Innovation 2020
Organised by
Computer Engineering Department, Dr. D. Y. Patil School of Engineering, Lohegaon, Pune, Maharashtra, India in association with
International Journal of Scientific Research in Science, Engineering and Technology

Smart Keyboard Using Internet of Things

Rahi Patel[1], Bhavik Bhrambhatt[2], Shivam Bendbar[3], Dr. Sheshang D. Degadwala[4]
[1-3]U.G. Scholar, Computer Engineering, Sigma Institute of Engineering Vadodara, Gujarat, India
[4]Associate Professor Computer Engineering, Sigma Institute of Engineering Vadodara, Gujarat, India

ABSTRACT

A keyboard is a main input device for any communication or any computer device mainly the portable one. Our objective was to interact between a computer and human-being, a system of keyboard is represented by means of camera and Image Processing. Our physical keyboard consists of a camera, Paper, Raspberry-pi-3b+ and a stand (approx. 35cm from the surface). The camera is used to capture several frames of finger tip to determine the button pressed. Every Keystroke can be determined by the camera using Imager thresholding and contours. Our Experiment shows that it is a reliable sense of technology with high accuracy.

Keywords : Paper Keyboard, Virtual Keyboard, Augmented Reality, Keyboard using sensor.

I. INTRODUCTION

People within ongoing condition are being occupied with their duties that they don't have a moment to waste. Others thrive to meet their daily job by using appropriate technology like Desktop Computers, laptops, mobile phones. So by going through we came to know to know that those gadgets contains very tiny little keys. Some people find it difficult to see the word on the keyboard and it is not ideal to use a full size keyboard onto the phones. The appearance of paper keyboard presented a newest replace to input on any portable device.

As compared with the other keyboards our system is more like to be paper based number board. We support user's enrolment in such a way that we design it according to their requirement for efficient typing and great accuracy. For now we just created a number pad that inputs the number on to the smart phone and clears it, but, in the future we will try to make it using qwerty.

II. KEYBOARD ARCHITECTURE

Our system consists of several components:

A paper with number keypad, A camera 5mp, Raspberry pi model 3b+, An acrylic stand, Bread board and led's, Power supply. The figure 1 shows the architecture of our system.

The Surface on which the keypad is printed should be a white surface and the background should also be white. That background should be free of objects and while working with the Image Processing this kind of back ground is must.

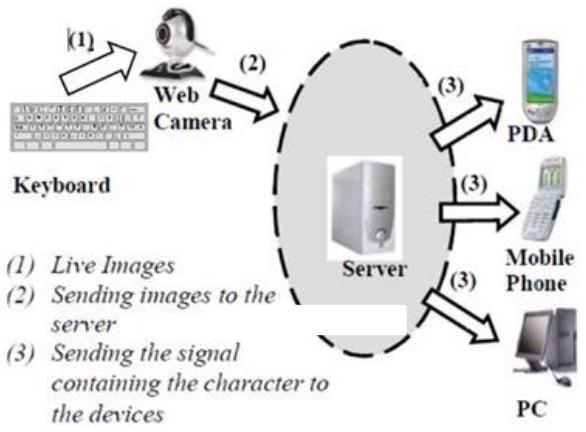

Figure 1. Design of System

The touching movement is improvised with a led light with red in color and the typed character is represented by a green led light. The typing movement on the paper board is captured and processed and sent to the server which we are using a firebase server in which user needs to login and after logging in you can type what ever you want. The camera captures the image frame one by one and processes it. First to spot the key locations the camera processes the initial image to identify the area of the contour and according to the array the second image is processed which leads to identifying the character. After this step the letter being identified by the raspberry-pi is uploaded to the hashmap which is in the firebase database. After the uploading is done then we start the application in phone and login. After Login the letter is typed directly onto the application. We choosed firebase beacause it is live database as soon as something happens it changes into the application.

A. Dividing the systems into Parts

The following diagram shows the full system which is divided into small sub systems like image capturing and processing it and character identification module which are described further:

