SMART ASSISTANT FOR BLIND PEOPLE USING RASPBERRY PI 3

[1]A.Manonmani, [2]M.Arivalagan, [3]M.Lavanya, [4]S.Sellakumar

ABSTRACT

An Optical Character Recognition (OCR) system which is a branch of computer vision and in turn a sub-class of Artificial Intelligence. Optical character recognition is the translation of optically scanned bitmaps of printed or hand-written text into audio output by using of Raspberry pi. OCRs developed for many world languages are already under efficient use. This method extracts moving object region by a mixture-of-Gaussians-based background subtraction method. A text localization and recognition are conducted to acquire text information. To automatically localize the text regions from the object, a text localization and tesseract algorithm by learning gradient features of stroke orientations and distributions of edge pixels in an Ada boost model. Text characters in the localized text regions are then binaries and recognized by off-the-shelf optical character recognition software. The recognized text codes are output to blind users in speech. Performance of the proposed text localization algorithm. As the recognition process is completed, the character codes in the text file are processed using Raspberry pi device on which recognize character using tesseract algorithm and python programming, the audio output is listed.

Keywords: *Optical Character Recognition (OCR), International Telecommunication Union (IoT), computed tomography (CT).*

I. INTRODUCTION

The Future Internet goal is to provide an infrastructure to have an immediate access to information about the physical world and its objects. Physical objects can be applicable to different application domains, such as e-health, warehouse management, etc. Each application domain may have different types of physical devices.

Each physical device can have its own specifications, which is required to use in order to interact with it. To achieve the future Internet goal, a layered vision is required that can facilitate data access. Internet of Things (IOT) is a vision that aims to integrate the virtual world of information to the real world of devices through a layered architecture.

[1] Assistant Professor, Department of Electronics and Instrumentation Engineering, Saveetha Engineering College, Chennai.
[2] Assistant Professor, Department of Electronics and Instrumentation Engineering, Saveetha Engineering College, Chennai.
[3] Assistant Professor, Electrical and Electronics Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, India.
[4] Assistant Professor, Mechanical Engineering, Saveetha Engineering College, Chennai, India.

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

The term “Internet of Things” consists of two words, namely Internet and Things. Internet refers to the global network infrastructure with scalable, configurable capabilities based on interoperable and standard communication protocols. Things are physical objects or devices, or virtual objects, devices or information, which have identities, physical attributes and virtual personalities, and use intelligent interfaces. For instance, a virtual object can represent an abstract unit of sensor nodes that contains metadata to identify and discover its corresponding sensor nodes. Therefore, IoT refers to the things that can provide information from the physical environment through the Internet.

Middleware is as an interface between the hardware layer and the application layer, which is responsible for interacting with devices and information management. The role of a middleware is to present a unified programming model to interact with devices. A middleware is in charge of masking the heterogeneity and distribution problems that we face when interacting with devices.

As of 2016, the vision of the Internet of things has evolved due to a convergence of multiple technologies, including ubiquitous wireless communication, real-time analytics, machine learning, commodity sensors, and embedded systems. This means that the traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things..

The concept of the Internet of things became popular in 1999, through the Auto-ID Center at MIT and related market-analysis publications. A significant transformation is to extend "things" from the data generated from devices to objects in the physical space. The thought-model for future interconnection environment was proposed in 2004. This thought model envisioned the development trend of the Internet of things.

1.1 IoT DEFINITION

The layered architecture for IoT .Internet of Things (IoT) has increasingly gained attention in industry to interact with different types of devices. IoT can have influence on industry and society by integrating physical devices into information network. To identify devices, we can use identification technologies like for example RFID, which allow each device be uniquely identified. International Telecommunication Union defines IoT as “A global infrastructure for the Information Society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies.

Internet connected devices are extensive. Multiple categorizations have been suggested, most of which agree on a separation between consumer, enterprise (business), and infrastructure applications. George Osborne, the former British Chancellor of the Exchequer, posited that the Internet of things is the next stage of the information revolution and referenced the inter-connectivity of everything from urban transport to medical devices to household appliances. The ability to network embedded devices with limited CPU, memory and power resources means that IoT finds applications in nearly every field. Such systems could be in charge of collecting information in settings ranging from natural ecosystems to buildings and factories, thereby finding applications in fields of environmental sensing and urban planning.

