International Journal of Engineering and Manufacturing Science. ISSN 2249-3115 Vol.8, No.3, (2018)

HSV COLOUR CONVERSION BASED VIRTUAL KEYBOARDWITH GAZE CONTROLLED MOUSE

Usharani M[1], Barath kumar G V[2], Mahesh M[3], Mallavajhala.Nagavikhyath[4], Sathyanarayanan S[5]
1-Asst.Professor,Department of Electronics and Communication Engineering,
2,3,4,5-UGStudents , Department of Electronics and Communication Engineering,,
Velammal Engineering college,Chennai.
usharani@velammal.edu.in

***Abstract*— In this modern era, technological improvements have taken the world by storm. Today, almost all the people in the world have access to computers and mobile phones, which have become an integral part of our lives. New research in promising fields like virtual reality, have radically transformed both the applications, and physical appearance of future computers, and has even been able to grant people with severe physical impairments the ability to use computers and other common digital devices that they otherwise could never use. Unfortunately, most of the current work in the field of virtual reality, only focusses on assisting the physically challenged, and the devices suggested are almost always too complex, or too expensive to use. These devices can be used only by physically impaired patients, and they are too arduous for a normal person to use. It would be another great technological improvement if scientists were able to create a virtual reality device, which can be used comfortably by both physically challenged patients and ordinary people. The principle behind virtual reality is to detect one of our five senses, mostly our senses of touch and sight as inputs to the device and convert them into a form which can be understood and processed by the device, to show us the desired output. In this case, we are going to take the example of a keyboard based on virtual reality, which is made using the concept of digital image processing. Using a camera to detect our hand movements and gestures as an input, the keyboard is designed to use a digital image processor along with open source computer vision software (OpenCV) to locate the position of our fingers, by means of sensing the colour stickers attached to them and matching the position of our fingers with the corresponding letters on an image of an ordinary keyboard, which is shown on a display device. The selected letters are then displayed on the screen. Many models of virtual keyboards exist, but they all face the common problem of being exorbitantly priced and having a complicated design. This version of the virtual keyboard is proposed to be highly cost effective, and very simple in terms of usage and design. This keyboard can be used by ordinary people as well, and only requires a simple camera instead of complex sensors and other hardware.**

***Index Terms*- Camera, color stickers, hand movements/gestures, image processing, Intel core i7 processor, virtual reality keyboard**

I. INTRODUCTION

Scientists and computer engineers around the world have been constantly working to improve existing technology, for the benefit of mankind, and to take us closer to the future step, by step. Virtual reality is one area of current research, where engineers have been trying to help severely disabled people, such as those who are totally paralyzed, or afflicted by severe speech disorders, neuro-locomotor problems, etc,[12] by modifying everyday devices like the keyboard and mouse, touchscreen of our personal computers or mobile phones in a manner that would be suitable for severely disabled people to use. This type of technology is referred to as assistive technology, which aims to help such people through various forms of human-machine interactive interfaces. The uses of assistive technology are not limited to helping the disabled but plays a very important role in many other fields. In the field of medicine, augmented reality, which is another form of virtual reality can be used to provide a direct visualization of an ultrasound imaging data on the patient's body improving the doctor's understanding and more recently with a cooperative surgical robot system, guided by hand gesture which incorporates the advantages of the AR visual guidance, the surgeon's experience and robotic surgery accuracy. Without this, it would not be possible for doctors to see what is happening inside our body. Many new possibilities for teaching and learning based on augmented reality have been developed. The merging of virtual objects with the real environment allows the use of the AR in classroom-based education for subjects like Chemistry, where we can show the molecular structure of elements with a higher level of clarity, in anatomy to visualize the relationships in three dimensions of various organs and their interdependent functions in 3 dimensions, Mathematics and geometry education, and so on.[13] Due to the increase of computer power and the decrease of component size, new mobile computing devices are rapidly accessible to people providing an access to online resources anytime and everywhere. Consequently, developers has exploited this flexibility to create an AR application that integrates virtual information into a person physical's environment, so people perceive that information as it exists in their surroundings. But as we all know, all such benefits come at a price. There are some drawbacks, which are stopping us from using virtual reality and augmented reality to its full potential, and we have made an attempt to solve some of these issues, in our model of the virtual keyboard.