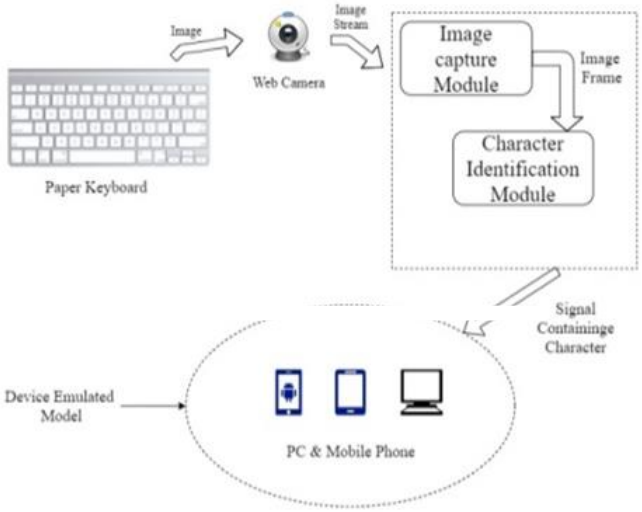

Figure 2. Division of system

B. Image Capturing Module:

This module is actually consisting of capturing the images of touching the paper board and storing them into the buffer like structure acting as queue. This modules come into play when a user has successfully decided which device he/she has to input the information in. The images are then coming out one by one to identify the character before giving the intitial frame rate for capturing of the buffer into the character identification module which is an aim to analyze it. After capturing the images they are stored as the backup copy which is again going for character identification once again. The following image shows the capturing of module:

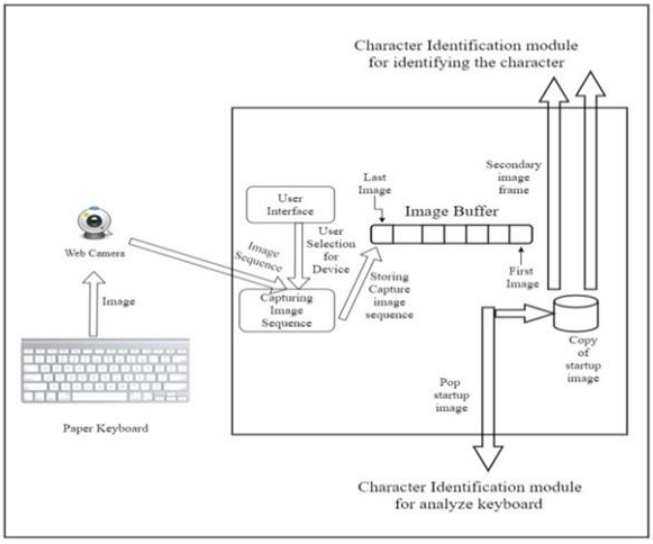

Figure 3 Image capturing sub systems

C. Character Identification module :

The images which were captured in the previous module are being sent to this module. That one image goes through many relevant techniques to determine the data which are essential and can be determined by various image processing techniques. The data are like:

1.) Analyzing the key locations on the paper
2.) Thresholding the background
3.) Creating contours
4.) Determining the character

After detection of edges which are nothing but contours form the first initial image the dimensions of the keyboard are being calculated. After that we use one method in OpenCV to determine the height and width of one each column in a row which apparently gives us the coordinates of each and every key. After each and every key being pressed it is hard to calculate the coordinates again and again. So the coordinates are stored in another file which is called at the starting of the camera to determine the area. After finding the blue pixel of the image, these coordinates are compared with the original array of coordinates to find the key block area and generate the letters.

D. Device Emulated Module:

Our server receives the implied code from the character identification module. The server accepts the signal from the raspberry-pi and forwards it to the keyboard driver. The application which is connected to the firebase server receives the signal from the driver. The user needs to login first into the application and then the given space is used for typing from the device driver. Thus this prototype simulates the keystroke by sending signal from raspberry-pi to the server using the firebase and device driver.

III. IMPLEMENTATION

The main operation of the proposed system is:

1.) the keyboard detected using edge detection.
2.) hand and finger tip detection using color segment.
3.) after detecting finger tip check for pressed key.
4.) obtain the press key.

A. Keyboard Detection:

Keyboard is identified using color differentiation. The border of keyboard is black, thus on thresholding, border of every key found.so location of keyboard is defined.

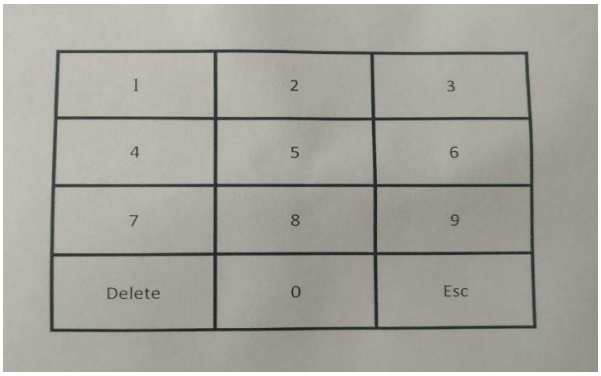

Figure 4 : Keyboard detection

B. Hand detection:

At first, a large number of hand images are collected. after observing those images the red component was higher than other component. So using image enhancement techniques we detected hand regions which are then thresholded to white and rest of image is black.