II. EXISTING SYSTEM

In existing approach, it is a method to design a Text to Speech conversion module by the use of Mat lab by simple matrix operations. Firstly, by the use of microphone some similar sounding words are recorded using a record program in the Mat lab window and recorded sounds are saved in. wave format in the directory. The recorded sounds are then sampled and the sampled values are taken and separated into their constituent phonetics. The separated syllables are then concatenated to reconstruct the desired words. By the use of various Mat lab commands i.e. wave read, subplot etc. the waves are sampled and extracted to get the desired result. This method is simple to implement and involves much lesser use of memory spaces.

We develop a framework for reconstructing images that are sparse in an appropriate transform domain from polychromatic computed tomography (CT) measurements under the blind scenario where the material of the inspected object and incident-energy spectrum are unknown. The existing navigation systems for the blind people require a precise GPS maps. This make them unusable in region where there are no GPS maps, they are not sufficiently accurate. The natural voice navigation, adaptive to the velocity and accuracy of the GPS data, start of the navigation from any waypoint, correlation of the direction of movement if it is necessary, return the user to the route if deviation is deviated, work with and without electronic compass, detection of the movement of the user in the opposite direction.

2.1 PROPOSED SYSTEM

In proposed system we have described a prototype system to read printed text on hand-held objects for assisting blind persons. In order to solve the common aiming problem for blind users, we have proposed a motion-based method to detect the object of interest, while the blind user simply shakes the object for a couple of seconds. The automatic ROI detection and text localization algorithms were independently evaluated as unit tests to ensure effectiveness and robustness of the whole system. We subsequently evaluated this prototype system of assistive text reading using images of hand-held objects captured by ten blind users in person. Two calibrations were applied to prepare for the system test. First, we instructed blind users to place hand-held object within the camera view. Since it is difficult for blind users to aim their held objects, we employed a camera with a reasonably wide angle. In future systems, we will add finger point detection and tracking to adaptively instruct blind users to aim the object. Second, in an applicable blind-assistive system, a text localization algorithm might prefer higher recall by sacrificing some precision. By using ultrasonic sensor, we will measure the distance between the blind people and obstacle then the distance will be played through ear phones.

2.2 SYSTEM REQUIREMENT

The hardware system requirements are Processor-Intel Pentium, Speed-1.1GHz, RAM- 2Gb(min), Hard Disk-1.5, Input -Webcam, Ultra sonic, Output - Headset, Buzzer. The software requirements are Operating System: Raspberry and Language Used: Python.

III. SYSTEM DESIGN & SYSTEM ARCHITECTURE

The system architecture is shown in Figure 1.