Most of the research projects and work that we have analysed mention the cost of the human-machine interfaces as a major hindrance to developing this technology and offering it to the common people. The devices also have to be worn as headgear, or in some other form, which may sometimes cause discomfort for the user, which is another common defect that comes to our notice. For example, various systems have been proposed, in particular, brain–computer interfaces(BCI)using non-invasive electroencephalography (EEG) recordings has been a fundamental active research field for the development of virtual keyboards that allow severely disabled people to communicate. Recent BCI systems include a combination of different modalities, such as eye-tracking. Non-invasive BCIs based on the detection of event-related potentials, have been the pioneer application in this field. As we can clearly see, equipment like the EEG are very expensive, intrusive, and difficult to bring into the scenario of our everyday life. The use of expensive eye trackers also causes fatigue, since we constantly have to keep focusing on particular areas of the screen, even if we want to type a simple word. Further, such technology has been designed, keeping only the disabled people in mind. It is undeniably a great service, but it only covers a small population of the people. Devices which can be used easily by both the disabled and the abled have not yet been fully developed.

Many different methods have also been devised to make a single device, like the virtual keyboard. One way which was implemented in already existing works was to use a tree-like algorithm, which consists of 8 commands, with each command as a group of six letters of the alphabet. Once one of the 8 commands were selected, the command would expand into six more commands to choose a particular letter to type on screen. This drastically changes the way we have to use the keyboard, which is yet another shortcoming. It also makes the process of typing one single letter, much longer than needed. In another model, the technique of using a combination of a camera and a projector has been suggested, but difficulties arise in turning a simple flat surface into an interactive touchscreen. A method of detecting our skin colour to find the position of contact has been suggested, but since our skin colour is spread throughout our body, and is not unique to our fingers alone, the camera and sensors used may be overwhelmed, and may make wrong decisions, which reduces the accuracy of the output. The usage of gloves with colour sensors were also suggested and implemented, but it limited the feeling of natural interaction with the keyboard and our hands. The infrared and capacitive sensors used to detect the touch or position of our fingers, were also very expensive.

Our project, as already mentioned, mostly aims to solve some of the above-mentioned difficulties. Our version of the virtual keyboard is comparatively more cost effective, and simpler in terms of the algorithm used, since it is done with a simple laptop, which is owned by most people due to advancements in technology. We made use of three key concepts, namely virtual reality/augmented reality, image processing and eye tracking. The three terms can be defined in short as follows. Digital image processing is the technique of improving and modifying the representation of digital pictures for human interpretation. It is also defined as the processing of image data for the storage, transmission and representation for autonomous machine perception. Augmented reality is a type of virtual reality that combines a real scene viewed by the user with computer generated data in order to augment the scene with additional information. Eye tracking is the process of tracing the path of the user's eyes, to determine various parameters related to his or her interest, and cognition, which is used in our project, in place of the mouse cursor. Our project has the following advantages and improvements over the previous models of the virtual keyboard.

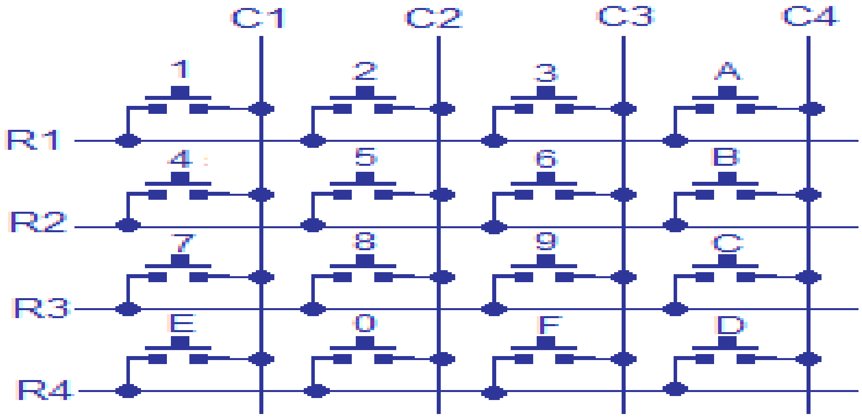
- All the existing models mention a common problem of cost, which has been reduced in our project
- We are using the eye tracker only as a mouse which reduces eye strain
- According to some models, the keys are coloured, and the algorithm is written accordingly, but in our keyboard, we use coloured gloves/ stickers, and the algorithm is written to sense these colours, and so it is easier to use and has a wide variety of colours
- This has already been suggested, but has not been implemented, due to a comparatively lower accuracy, but it is simple, cost effective and meets the basic requirements for functionality, since we are using only an already existing laptop, with Python and OpenCV computer software.
- It can be used to give presentations, for teaching classes, and can be used anywhere, anytime, as in the case of previous models.
- Other virtual keyboards that we have seen, do not have a mouse, to go along with them, which we think is an essential part, when using the computer, and so we have included this in the form of an eye tracker, which is our small contribution to the virtual keyboard.