Figure 5 : Hand Detection using openCV

C. Edge Detection:

In this module, we use sobble technique, which is best for edge detection. Edge detection is use for detection of finger tip and we also use canny edge detection technique to complete the broken edges.

Figure 6 : Edge detection

The determination of those edges is one of the most difficult task because as our pi camera could capture about 640X480 pixel values. Since keeping above pixel values in mind we need to set the keyboard at an appropriate height which is approx. 32-35 cm. Only and only then we could capture an entire keyboard and as well as contours of it. However, you can use another external camera which has more resolution and much better fps than ours.

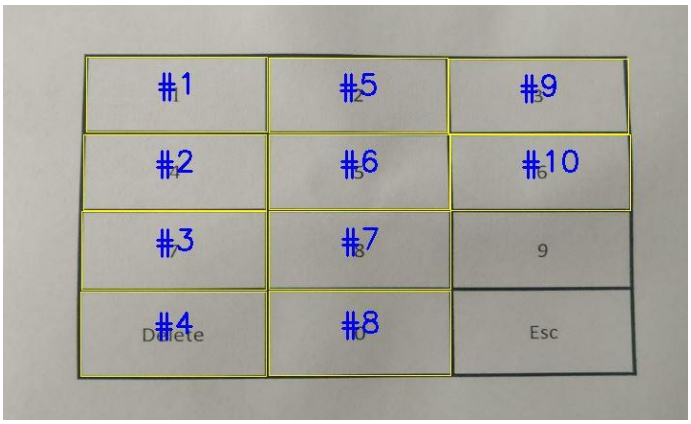

Figure 7 : Edges or contours of Keyboard

D. Touch Detection:

A small region around the finger is detected for shadows as shown. The figure shows that the tips which are discovered are white and the shadows are kept black through the thresholding as we want just the finger point. The white and black pixels ratio is determined, if the dimension of the ratios of non-shadowed area to the shadowed area exceeds a particular threshold we can determine that keys are being touched. We could use only trial and error method to determine the perfect threshold for this. The idea for touch detection is based on [11].

E. Firebase:

The main work for extraction of character is using above contours and edge detection techniques. Now comes the uploading or pushing the character value to database. Here, we are using Google's firebase because it has real time updation of data into any other device. It updates the data within couple of seconds. We have created a hash map for firebase which has name and value of the user. The name part is being fetched from phone in which user has logged in. The value comes from the raspberry pi which we have extracted using all the above techniques. So when the value part is being updated the phone's label box is also being updated constantly. When you login the first page is containing the on/off switch a profile picture and a label for the characters. The on/off switch allows users to change their status for retrieving the data from firebase.

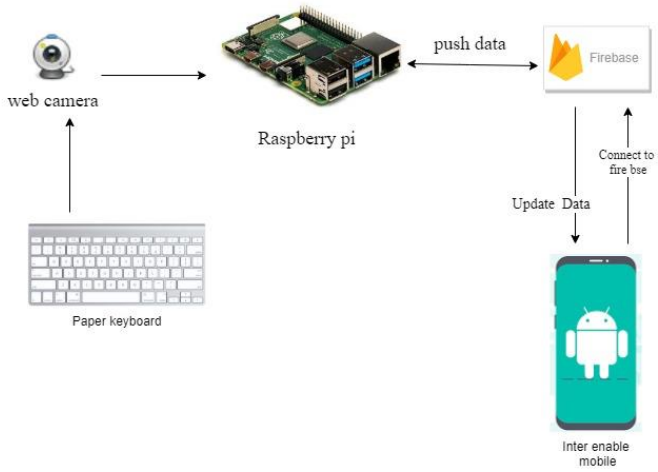

Figure 8 : Firebase connecting Raspberry

Figure 10 : The mobile application having the typed data "123"