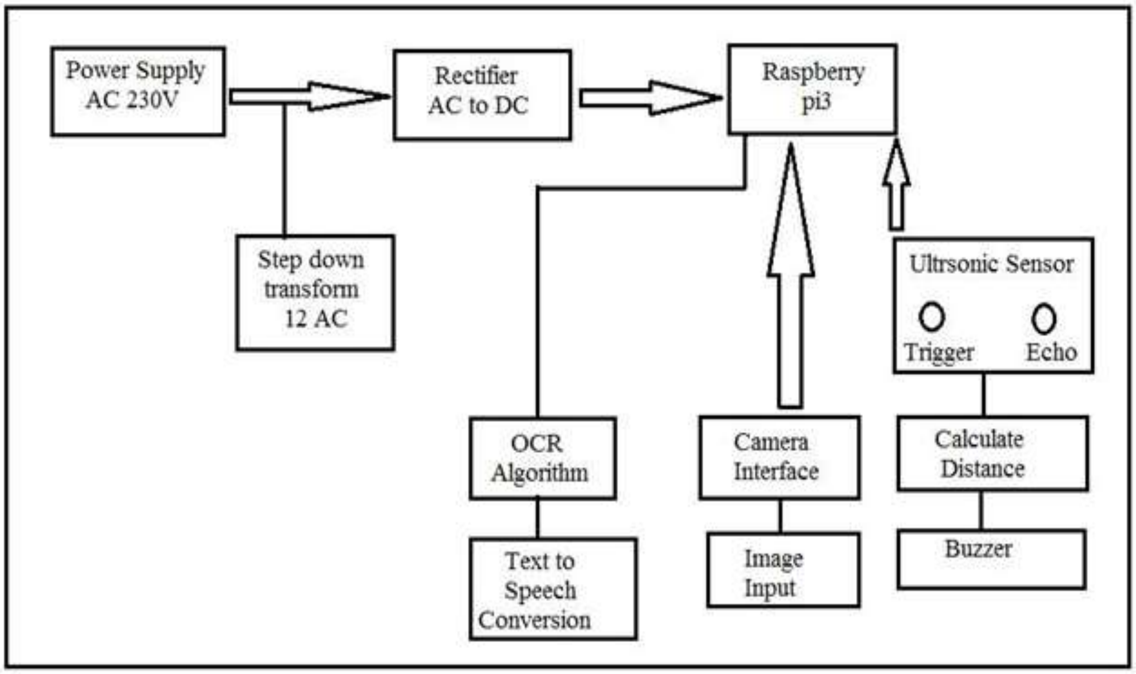

Figure 1 System Architecture

A prototype system to read printed text on hand-held objects for assisting blind persons. In order to solve the common aiming problem for blind users, we have proposed a motion-based method to detect the object of interest, while the blind user simply shakes the object for a couple of seconds. The automatic ROI detection and text localization algorithms were independently evaluated as unit tests to ensure effectiveness and robustness of the whole system.

When capture button is clicked this system captures the product image placed in front of the web camera which is connected to ARM microcontroller through USB After selecting the process button the captured label image undergoes Optical Character Recognition(OCR) Technology. OCR technology allows the conversion of scanned images of printed text or symbols (such as a page from a book) into text or information that can be understood or edited using a computer program. The most familiar example is the ability to scan a paper document into a computer where it can then be edited in popular word processors such as Microsoft Word. However, there are many other uses for OCR technology, including as a component of larger systems which require recognition capability, such as the number plate recognition systems, or as tools involved in creating resources for SALT development from print based texts.

3.1 MODULES

The modules used in the system are Image capturing and pre-processing. Automatic text extraction, Text recognition and audio output. We have described a prototype system to read printed text on hand-held objects for assisting blind persons. In order to solve the common aiming problem for blind users, we have proposed a motion-based method to detect the object of interest, while the blind user simply shakes the object for a couple of seconds. The automatic ROI detection and text localization algorithms were independently evaluated as unit tests to ensure effectiveness and robustness of the whole system. The image capture is shown in Figure 2.

