Applications Using Microsoft Kinect Sensor with Robust Solutions and Algorithms

Ashikur Jhalak[1], Utshab Roy[1], Shoumik Rahman Mehedy[2]
Department of Computer Science & Engineering, Brac University, Dhaka, Bangladesh[1]
Department of Electrical and Computer Engineering, North South University, Dhaka,Bangladesh[2]

***Abstract*— Kinect is the machine device of Microsoft X-box 360. Information is connected more by text messages rather than voice. The paper focused on efficient and cost effective technique for voice to text conversion and image to text extraction. The paper also demonstrated the work to show the color, depth of kinect as well as body tracking system. The kinect can track up mostly 20 body joints. Obviously for the depth sensor, the tracking accuracy has been significantly improved. OCR which is mainly known as optical character recognition is used in kinect to recognize any object or image. Tesseract is an OCR engine. It is a tool used in OCR. By this tool system can understand and compile text value from real-time images.**

Keywords*— Kinect, OCR, Tesseract, image processing, pattern understanding, Audio output.

1. INTRODUCTION

For the last decades machine has been improved tremendously for building the ability from image to text. With the building research with the kinect Sensor, many problem in day to life has been optimized. The work of this paper will show the process of converting process of image to plain text, object detection as well as human body tracking jointing mostly 20 parts of the body. Voice to text message extraction is another feature of this paper research. This features can be helpful to many applications like: visually impaired person, any object detection in day to day life. Kinect produce high quality 3D scan of small and big object, virtual reality interaction, controlling robots with body movement and many parts of medicine and health care technology and applications[1,4].

Well, in many ways, kinect's color VGA video camera is called "RGB camera" shows the elements of colors it detect. Depth sensor used to show the 3D resolution regardless of the situation of the light. Kinect has four microphones which can recognize the voice the user from the different noises in the enclosed place like any room. This kind of features let the user to be few distance away like 3-4 feet from the microphone device and authorize the use of voice. In later parts the discussions have been elaborated with real time images with authentic solution. Of course the paper has showed the algorithms of the described work.

2. LITERATURE REVIEW

Singla and Yadav [1] explain to develop a cost effective and flexible Optical Character recognition speech synthesis system. The paper focuses with understanding of characters based on LabVIEW. Filipe et al. [2] represent a blind navigation support system on Microsoft Kinect. They showed a system that extends the use of traditional white cane by the blind for navigation. Depth data has used for pattern recognition using Kinect. For extracting features neural network has been applied.

A proposed work [3] called an assistance system by Khenkar et al. that showed ENVISION for visual impaired people who use smart phone. The system defines an smart decision to help visual impaired people. The system updates the tracks using GPS technology directions as well as new obstacle detection system. Tarkowski et al. [4] worked on a system that process data gained from Microsoft kinect. The data methods describes the shape of the surface and give a visualization about the thing's position and the object changed status.

In the paper [5,9] has worked on detecting objects using color and depth segmentation with kinect sensor using color segmentation. By RGB components this technique is done. For avoiding the disadvantage like sensitivity to changes of lighting color RGB image is converted to CIE-Lab color space that increase the accuracy of the color segmentation and avoid light sensitivity. Image to speech conversion for visual impaired [6] people, the process has been developed by Hagargund et al. from image to speech conversion so that they can easily understand from the images. The concept is an embedded system that took images, extract region of interest and convert the text to speech that will be helpful for blind people. Almost same concept has been developed in [7] by Sanjana and Rejina. The purpose to help visual impaired people.

All the proposed papers and researchers as well as authors basically proposed Microsoft kinect based OCR object detection and RGB image based segmentation to detect the obstacle for visually impaired people. Some works that show Image to text conversion and then speech conversion also shown on these papers. Some other work has also done based on smart phone GPS tracking system to find the obstacle for the blind. But frankly speaking it will be very difficult for the blind people to follow the instruction part by part just by hearing the voice command. Many papers show just their work on their purpose for visual impaired person.

