2013). Visually impaired assistive system. International Journal of Computer Applications, 77(16).
10. Mathankumar, M., & Sugandhi, N. (2013, August). A low-cost smart shopping facilitator for the visually impaired. In 2013 International Conference on Advances in Computing, Communications, and Informatics (ICACCI) (pp. 1088-1092). IEEE.
11. Rafian, P., & Legge, G. E. (2017). Remote sighted assistants for indoor location sensing of visually impaired pedestrians. ACM Transactions on Applied Perception (TAP), 14(3), 1-14.