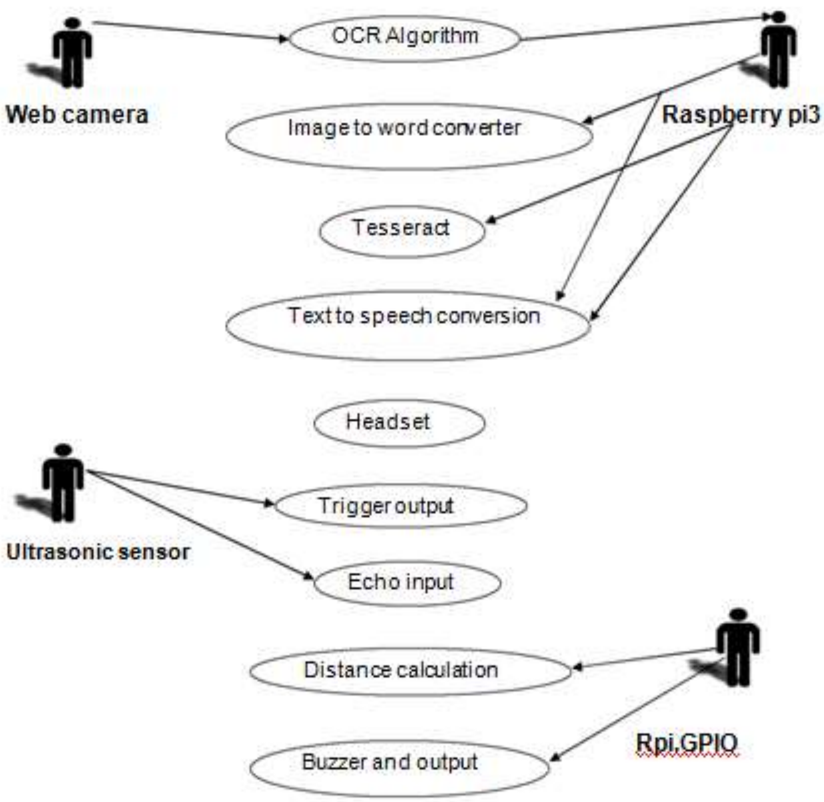

Figure 2 Image Capture and Pre-processing

When capture button is clicked this system captures the product image placed in front of the web camera which is connected to ARM microcontroller through USB After selecting the process button the captured label image undergoes Optical Character Recognition(OCR) Technology. OCR technology allows the conversion of scanned images of printed text or symbols (such as a page from a book) into text or information that can be understood or edited using a computer program. The most familiar example is the ability to scan a paper document into a computer where it can then be edited in popular word processors such as Microsoft Word. However, there are many other uses for OCR technology, including as a component of larger systems which require recognition capability, such as the number plate recognition systems, or as tools involved in creating resources for SALT development from print based texts.

3.1.1 IMAGE CAPTURING AND PRE-PROCESSING

The video is captured by using web-cam and the frames from the video is segregated and undergone to the pre-processing. First, get the objects continuously from the camera and adapted to process. Once the object of interest is extracted from the camera image and it converted into gray image. Use hears cascade classifier for recognizing the character from the object. The work with a cascade classifier includes two major stages: training and detection. For training need a set of samples. There are two types of samples: positive and negative. To extract the hand-held object of interest from other objects in the camera view, ask users to shake the hand-held objects containing the text they wish to identify and then employ a motion-based method to localize objects from cluttered background.

In this module the image is capture by the web camera based on using lens, pixels, and frames by +5v ().The frame from the image is segregated and it undergoes to pre-processor with the spatial domain and frequency domain by brightness (), contrast (). Power supply is also used to capture image with the help VCC, AC (230) with function of bridge rectifier (), voltage regulator (). The Sequence Diagram for Image Capturing and Pre-Processing Module is shown in Figure 3.

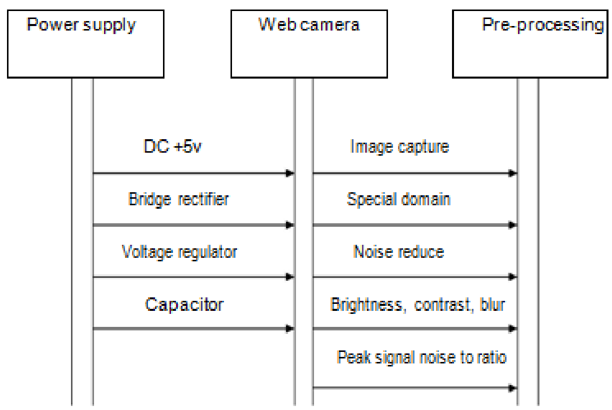

Figure 3 Sequence Diagram for Image Capturing and Pre-Processing Module

In this module, the power supply flows direct current to the webcam by +5v,the bridge rectifier are used convert alternating current input to direct current output to the web camera, voltage are used to regulate one or more AC or DC voltages. Webcam are used to capture the segregated image from pre-processing. If the image captured by webcam has brightness, contrast, blur is stored by pre-processor. It also reduces the noise captured by webcam. Therefore, it is a PNR. A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. A sequence diagram shows how object interaction arranged in time sequence.

3.1.2 AUTOMATIC TEXT EXTRACTION

In order to handle complex backgrounds, two novel feature maps to extracts text features based on stroke orientations and edge distributions, respectively. Here, stroke is defined as a uniform region with bounded width and significant extent. These feature maps are combined to build an Adaboost based text classifier. The extraction information from audio and image source restricted to information execution from text. The actual transduction of audio and image data into text is the processing of OCR output.