Section I gives a small comparison between an ordinary keyboard and our keyboard, Section II describes the components and software used in detail, Section III explains the algorithm and coding used, and Section IV and V explain the procedure and result.

II. COMPARISON OF ORDINARY KEYBOARD AND VIRTUAL KEYBOARD

All digital devices with some processing capability have an interactive interface which obtains the inputs from the user, and gives a specified output, according to the function they are meant to perform. In the case of an ordinary desktop or laptop computer, this is the keyboard. With an ordinary keyboard, we would have to touch and press each of the buttons on it to instruct the computer to display the words that we intend to write, on the screen. Other virtual keyboards, as mentioned before, either use only the sense of sight as a means of interaction with the user, or use complex algorithms involving IR sensing of our finger's colours, to take in our input. Our keyboard, makes use of a simple camera-monitor system, which is already available in our laptops, to sense the various colours on our hand gloves, using digital image processing. To understand the fundamental differences between how an ordinary keyboard operates, and our keyboard, and the differences between our virtual keyboard and other virtual keyboards, we will now see the operation of a simple keyboard.

An ordinary keyboard contains the various letters arranged in the form of a matrix, of capacitive switches. Capacitive switches are considered to be non-mechanical because they do not physically complete the circuit. Instead, current constantly flows through all parts of the key matrix. Each key is spring-loaded and has a tiny plate attached to the bottom of it. When you press a key, it moves this plate closer to the plate below it. As the two plates move closer together, the amount of current flowing through the matrix changes. The processor detects the change and interprets it as a key press for that location. Capacitive switch keyboards are expensive, but they have a longer life than any other keyboard. The figure shows the basic keyboard matrix (Hex keyboard).[16]

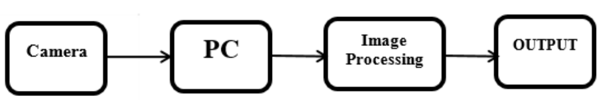


The keypad consists of a number of wires arranged in a mesh structure to form rows and columns, which are connected by MOS transistors as shown in the diagram. The keyboard uses the processor interface to perform row and column scanning. The key we press connects the particular row and column by switching on the MOS transistor. The selected row and column go low, while the other rows and columns are maintained at a high state, to indicate the key press, which is detected using the row and column scanning methods of the processor. This is sent to the display lines which are eventually used to display the selected letter on our monitor.

Here, we can see that all key presses are detected by the physical intersection of a row and column. In our keyboard, the detection of the keys is done using digital image processing, as follows.

First, an image of a keyboard will be projected on the screen. Next the, user is asked to wear the coloured gloves or stickers on his hands, and the camera on the laptop will be turned on to capture live video images of the user's finger movements. When the user points to or selects a particular letter on the keyboard, the computer will lock the colour of the user's fingers within the particular area allocated for the key which is being pressed, and the particular letter will be displayed on the output screen. Colour conversion from RGB to HSV colour codes, are used for this purpose, which will be explained in the following sections.

III. HAND GESTURE DETECTION USING THE CAMERA



The basic block diagram of our virtual keyboard system is as shown in the figure. The system consists of a PC, camera and monitor. The camera is used to capture and recognize an object in view. In this case, it tracks the user's hand gestures using computer- vision based techniques. It sends the data to the PC. The camera, in a sense, acts as a digital eye, seeing what the user sees. It also tracks the movements of the thumbs and index fingers of both of the user's hands. When the user touches a key on the keyboard, the axis coordinates are found, and the coordinates are compared with already taken template coordinates to find which key the user pressed by using Python as the programming language and open CV library files. We will now see the basic logic behind the camera.