In this paper the addition feature will be video to text conversion. Also other work has been demonstrated part by part according to algorithm wise and that might be helpful to other peoples especially some blind people. Human body as well as other object showing system is a feature in Microsoft kinect.

3. METHODOLOGY

At first, this system takes input as voice command that has recognized and extracted by Microsoft speech recognition tool. Kinect has built in microphone hardware that allows the Microsoft speech recognition library to extract the voice commands into simple text form. Then, system will follow the text instruction as a command that has been extracted by Kinect before. These instructions usually sent request to the system to search a particular object in between Kinect visual area.

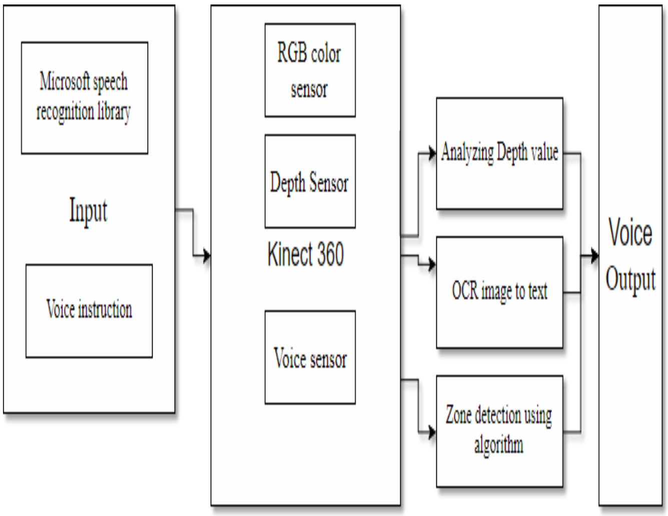


Fig.01 Block Diagram of the proposed Method

The System observe and scan for the text contained object from real-time video buffer of Kinect Camera. To recognize the text area of the object, this system uses Tesseract OCR-engine as Optical Character Recognition (OCR) library and image processing tool named EmguCV as an OpenCV library that built for C# wrapper. Afterwards, system will try to follow the instructions by using RGB (Red, Green and Blue) color sensor and depth sensor that is already included in Kinect hardware. After successfully extracted the text from video frames using OCR toolkit, the system will analyze depth and RGB value using Kinect sensors. Using zone detection algorithm, Kinect sensor will provide the 3D coordinate values of (X, Y, Z) form that gives the exact location of the object from the environment. At this point the system will put an end to its processing part and finalizing its output.

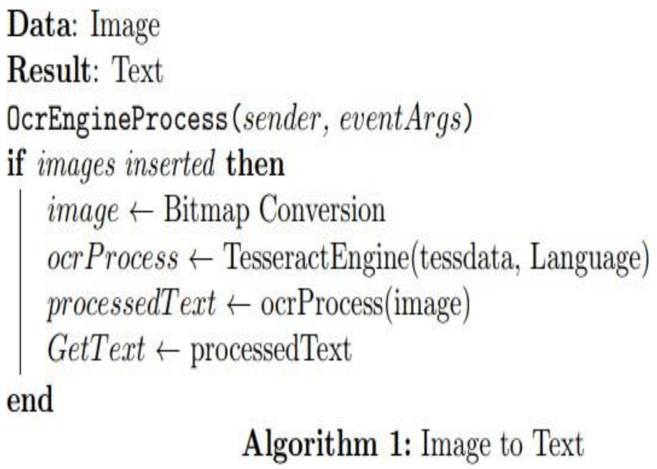
4. EXPERIMENTS

In this part, the explanation is all about the outcomes of the project. At first, we applied all the operation on a single image. Our first approach is Aspose. OCR tool which is used for the text extraction form any image. It can extract text from any plain image but the challenge of that engine is that it cannot extract any text. There are other values present alongside the text. In a simple word that engine only extract text where there is nothing but text is presented. Moreover, it has the charter limitation. It can only extract text up to a limited number of charter. More the 25 charter cannot be determined by the Aspose.