3.1.3 TEXT RECOGNITION AND AUDIO OUTPUT

Text recognition is performed by off-the-shelf OCR prior to 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 characters. However, this experiment show that OCR generates better performance text regions are first assigned proper margin areas and binaries to segments text characters from background.

IV. SERIAL COMMUNICATION

Serial communication is basically the transmission or reception of data one bit at a time. Today's computers generally address data in bytes or some multiple thereof. A byte contains 8 bits. A bit is basically either a logical 1 or zero. Every character on this page is actually expressed internally as one byte. The serial port is used to convert each byte to a stream of ones and zeroes as well as to convert a stream of ones and zeroes to bytes. The serial port contains an electronic chip called a Universal Asynchronous Receiver/Transmitter (UART) that actually does the conversion. The serial port has many pins. We will discuss the transmit and receive pin first. Electrically speaking, whenever the serial port sends a logical one (1) a negative voltage is effected on the transmit pin. Whenever the serial port sends a logical zero (0) a positive voltage is affected.

When no data is being sent, the serial port's transmit pin's voltage is negative (1) and is said to be in a MARK state. Note that the serial port can also be forced to keep the transmit pin at a positive voltage (0) and is said to be the SPACE or BREAK state. (The terms MARK and SPACE are also used to simply denote a negative voltage (1) or a positive voltage (0) at the transmit pin respectively). When transmitting a byte, the UART (serial port) first sends a STARTBIT which is a positive voltage (0), followed by the data (general 8 bits, but could be 5, 6, 7, or 8 bits) followed by one or two STOP Bits which is a negative (1) voltage. The sequence is repeated for each byte sent. Figure 1 shows a diagram of what a byte transmission would look like.

At this point you may want to know what the duration of a bit is. In other words, how long does the signal stay in a particular state to define a bit. The answer is simple. It is dependent on the baud rate. The baud rate is the number of times the signal can switch states in one second. Therefore, if the line is operating at 9600 baud, the line can switch states 9,600 times per second. This means each bit has the duration of 1/9600 of a second or about 100µsec.

LSB (0 1 0 0 0 0 0 1 0 1) MSB

The above represents (Start Bit) (Data Bits) (Stop Bit). To calculate the actual byte, transfer rate simply divide the baud rate by the number of bits that must be transferred for each byte of data. In the case of the above example, each character requires 10 bits to be transmitted for each character. As such, at 9600 bauds, up to 960 bytes can be transferred in one second. These signals are the Carrier Detect Signal (CD), asserted by modems to signal a successful connection to another modem, Ring Indicator (RI), asserted by modems to signal the phone ringing, Data Set Ready (DSR), asserted by modems to show their presence, Clear To Send (CTS), asserted by modems if they can receive data, Data Terminal Ready (DTR), asserted by terminals to show their presence, Request To Send (RTS), asserted by terminals if they can receive data. The section RS232 Cabling describes these signals and how they are connected. It re-asserts it when it can receive again. A terminal does the same thing instead with the RTS signal. Another method of hardware flow control in practice is to perform the same procedure in the previous paragraph except that the DSR and DTR signal.

4.1 NULL MODEM

Serial communications with RS232. One of the oldest and most widely spread communication methods in computer world. The way this type of communication can be performed is pretty well defined in standards. I.e. with one exception. The standards show the use of DTE/DCE communication, the way a computer should communicate with a peripheral device like a modem. For your information, DTE means Data Terminal Equipment (computers etc.) where DCE is the abbreviation of Data Communication Equipment (modems).

One of the main uses of serial communication today where no modem is involved—a serial null modem configuration with DTE/DTE communication—is not so well defined, especially when it comes to flow control. The terminology null modem for the situation where two computers communicate directly is so often used nowadays, that most people don't realize anymore the origin of the phrase and that a null modem connection is an exception, not the rule.

