A
Minor Project Report on
"VIRTUAL KEYBOARD"

In partial fulfillment of requirements for the degree of

Bachelor of Technology (B. Tech.)

in

Computer Science and Engineering

Submitted by
Ms. Yukta Dadhich(170418)

Under the Guidance of
Dr. Sanjeev Patwa

Computer Science and Engineering
SCHOOL OF ENGINEERING AND TECHNOLOGY
Mody University of Science and Technology
Lakshmangarh, Distt. Sikar-332311

December 2020

ACKNOWLEDGEMNT

I would like to thank everyone who has helped in every possible way in successful completion of this project. Firstly, I would like to express my gratitude to my mentor **Dr. Sanjeev Patwa** (Associate Professor, Department of Computer Science and Engineering) who have provided me an opportunity to do this project under his guidance. He has provided valuable ideas and support during the course of this work. This work would not have been possible without his support. I would like to thank **Dr. A. Senthil**(Head of Department,Computer Science and Engineering) for providing valuable input through the course of this work.

I would also like to thank my friends for their help in the completion of this project.

Yukta Dadhich

CERTIFICATE

This is to certify that the minor project report entitled "Virtual Keyboard" submitted by Yukta Dadhich, as a partial fulfillment for the requirement of B. Tech. VII Semester examination of the School of Engineering and Technology, Mody University of Science and Technology, Lakshmangarh for the academic session 2020-2021 is an original project work carried out under the supervision and guidance of Mr. Sanjeev Patwa has undergone the requisite duration as prescribed by the institution for the project work.

PROJECT GUIDE:
Approval Code:AUT_20_CSE_F10_80
Name: Dr. Sanjeev Patwa
Date: 27th Dec'2020

HEAD OF DEPARTMENT
Signature:
Name:Dr. A.Senthil
Date: 27th Dec'2020

EXAMINER-I:
Name: Dr. Sunil Jangir
Dept : CSE

EXAMINER-II
Name: Dr. Ajay Kumar Singh
Dept : CSE

ABSTRACT

Recent years have denoted a major increase inside the range of the path in which users can interact with PC's. The keyboard and mouse are the primary interfaces for a pc, clients at present use touchscreens, infrared cameras. In relation of those progressions and furthermore the multiplication of minor cameras in PC's and tablets, human pc interface analysts have examined the chance of executing a keyboard style interface using a camera as a substitute for genuine keyboard equipment.

A camera watches the client's hands, that lay on a flat surface. The camera may watch the hands from over the surface, or at partner degree point. The virtual keyboard's bundle investigations those photos in period to see the succession of keystrokes picked by the client.

In a few nations (for instance, India), clients communicate in different dialects, that makes fabricating physical keyboards for different languages expensive. A camera-based console will help for this problem, Smart-phone clients may at times wish to utilize a full- sized console with their gadget, but are reluctant to hold a physical console. Since most cell phones are provided with a camera, a camera-based keyboard may be an alternate in order to avoid this drawback.

The goal of this project was to actualize a virtual keyboard exploiting the picture inspection systems in order to create it for security purposes. We already have security lock systems present for domestic and commercial use. But the issue is that these systems are not secure enough as there can be breakage through key stroke logging, so in order to ensure more stealth and security we can replace the normal password typing keypads introduced in these systems by this virtual keyboard application which can be implemented using the regular security camera present in the security lock room, very less additions would be required to make the system work.

Inside the framework we tend to authorize, one high-quality camera which catches RGB photos of a user's hands, that bit a freckled surface, or console

tangle, in order to choose keystrokes and further using image processing techniques, these touch made by the user are monitored and output keystrokes are generated.

Table of Contents

Chapter 1 : Introduction

Virtual keyboard is one of the examples of innovation in the field of computer technology. These days computing is not only limited to the desktops but can also be seen on our mobil es. One thing that has remained constant since the beginning is the interface of the keyboar d i.e. QWERTY keyboard. Virtual keyboard is the new innovation in the field of computer t echnology.