Though in a object the name of the object is less than 25 charter but the object contains some visual text and design. It sometimes has visual instruction of usage of that product and the company name and logo is also present there. So that it gave us some unwanted value that is not at all related to the object name. The avg. total charter error is 20.9 and the word precision error is 37.2. It’s a high precision error rate and many cases we cannot even able to find the desire result which we are looking for. That is why we had to look for the second approach to solve that issue. For that purpose, we chose another OCR engine that is Google Tesseract. It is basically an open source library that has been developed and maintained by Google. As it is an open source project that is why we had the full access of that library. We change and modified that library according to our demand and changed the threshold value of the image and get almost close result compared to our expected result. According to accuracy and performance [number]. The avg. total charter error is 11.0 and the word precision error is 25.4 which is a good rate of accuracy.

4.1 Image And Voice To Text Algorithm

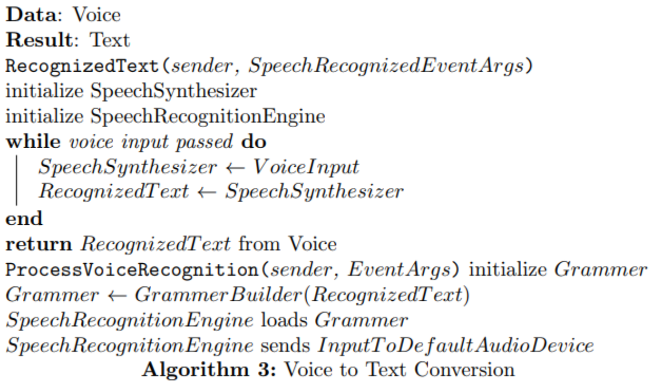
In the beginning, we started our work by taking one test image file as input. We took the image file in order to convert that image into plaintext format. We followed the algorithm that is shown below figure in purpose of convert image to text format. Same image file is processed into several OCR engines to get better accuracy and decision to pick one of the OCR engine that should be reliable for further implementation.



```
Data: Image
Result: Text
OcrEngineProcess(sender, eventArgs)
if images inserted then
    image ← Bitmap Conversion
    ocrProcess ← TesseractEngine(tessdata, Language)
    processedText ← ocrProcess(image)
    GetText ← processedText
end
Algorithm 1: Image to Text
```

Fig.02 Image to text conversion Algorithm

In this algorithm, the method OcrEngineProcess (sender, eventArgs) takes an object named ‘sender’ and an event named ‘eventArgs’ where the sender object includes a reference that control an event of a program.

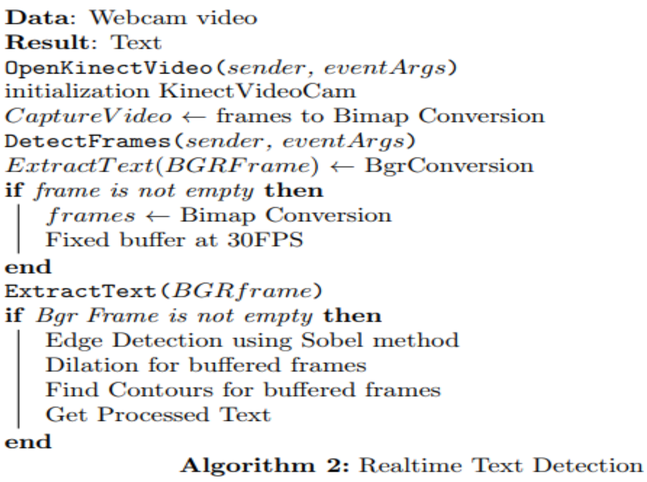


```
Data: Voice
Result: Text
RecognizedText(sender, SpeechRecognizedEventArgs)
initialize SpeechSynthesizer
initialize SpeechRecognitionEngine
while voice input passed do
    SpeechSynthesizer ← VoiceInput
    RecognizedText ← SpeechSynthesizer
end
return RecognizedText from Voice
ProcessVoiceRecognition(sender, EventArgs) initialize Grammer
Grammer ← GrammerBuilder(RecognizedText)
SpeechRecognitionEngine loads Grammer
SpeechRecognitionEngine sends InputToDefaultAudioDevice
Algorithm 3: Voice to Text Conversion
```