4.2 RS232

When we look at the connector pin out of the RS232 port, we see two pins which are certainly used for flow control. These two pins are RTS**,** request to send and CTS, clear to send. With DTE/DCE communication (i.e. a computer communicating with a modem device) RTS is an output on the DTE and input on the DCE**.** CTS are the answering signal coming from the DCE**.** Before sending a character, the DTE asks permission by setting its RTS output. No information will be sent until the DCE grants permission by using the CTS line. If the DCE cannot handle new requests, the CTS signal will go low. A simple but useful mechanism allowing flow control in one direction. The assumption is that the DTE can always handle incoming information faster than the DCE can send it. In the past, this was true. Modem speeds of 300 baud were common and 1200 baud was seen as a high speed connection.

The last flow control signal present in DTE/DCE communication is the CD carrier detect. It is not used directly for flow control, but mainly an indication of the ability of the modem device to communicate with its counterpart. This signal indicates the existence of a communication link between two modem devices.

4.3 NULL MODEM WITHOUT HANDSHAKING

How to use the handshaking lines in a null modem configuration? The simplest way is to don't use them at all. In that situation, only the data lines and signal ground are cross connected in the null modem communication cable. All other pins have no connection. An example of such a null modem cable without handshaking can be seen in the Table 1.

Table 1 Simple null modem without handshaking

Connector 1	Connector 2	Function	
2	3	RX	TX
3	2	TX	RX
5	5	Signal ground	

4.4 COMPATIBILITY ISSUES

There is a problem, if either of the two devices checks the DSR or CD inputs. These signals normally define the ability of the other side to communicate. As they are not connected, their signal level will never go high. This might cause a problem. The same holds for the RTS/CTS handshaking sequence. If the software on both sides is well structured, the RTS output is set high and then a waiting cycle is started until a ready signal is received on the CTS line. This causes the software to hang because no physical connection is present to either CTS line to make this possible. The only type of communication which is allowed on such a null modem line is data-only traffic on the cross connected Rx/TX lines.

This does however not mean that this null modem cable is useless. Communication links like present in the Norton Commander program can use this null modem cable. This null modem cable can also be used when communicating with devices which do not have modem control signals like electronic measuring equipment etc.

4.5 UART

A Universal Asynchronous Receiver/Transmitter, abbreviated UART is a piece of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as EIA, RS-232, RS-422 or RS-485. The *universal* designation indicates that the data format and transmission speeds are configurable. The electric signaling levels and methods (such as differential signaling etc.) are handled by a driver circuit external to the UART. A UART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port.

UARTs are now commonly included in microcontrollers. A dual UART, or DUART, combines two UARTs into a single chip. An octal UART or OCTART combines eight UARTs into one package, an example being the NXP SCC2698. Many modern ICs now come with a UART that can also communicate synchronously; these devices are called USARTs (Universal Synchronous/Asynchronous Receiver/Transmitter).

4.6 CHARACTER FRAMING

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

The right-most (least significant) data bit is always transmitted first. If parity is present, the parity bit comes after the data bits but before the stop bit(s). The character framing is shown in Table 2.

Table 2 Character Framing

Bit number	1	2	3	4	5	6	7	8	9	10	11
	Start				5–8 data bits						Stop
	bit										bit(s)
	Start	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Stop	

The idle, no data state is high-voltage, or powered. This is a historic legacy from telegraphy, in which the line is held high to show that the line and transmitter are not damaged. Each character is sent as a logic low start bit, a configurable number of data bits (usually 8, but users can choose 5 to 8 or 9 bits depending on which UART is in use), an optional parity bit if the number of bits per character chosen is not 9 bits, and one or more logic high stop bits.

The start bit signals the receiver that a new character is coming. The next five to nine bits, depending on the code set employed, represent the character. If a parity bit is used, it would be placed after all of the data bits. The next one or two bits are always in the mark (logic high, i.e., '1') condition and called the stop bit(s).