The virtual keyboard makes use of camera/webcam and image processing techniques that al low user to work on any flat surface using virtual keyboard. Virtual keyboard allows us to c reate a keyboard in any of the preferred language on almost every existing platform. The pr operties of virtual keyboard being small and easy to use application makes it a good solutio n for text input across various different platforms.

The most important feature of the virtual keyboard is that it is very environmental friendly. It requires large amounts of resources to produce physical keyboards. When these physical keyboards become old or gets broken they are discarded by users and thus generate huge a mounts of waste, which is a burden on our environment. The virtual keyboard can be made on a plain paper, or it can be a projection keyboard on a desk or a laser keyboard on a wal. Thus, it can be said that the proposed virtual keyboard will not lead to any environmental p ollution.

Secondly, another advantage of the virtual keyboard is that users will be able to replace this keyb oard by themselves when they want a new one. It is simple for anyone to abandon the previous vi rtual keyboard and make a new customized one.

The concept of a virtual keyboard is that it has the same input and output functions as a traditional mechanical or physical keyboard but it is neither mechanical nor material. The different technolo gies which are used are digital image processing, pattern recognition, finger recognition, and

tracking. But right now there are a variety of challenges that needs to be overcome in order to dev elop a robust virtual keyboard.These challenges include various conditions such as different skin t ones between users, complex environments, inconvenient addon devices, and the limitations of va rious algorithms.

On comparing with existing studies of various other virtual keyboards, the virtual keyboard is visi ble and flexible. Users can still have the feeling and view of normal typing. Other advantage of th is keyboard is that users can replace the virtual keyboard by themselves when they want a new on e for any reason. It is easy for anyone to leave the previous paper keyboard and print a new custo mized one. For the paper keyboard, users can stick it on any plane surface.

1.1 Present System

A keyboard requires a lot of resources and is restricted by the physical features
that it has. Also, discarded keyboards also naturally contribute to environmental pollution. Cons equently, the touch screen is designed to replace the original physical keyboard and thus further reduces these flaws. However, the internal digital keyboard on the touch screen takes up a larg e amount of space, which also causes some content to be covered. The touch screen of system g ets dirtier by the touch of fingers and become worn over time. That is why it is necessary to de velop a new type of environment-friendly virtual keyboard with fewer flaws.

1.2 Proposed System

As the technological advancements develops, new human-
PC interfaces are authorized to supply various co-
operations among clients and machines. In this case, introduction for some human-to-
PC collaborations remains the old console/mouse. We are trying to exhibit here a cutting ed ge innovation, that will be that the Virtual Keyboard. Since the name tells it has no physical look. Virtual keyboard is an application that virtualizes equipment console with various dif ferent formats along these lines, that allows user to modify the design according to their ap plication. E.g. user will pick a different dialect for proofreader or pick a specific format for

bad habit applications. User can even style his/her very own design in equipment variant.

Virtual keyboard is nothing but one more advancement in field of personal computer innova tion. These days registering is not constrained to the work areas yet can be found in our mbiles. One thing that has not changed since the start is the console interface i.e. QWERTY keyboard.

The virtual keyboard innovation makes use of camera and picture handling (image processi ng) procedures allowing clients to use it at any level surface utilizing paper console. Virtual keyboard allows us to make a console in any of the favored dialect on almost every current stage. The properties of virtual console being little and simple to utilize application makes i t a answer for content contribution over various stages.

Virtual keyboard is better than physical keyboard as physical keyboards are unit expansive and serve little or no improvement while virtual keyboard are smaller in size and are easy to use. Main features of the virtual keyboard are: security lock systems are the prime applicati on, it supports numerous content dialects, is small in size, easy to use.

Chapter 2: System Design

2.1 Introduction

As the technological advancements develop, new human-
PC interfaces are authorized to supply various co-
operations among clients and machines. In this case, introduction for some human-to-PC collaborations remains the old console/mouse. We are trying to exhibit here a cutting ed ge innovation, that will be that the Virtual Keyboard. Since the name tells it has no physical look. Virtual keyboard is an application that virtualizes equipment console with various dif ferent formats along these lines that allows user to modify the design according to their appl ication. E.g. user will pick a different dialect for proofreader or pick a specific format for ba d habit applications. User can even style his/her very own design in equipment variant.