The camera converts the colour on the user's fingers from an RGB format to HSV format. In order to understand clearly, we need to understand the following simple concept. There are three basic types of images in digital image processing:

- RGB Image: Red-Green-Blue Image

- Grayscale Image
- Black and White Image

Each image is made up a specified number of pixels. A pixel is a fragment of the total image. Each of these pixels contain information about the colour and intensity of each part of the image. It is based on this information that the images are classified. In the RGB image, only 3 colours, namely red, green, and blue are used, and all other colours are derived as a combination of these colours. The pixels hold a value from 0-255 for each colour, representing the amount of red, green and blue components. In a grayscale image, the pixels hold values from 0-255, which represent the intensity of just one colour, grey, and so, the image appears to be grey. Black and white images represent black colour components by a binary 1 and white colour components as a binary 0. There is another type of image called the HSV image (Hue, Saturation, and Value). Hue refers to the colour of the image, with a range of colours from 0-255, red being 0. It also specifies the area of the coloured region in the form of a circular region. Saturation refers to the colour intensity, again on a scale of 0-255. Value is the brightness of the colour, from 0 to 255.

Once the user presses the key, the camera converts the RGB image to HSV as it has all the features of the 3 basic types of images. In short, once the user presses a key, the following steps occur:

- RGB image is converted to grayscale, using the formula Gray=[R×(2^2)]+[B×(2^4)]+[G×(2^6)]
- The grayscale image is now converted to a black and white format using the following steps:
- First, take the gray value of the central pixel, and compare it with all other pixels. If the value of the pixel is less than the central pixel, mark the new black and white value as 0. Otherwise mark the value of the pixel as 1. Here, 1 represents white, and 0 represents black.
- Now we can convert the RGB image to HSB. The H value contains the area, and the colour, taken from RGB, saturation contains the colour intensity, and a value of 0 or 1 is stored in the value field based on the information provided by the black and white image.

This allows the computer to isolate the selected key, by representing it as a black dot, and all the other keys in white. We are using Python for programming since it is very flexible and easy to understand when compared to other coding languages, and openCV, to enable the interfacing of the camera to the computer processor.[16]

RGB to HSV conversion formula

The R,G,B values are divided by 255 to change the range from 0..255 to 0..1:

$R' = R/255$

$G' = G/255$

$B' = B/255$

$Cmax = \max(R', G', B')$

$Cmin = \min(R', G', B')$

$\Delta = Cmax - Cmin$

Hue calculation:

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation:

V = Cmax

RGB to HSV color table

Color	Color name	Hex	(R,G,B)	(H,S,V)
	Black	#000000	(0,0,0)	(0°,0%,0%)
	White	#FFFFFF	(255,255,255)	(0°,0%,100%)
	Red	#FF0000	(255,0,0)	(0°,100%,100%)
	Lime	#00FF00	(0,255,0)	(120°,100%,100%)
	Blue	#0000FF	(0,0,255)	(240°,100%,100%)
	Yellow	#FFFF00	(255,255,0)	(60°,100%,100%)
	Cyan	#00FFFF	(0,255,255)	(180°,100%,100%)
	Magenta	#FF00FF	(255,0,255)	(300°,100%,100%)
	Silver	#C0C0C0	(192,192,192)	(0°,0%,75%)
	Gray	#808080	(128,128,128)	(0°,0%,50%)
	Maroon	#800000	(128,0,0)	(0°,100%,50%)
	Olive	#808000	(128,128,0)	(60°,100%,50%)
	Green	#008000	(0,128,0)	(120°,100%,50%)
	Purple	#800080	(128,0,128)	(300°,100%,50%)
	Teal	#008080	(0,128,128)	(180°,100%,50%)
	Navy	#000080	(0,0,128)	(240°,100%,50%)