Fig.03 Voice to text conversion Algorithm

4.2 Real Time Text Detection Algorithm

After successful conversion of image to text, we build another algorithm that extracts text from real-time Kinect video. This algorithms is shown and illustrated in figure 03. In this algorithm, it takes input data as buffered frames from Kinect Video capture in real-time. And as an output it gives plain text by extracting text from object in real-time video frames. Initially, the method Open Kinect Video (sender, eventArgs) takes an object named ‘sender’ and an event named ‘eventArgs’ where the sender object includes a reference that control an event of a program. At the final step of this method, the extracted text included frames will be shown in thirty FPS (Frame per Second) buffer rate.



```
Data: Webcam video
Result: Text
OpenKinectVideo(sender, eventArgs)
initialization KinectVideoCam
CaptureVideo ← frames to Bimap Conversion
DetectFrames(sender, eventArgs)
ExtractText(BGRFrame) ← BgrConversion
if frame is not empty then
    frames ← Bimap Conversion
    Fixed buffer at 30FPS
end
ExtractText(BGRframe)
if Bgr Frame is not empty then
    Edge Detection using Sobel method
    Dilation for buffered frames
    Find Contours for buffered frames
    Get Processed Text
end
Algorithm 2: Realtime Text Detection
```

Fig.04 Real time text detection Algorithm.

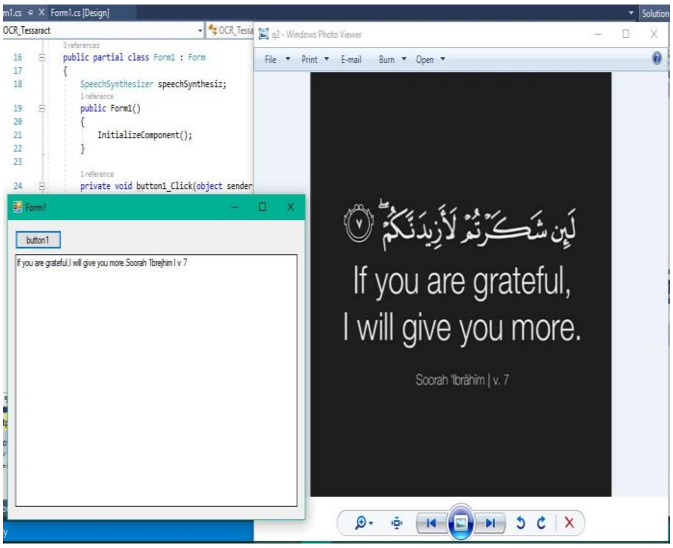
5. RESULT ANALYSIS

For text extraction we have used two OCR engine with different performance. For betterment of our system we used one of them. Performance Analysis of OCR engine has been analyzed.

Table.01 Tesseract OCR vs. Aspose OCR engines

OCR Engine	Test Language	Text extraction		Total Character Errors	Word Precision Errors
		Plain Text	RGB Colored Text		
Google Tesseract	English	Yes	Yes	11.0	25.4
Aspose OCR	English	Yes	No	20.9	37.2

From above the character and precision errors, it is noticeable that Tesseract OCR engine has less errors than Aspose OCR engine. Aspose also fails to extract text from colored image where Tesseract OCR engine can perform.



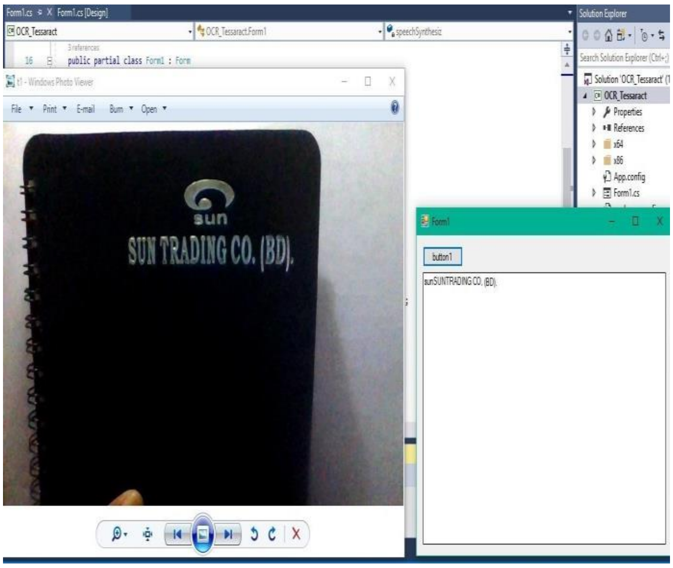
```
public partial class Form1 : Form
{
    SpeechSynthesizer speechSynthesiz;
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender
```