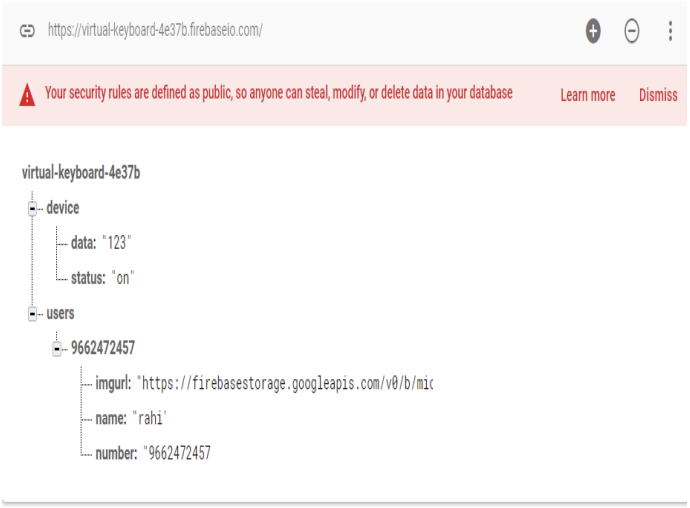

Figure 9 : Firebase structure

The above structure has two maps one is device and the other is users. The device has two fields data which is nothing but the characters form the raspberry-pi. Status is being fetched from user's phone for allowing or denying the update of characters. The users has only one field which is the phone number of user. The "imgurl" is the image url which is obtained by uploading an image to the firebase bucket and then that can return back a URL and that is the permanent URL for your phone to be determined by the database. Then any user can use this URL for any application for their purpose. Basically in our project it determines the profile picture that you have uploaded in our application. Now lets take look at mobile application:

IV. RESULTS

We are using a red circular pixel as threshold to determine the pressed key. In context with the red pixel to determine, the background needs to be a little brighter because as we are using with background and it is seems clear that the camera reduces noise in the image and helps improving the accuracy. Hence, more the light more the accuracy. Our project is the evidence that the time required to type one character on the application is a little low as per the physical keyboard due to the cameras resolution and the frame-rates (fps of camera is 10). So, if you need to have a better typing I suggest you to use a Logitech camera maybe 720p or 1080p with 60 fps. Apparently the time needed to type in one character was about 2 seconds/character. It should be kept in mind that this time varies each time and the minimum time is 32 milliseconds/character.

TABLE I

Actual result for typing character into keyboard

TESTED VALUE	PHYSICAL KEYBOARD	PAPER KEYBOARD
Time Taken	51.9 ms	470 ms

Failure	0	12
Time per character	0.37 s	2 s
Accuracy	100 %	90 %

V. CONCLUSION

As compared with the physical keyboard, paper keyboard has more advantages for own personalization, convenience and is applicable to favorable environment. The precision and accuracy of the keyboard have been tested thoroughly. Based on the above conclusion we have reached to an agreement that this prototype is easily usable and practical. This paper should help other to develop their own keyboard much efficient and smarter than ours. The diagrams will surely help for the faster development of smarter keyboards in computer/IT industry.

VI. REFERENCES

[1] 2018 IEEE International Symposium on Circuits and Systems (ISCAS) : proceedings : ISCAS 2018 : 27-30 May 2018, Florence, Italy. .

[2] IEEE INFOCOM 2016 : the 35th Annual IEEE International Conference on Computer Communications.

[3] Y. Adajania, J. Gosalia, A. Kanade, H. Mehta, and N. Shekokar, "Virtual keyboard using shadow analysis," in Proceedings - 3rd International Conference on Emerging Trends in Engineering and Technology, ICETET 2010, 2010, pp. 163–165.

[4] V. Agarwal, S. Mishra, M. Nehriya, and R. Satpute, "Virtual laser keyboard."

[5] R. G. Gupta, R. S. Reddy, and N. N. Parmar, "IRJET) e-2 B.E, Department of electronics engineering," Int. Res. J. Eng. Technol., 2009.

[6] M. Hagara and J. Pucik, Fingertip detection for virtual keyboard based on camera. 2013.

[7] M. Karray, M. Krid, and D. Sellami, A survey of AR systems and a case study of virtual keyboard based camera projector system. 2017.

[8] R. Kumar and P. Chaudhary, User defined custom virtual keyboard. 2017.

[9] C. Livada, M. Proleta, K. Romic, and H. Leventic, "Beyond the touch: A web camera based virtual keyboard," 2017.

[10] N. Matsui and Y. Yamamoto, "A New Input Method of Computers with One CCD Camera : virtual Keyboard."

[11] A. Syaichu-Rohman, Institute of Electrical and Electronics Engineers., Institute of Electrical and Electronics Engineers. Indonesia Section., Institute of Technological Bandung. School of Electrical Engineering and Informatics., and University Kebangsaan Malaysia. Fakulti Teknologi dan Sains Maklumat., Proceedings of the 2011 International Conference on Electrical Engineering and Informatics : 17-19 July 2011, Bandung, Indonesia. IEEE, 2011.

[12] Y. Zhang, W. Yan, and A. Narayanan, A virtual keyboard implementation based on finger recognition, vol. 2017-December. 2018.

[13] Q. Zhao, Virtual Keyboard : A human-computer interaction, no. 61471110. 2015.