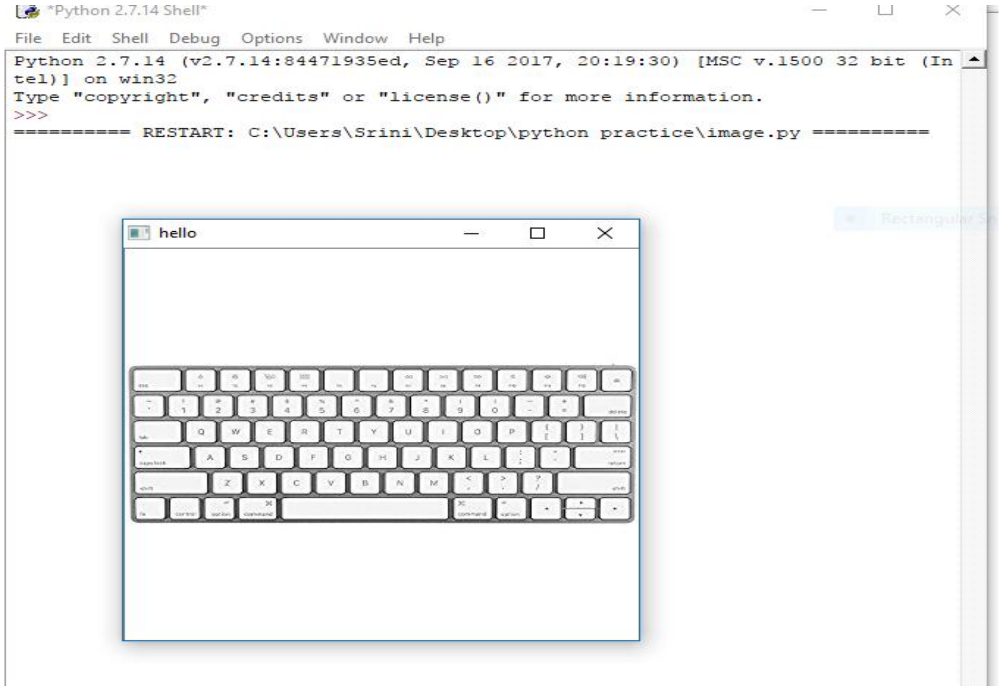
IV. PROGRAMS AND EXPLANATIONS

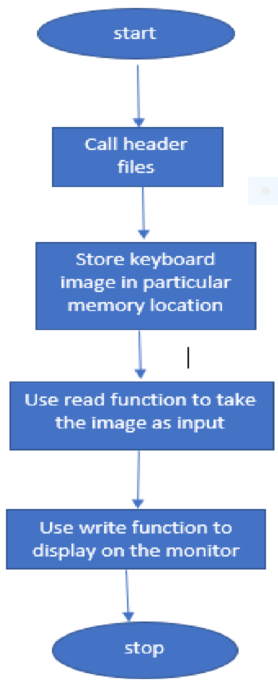
The python code for displaying the keyboard image on the screen, using the camera to track the position of our fingers, and for conversion from RGB to HSV format is explained as follows:

A) Program for displaying the keyboard:

The program for displaying the image of the keyboard is shown in the figure. Import commands are used to call the Open source Computer Vision software, so that we can use it to interface the camera and the monitor. The first two commands in the code serve this purpose. The time header file is used to provide a small delay if required, before the keyboard image is displayed. This is done in the third line. The next two commands are used to store the existing image of the keyboard and display it on the monitor. The "time.sleep" command is used to specify the delay time in milliseconds.

The sample output is as shown in the figure.

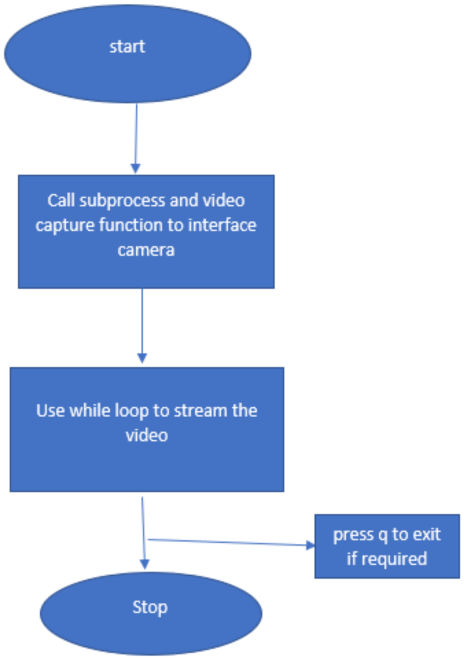




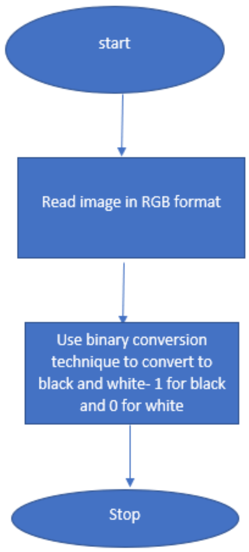
B) Program for capturing live video images using the laptop camera