If you are grateful,
I will give you more.
Soorah 'Ibrâhîm | v. 7

If you are grateful.I will give you more.Soorah 1brejhim l v 7

Fig. 05 Plain-Text Result of Google Tesseract OCR Engine.

This result example gives us the extracted text value from the image. The black background image is a simple image which contains some text in two different language. As we can see in the output panel the English text part has been successfully extracted which processed by OCR engine using English language's trained data.

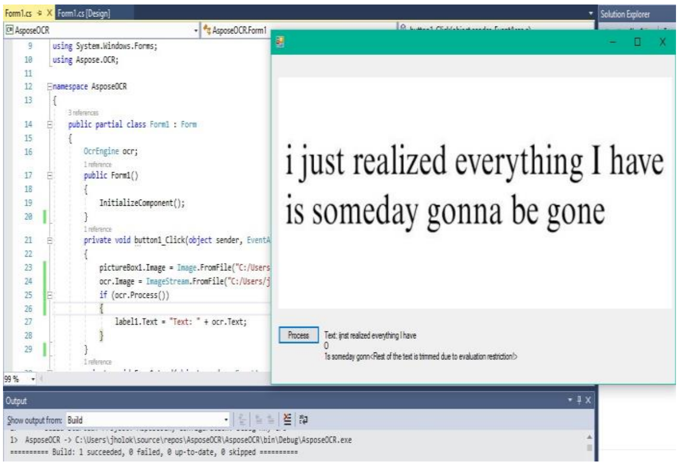


```
public partial class Form1 : Form
```

sunSUNTRADING CO. (BD).

Fig. 06 Colored-Text Result of Tesseract OCR Engine.

In figure 06, from the output panel, we can see that here also the system extracted the text successfully and the unwanted logo part has been cut off from the extraction as it was not recognized by OCR engine as characters.



```
using System.Windows.Forms;
using Aspose.OCR;

namespace AsposeOCR
{
    public partial class Form1 : Form
    {
        OcrEngine ocr;
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventA
        {
            pictureBox1.Image = Image.FromFile("C:/Users
            ocr.Image = ImageStream.FromFile("C:/Users/j
            if (ocr.Process())
            {
                label1.Text = "Text: " + ocr.Text;
            }
        }
```

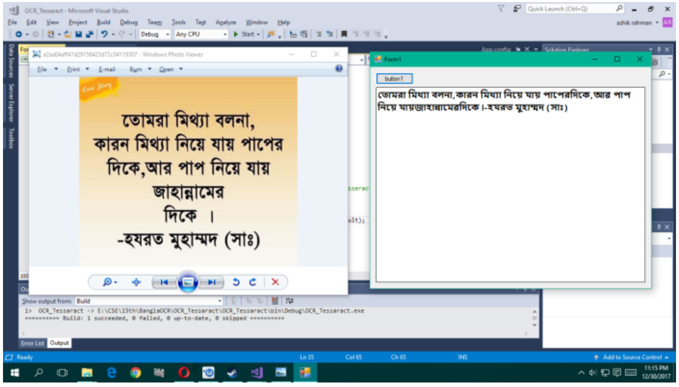
i just realized everything I have
is someday gonna be gone