4.7 RECEIVER

All operations of the UART hardware are controlled by a clock signal which runs at a multiple of the data rate, typically 8 times the bit rate. The receiver tests the state of the incoming signal on each clock pulse, looking for the beginning of the start bit. If the apparent start bit lasts at least one-half of the bit time, it is valid and signals the start of a new character. If not, it is considered a spurious pulse and is ignored. After waiting a further bit time, the state of the line is again sampled and the resulting level clocked into a shift register. After the required number of bit periods for the character length (5 to 8 bits, typically) have elapsed, the contents of the shift register are made available (in parallel fashion) to the receiving system. The UART will set a flag indicating new data is available, and may also generate a processor interrupt to request that the host processor transfers the received data.

Communicating UARTs usually have no shared timing system apart from the communication signal. Typically, UARTs resynchronize their internal clocks on each change of the data line that is not considered a spurious pulse. Obtaining timing information in this manner, they reliably receive when the transmitter is sending at a slightly different speed than it should. Simplistic UARTs do not do this, instead they resynchronize on the falling edge of the start bit only, and then read the center of each expected data bit, and this system works if the broadcast data rate is accurate enough to allow the stop bits to be sampled reliably.

4.8 TRANSMITTER

Transmission operation is simpler since it is under the control of the transmitting system. As soon as data is deposited in the shift register after completion of the previous character, the UART hardware generates a start bit, shifts the required number of data bits out to the line, generates and appends the parity bit (if used), and appends the stop bits. Since transmission of a single character may take a long time relative to CPU speeds, the UART will maintain a flag showing busy status so that the host system does not deposit a new character for transmission until the previous one has been completed; this may also be done with an interrupt. Since full-duplex operation requires characters to be sent and received at the same time, UARTs use two different shift registers for transmitted and received characters.

4.9 APPLICATION

Transmitting and receiving UARTs must be set for the same bit speed, character length, parity, and stop bits for proper operation. The receiving UART may detect some mismatched settings and set a "framing error" flag bit for the host system; in exceptional cases the receiving UART will produce an erratic stream of mutilated characters and transfer them to the host system. Typical serial ports used with personal computers connected to modems use eight data bits, no parity, and one stop bit; for this configuration the number of ASCII characters per second equals the bit rate divided by 10.

Some very low-cost home computers or embedded systems dispense with a UART and use the CPU to sample the state of an input port or directly manipulate an output port for data transmission. While very CPU-intensive (since the CPU timing is critical), the UART chip can thus be omitted, saving money and space. The technique is known as bit-banging.

4.10 SPECIAL RECEIVER CONDITIONS

4.10.1 OVERRUN ERROR

An "overrun error" occurs when the receiver cannot process the character that just came in before the next one arrives. Various devices have different amounts of buffer space to hold received characters. The CPU must service the UART in order to remove characters from the input buffer. If the CPU does not service the UART quickly enough and the buffer becomes full, an Overrun Error will occur, and incoming characters will be lost.

4.10.2 UNDERRUN ERROR

An "Underrun error" occurs when the UART transmitter has completed sending a character and the transmit buffer is empty. In asynchronous modes this is treated as an indication that no data remains to be transmitted, rather than an error, since additional stop bits can be appended. This error indication is commonly found in USARTs, since an underrun is more serious in synchronous systems.

4.10.3 FRAMING ERROR

A "framing error" occurs when the designated "start" and "stop" bits are not found. As the "start" bit is used to identify the beginning of an incoming character, it acts as a reference for the remaining bits. If the data line is not in the expected state (hi/lo) when the "stop" bit is expected, a Framing Error will occur.

4.10.4 PARITY ERROR

A Parity Error occurs when the parity of the number of 1 bits disagrees with that specified by the parity bit. Use of a parity bit is optional, so this error will only occur if parity-checking has been enabled.

4.10.5 BREAK CONDITION

A "break condition" occurs when the receiver input is at the "space" level for longer than some duration of time, typically, for more than a character time. This is not necessarily an error, but appears to the receiver as a character of all zero bits with a framing error.

V. RASPBERRY Pi

The raspberry Pi is shown in the Figure 4.