The program for capturing the live video of the user selecting the required key is as shown in the next figure. The first two commands are the same, and are common for all other program modules since we are using openCV in all cases. The "subprocess" header file is called for interfacing external devices with the processor in general. Additional code must be written for this purpose, but since the camera is inbuilt in modern laptops, it is simply enough if we include this header file.
VideoCapture is an inbuilt function of python, used to stream live video. We are making use of this function to capture the user's hand gestures, while typing on the keyboard. We also specify an infinite while loop, since we have to continuously stream the video as an input. The return command is used to

return a true or false value to the processor, based on which we will be doing the processor selects the key which is pressed. The video is also simultaneously displayed on the screen.



C) Program for converting video from RGB to grayscale image:
This program converts the RGB image to the HSV image. The formula mentioned in the algorithm is implemented using the command:
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)



V.RESULTS

In our project, we have managed to create a virtual reality keyboard which overcomes some of the critical drawbacks of existing virtual keyboards, mainly the cost. Our version of the virtual reality keyboard can be used by both the disabled and the abled, as it can be programmed according to the specific needs of both kinds of people, which gives our keyboard more flexibility over other keyboards. The speed and accuracy may not be as good as existing models, but it is justifiably fast and reliable enough, so that it can be used in everyday life, hopefully in the near future.

VI. REFERENCES

[1] B. Wu, T. Mei, W.-H. Cheng, and Y. Zhang, "Unfolding temporal dynamics: Predicting social media popularity using multi-scale temporal decomposition," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 272–278.

[2] T. Mei, Y. Rui, S. Li, and Q. Tian, "Multimedia search reranking: A literature survey," *ACM Comput. Surv.*, vol. 46, no. 3, p. 38, 2014.

[3] W. H. Hsu, L. S. Kennedy, and S.-F. Chang, "Reranking methods for visual search," *IEEE Multimedia*, vol. 14, no. 3, pp. 14–22, Jul./Sep. 2007.

[4] W. H. Hsu, L. S. Kennedy, and S.-F. Chang, "Video search reranking through random walk over document-level context graph," in *Proc. 15thACM Int. Conf. Multimedia*, 2007, pp. 971–980.

[5] R. Yan, A. Hauptmann, and R. Jin, "Multimedia search with pseudorelevance feedback," in *Image and Video Retrieval*. Berlin, Germany: Springer, 2003, pp. 238–247.

[6] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for Web image search," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4649–4661, Nov. 2013.

[7] T. Yao, C. W. Ngo, and T. Mei, "Circular reranking for visual search," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1644–1655, Apr. 2013.

[8] J. Cui, F. Wen, and X. Tang, "Real time Google and live image search re-ranking," in *Proc. 16th ACM Int. Conf. Multimedia*, 2008, pp. 729–732.

[9] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans.Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 644–655, Sep. 1998.

[10] X. Tian, D. Tao, X.-S. Hua, and X. Wu, "Active reranking for Web image search," *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 805–820, Mar. 2010.

[11] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Syst.*, vol. 8, no. 6, pp. 536–544, Apr. 2003.

[12]Hubert Ceccoti," A Multimodal Gaze-Controlled Virtual Keyboard", *IEEE Transactions on Human Machine systems*

[13]MarwaKarray, Mohamed Krid, and DorraSellami," A survey of AR systems and a case study of virtual keyboard based camera projector system", IEEE IPAS'16: INTERNATIONAL IMAGE PROCESSING APPLICATIONS AND SYSTEMS CONFERENCE 2016

[15]https://www.computerhope.com/jargon/k/keyboard.htm

[16] https://www.rapidtables.com/convert/color/rgb-to-hsv.html

[17] http://www.circuitstoday.com/interfacing-hex-keypad-to-8051

[18] https://computer.howstuffworks.com/keyboard3.htm

[19]https://www.slideshare.net/sahilbiswas/image-processing-27960248