Text: ijust realized everything I have
0
Is someday gonn<Rest of the text is trimmed due to evaluation restriction!>

```
1>  AsposeOCR -> C:\Users\jholok\source\repos\AsposeOCR\AsposeOCR\bin\Debug\AsposeOCR.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Fig. 07 Plain-Text Result of Aspose OCR Engine

This is an example of Aspose OCR. This program has been developed by using Aspose.OCR engine. It only works perfectly when there is plain text and no background image or design. It do not even give any veiled result for handwritten text. So, when we applied this system for the object we could not able to select and region of interest and extract text from any handwritten text. It extract text faster but has some serious limitation. Our task need something different and upgraded engine.



তোমরা মিথ্যা বলনা,
কারন মিথ্যা নিয়ে যায় পাপের
দিকে,আর পাপ নিয়ে যায়
জাহান্নামের
দিকে ।
-হযরত মুহাম্মদ (সাঃ)

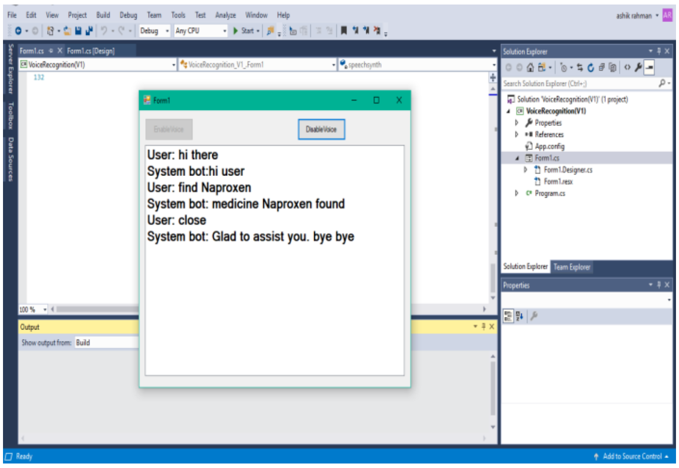
তোমরা মিথ্যা বলনা,কারন মিথ্যা নিয়ে যায় পাপেরদিকে,আর পাপ নিয়ে যায়জাহান্নামেরদিকে ।-হযরত মুহাম্মদ (সাঃ)

```
1>  OCR_Tessaract -> E:\CSE\15th\BanglaOCR\OCR_Tessaract\OCR_Tessaract\bin\Debug\OCR_Tessaract.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Fig. 08 Plain-Text Result of Tesseract OCR Engine in Bangla.

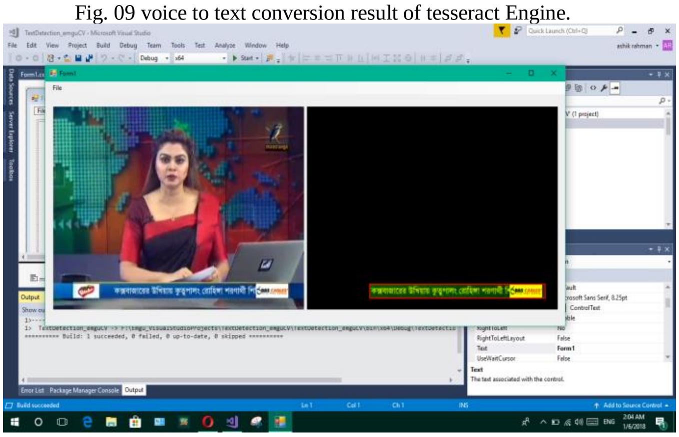
The image to text algorithm works fine with the Bangla language testing. As Bangla language is the first language for our country Bangladesh and people are likely to use this language most of the time and feel comfortable.

Now coming to the point of voice to text conversion, basically a BOT system is provided with some trained data that is capable to reply the user's question on the basis of category. A microphone is needed to understand the user's inquiry and set him/her the proper answer that is required for the situation and make perfect balance between conversation.