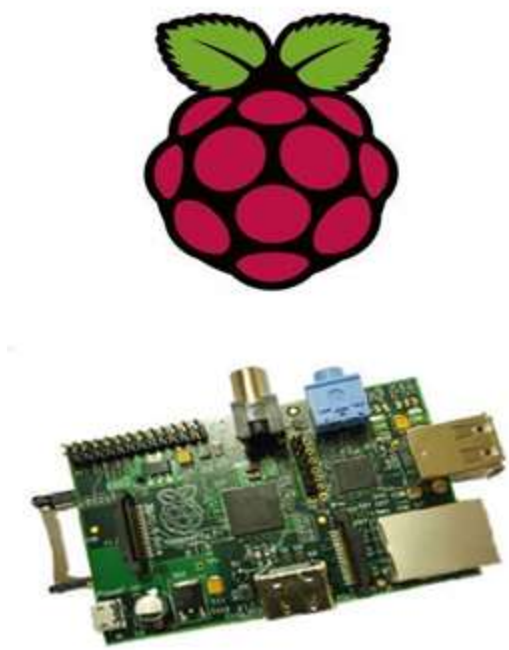

Figure 4 Raspberry Pi

5.1 RASPBERRY PI BASIC HARDWARE SETUP

The hardware setup of Raspberry Pi is shown in Figure 5.

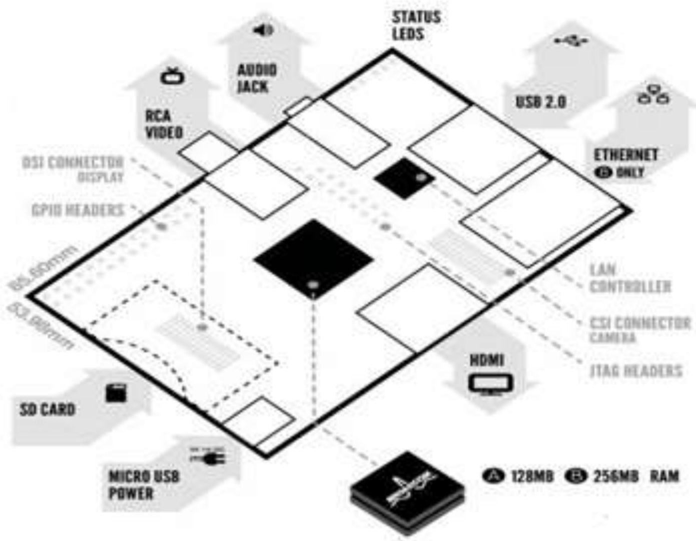

Figure 5 Hardware setup

VI. CONCLUSION

A text localization and recognition are conducted to acquire text information are observed. To automatically localize the text regions from the object, a text localization and tesseract algorithm by learning gradient features of stroke orientations and distributions of edge pixels in an Ada boost model are verified. Text characters in the localized text regions are then binaries and recognized by off-the-shelf optical character recognition software. The recognized text codes are output to blind users in speech. Performance of the text localization algorithm is dine. As the recognition process is completed, the character codes in the text file are processed using Raspberry pi device on which recognize character using tesseract algorithm and python programming, the audio output is listed.

REFERENCES

[1] AMD Alliance International, "The global economic cost of visual impairment.", 2010. Internet documenthttp://www.icoph.org/resources/146/The-Global-Economic-Cost- of-Visual-Impairment.html Last access Aug 22, 2015.

[2] L. Braille, "Nouveau procede pour representer des points la forme meme des letters, les cartes de geographie, les figures degeometrie, les caracteres de musiques, etc., a l'usage des aveugles," Institution royale des jeunes aveugles, 1839.

[3] V. Coskun, K. Ok, B. Ozdenizci, Near Field Communication (NFC): From Theory to Practice. Wiley, 2012.

[4] M. S. Gast, Building Applications with iBeacon: Proximity and Location Services with Bluetooth Low

Energy. O'Reilly Media, Inc., 2014.

[5] B. Glover, RFID Essentials. O'Reilly Media, Inc., 2006.

[6] J.E. Javitt, Z. Zhou, R.J. Willke, "Association between vision loss and higher medical care costs in Medicare beneficiaries' costs are greater for those with progressive vision loss," Ophthalmology, 114(2), 238-245, 2007.