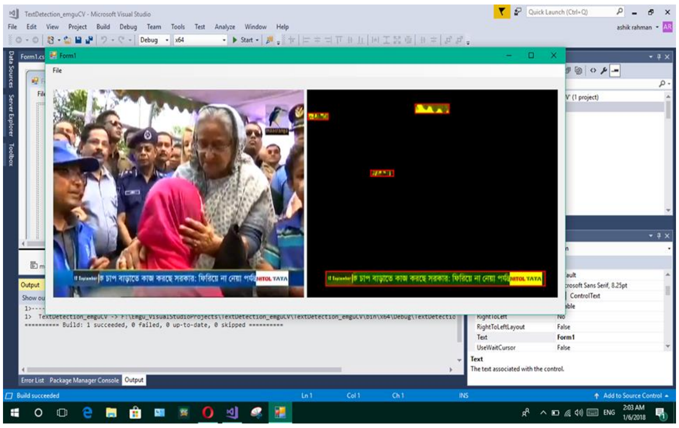
User: hi there
System bot:hi user
User: find Naproxen
System bot: medicine Naproxen found
User: close
System bot: Glad to assist you. bye bye

Fig. 09 voice to text conversion result of tesseract Engine.



```
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Fig.10 video to text conversion result of tesseract engine.



```
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Fig.11 video to text conversion with more accurate result.

5.1 Kinect Depth Image Classification

Any kinds of object or other types of thing detection is done separately for each frame based on depth and color maps. RGB-D (depth sensor) cameras are sensing systems that capture RGB images along with per-pixel depth information [8] or time-of-flight sensing to generate depth estimates at a large number of pixels. RGB-D cameras allow the capture of reasonably accurate mid-resolution depth and appearance information at high data rates. It has generally 180 degree laser scanner for object detection [9]. It could be any obstacles or Human and any kind of matter that is visible. In figure 12,13,14 body joints are identified by kinect sensor and other depth image analyses.

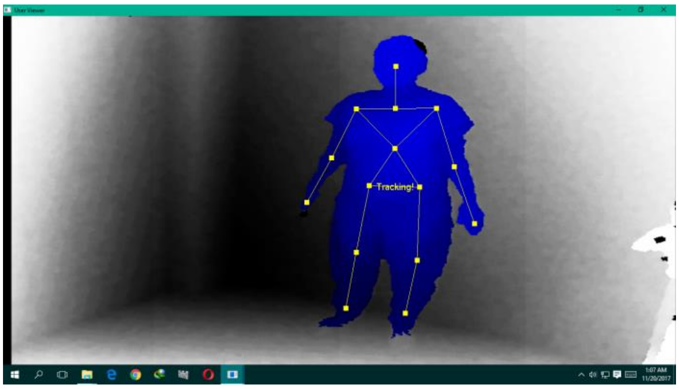


Fig.12 Kinect RGB (red,green,blue) camera depth image-1

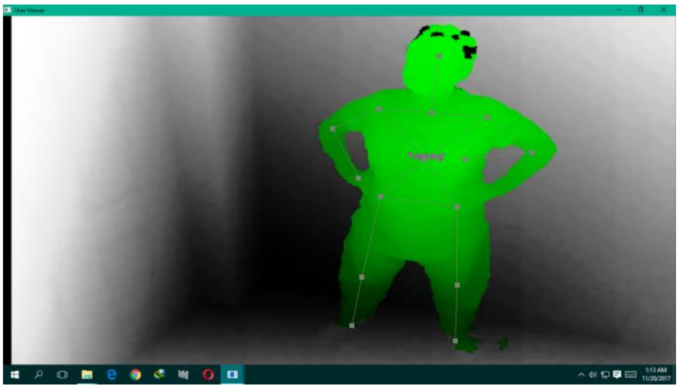


Fig.13 Kinect RGB (red,green,blue) camera depth image-2

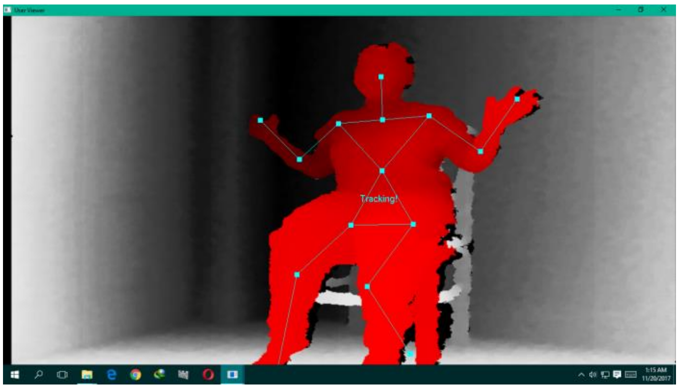


Fig.14 Kinect RGB (red,green,blue) camera depth image-3

Color class map defers green, red and blue color that is identifying a human. These Color classification is made on color image of the kinect. The classification is applied only to those pixel whose slope is above a threshold [9].

6. FUTURE WORK

1. Other optical character recognition model like AForge OCR, ABBYY OCR, Iron-OCR are also popular tools which can be used for better text extraction result which improve the accuracy of the extracted text from images. 2.By better Training we will Improve extraction Accuracy. 3. Add more features on Kinect and make it looks better.

7.CONCLUSION

So far the proposed work has been done on Microsoft kinect application on different aspect and it's robust algorithm with different features. Image to text extraction, real time video to text conversion and voice to text reply by BOT system to the user using kinect is the focus of the paper. Object detection like human with showing body joints is a feature of kinect RGB-D camera sensor. Better texture classification can be achieved by using a higher resolution image. The proposed work has been successfully shown about Applications using microsoft Kinect sensor with robust solutions and algorithms.

.REFERENCES

[1] S.K. Singla and R.K. Yadav(2014),"Optical Character Recognition Based Speech Synthesis System Using LabVIEW", Journal of Applied Research and Technology, vol.12, pp. 919-926. Available:https://doi.org/10.1016/S1665-6423(14)70598X [Accessed in September 25, 2017]

[2] F. Vitor, F. Filipe, F. Hugo, S. Antonio, P. Hugo and Joao Barroso, "Blind navigation support system based on Microsoft Kinect", Proceedings of the 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2012), 2012,pp.94-101.

[3] K. Shoroog, A. Hanan, I. Shahad, F. Alaa, K.M. Salma and B.A Hanene, "ENVISION: Assisted Navigation of Visually Impaired Smartphone Users", Conference on ENTERprise Information Systems/ International Conference on Project management/ Conference on Health and social care Information System and Technologies, CENTERIS/ ProjMAN/HCist, October 5-7, 2016, pp. 128-135.

[4] T. Pawel, Ireneusz Malujda, T. Krzysztof, K. Mateusz and G. Jan, "Adjustment of the distance of objects to the Microsoft kinect device fitted with nyko zoom attachment used in a three-axis manipulator", XXI International Polish-Slovak Conference "machine Modeling and Simulations 2016", 2017, pp. 387-392.

[5] H.L. Jose-Juan, Q.O. Ana-Linnet, L.R. Jose-Luis, R.B. Francisco-Javier, I.B. Mario-Alberto and A.O Dora-Luz, "Detecting object using color and depth segmentation with kinect sensor", The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science, 2012, pp. 196-204.

[6] G.H Asha, V.T. Sharsha, B. Mitadru and F.S Eram(2017), "Image to Speech Conversion for Visually Impaired", International Journal of Latest Research in Engineering and Technology(IJLRET), Vol.03-Issue 06, pp.09-15, Available: http://www.ijlret.com/Papers/Vol-3-issue-6/2-B2017160.pdf [Accessed in December 21, 2017]

[7] Sanjana.B and J. Rejina Parvin(2016), "Voice Assisted Text Reading System for Visually Impaired Persons Using TTS Method", IOSR Journal of VLSI and Sigal Processing (IOSR-JVSP), Vol. 6, Issue 3, version III, pp. 15-23.

[8] k. Konolige, "Projected texture sterio", Proceeding of the IEEE Conference on Robotics and Automation(ICRA), 2010.

[9] N. Sharon, G. Jacob and A. Victor(2015), "Obstacle detection in a greenhouse environment using the kinect sensor", Computers and Electronics in Agriculture, vol. 113, pp.104-115. . Available: https://doi.org/10.1016/j.compag.2015.02.001 [Accessed in December 21, 2017]