Virtual keyboard is nothing but one more advancement in field of personal computer innova tion. These days registering is not constrained to the work areas yet can be found in our mo biles. One thing that has not changed since the start is the console interface i.e. QWERTY k eyboard.

The virtual keyboard innovation makes use of camera and picture handling (image processi ng) procedures allowing clients to use it at any level surface utilizing paper console. Virtual keyboard allows us to make a console in any of the favored dialect on almost every current stage. The properties of virtual console being little and simple to utilize application makes i t a answer for content contribution over various stages.

Virtual keyboard is better than physical keyboard as physical keyboards are unit expansive and serve little or no improvement while virtual keyboard are smaller in size and are easy to use. Main features of the virtual keyboard are: security lock systems is the prime applicatin , it supports numerous content dialects, is small in size, easy to use.

2.2 Image processing techniques

2.2.1 Threshold

Threshold is a method to concentrate only on the necessary information and ignoring the un necessary details. It is a non-
direct activity that changes a gray scale picture into a binary picture where the two dimensi ons are assigned out to pixels that are beneath or over the predetermined threshold esteem. For thresholding, typically a mean 8-
bit value is taken from the original image. Now, the original image is divided into two porti ons. Pixel esteems that are not exactly or equivalent to the edge; foundation. Pixels esteem i s more prominent than the edge; frontal area.

2.2.2 Segmentation

Picture segmentation is the way toward dividing a computerized picture into numerous secti ons (sets of pixels). Picture segmentation is ordinarily used to find objects and their limits i n pictures. A compelling way to deal with performing picture division incorporates utilizing calculations, apparatuses, and a far reaching condition for information investigation, repres entation, and calculation advancement.
The objective of segmentation is to improve as well as change the portrayal of an image int o something that is increasingly important and simpler to dissect. The consequence of pictur e segmentation is a lot of fragments that all things considered spread the whole picture, or a lot of shapes removed from the picture. Every one of the pixel in a locales are comparable as for some trademark or processed property, for example, color, intensity, or texture nearb y locales are all together extra ordinary as for similar trademark.

2.2.3 Image Enhancement

Image enhancement means enhancing and emphasizing an image's features such as edges, c ontrast, sharpness and so on. Therefore, changing the dynamic ranges of these feature impro ves the graphic effect of the image and help in extraction of important information easily.

There are two methods for enhancing an image:

1) Spatial domain method: spatial domain refers to the plane of the image and is based on d irect processing of the image pixel.
2) Frequency domain method: processing technologies like Fourier transform, wavelet trans form are based on modifying frequencies of transformed images.

Image enhancement generally stretches the gray value range of an image i.e. 0-255 which helps in boosting the contrast of an image along with its quality. The concept of image enhancement is generally based on histogram.

2.2.4 Image Binarization

Image binarization is the conversion of image from RGB to binary image. It plays an impor tant role in digital image processing as most of the image processing systems are based on b inary images.
Firstly, the image will be transformed into gray scale image than we apply threshold to it. T his threshold might be fixed or adaptive. Adaptive image binarization is required wherever an optimal threshold is selected for the image.
Conversion of colored images to gray scale generally uses 3 methods.

1) Maximum method:

The RGB value of each pixel is equal to maximum of its RGB values.

2) Averaging method:

The RGB value of each pixel is equal to average of its RGB values.

3) Weighted averaging method:

The RGB value of each pixel is equal to average of the weighted RGB values of every pixel.

2.2.5 Image Smoothing

There are lot of noises which are present in an image ,in orcer to makeit smooth we need to remove them. But if it is not done properly then it can even makeit more blur.So we need to u se proper methods.It could be done in spatial and frequency domain:

1) Methods used in sspatial: neighborhood averaging, median filtering, multiple images ave ragingetc.
2) Methods used in frequency domain: ideal low pass filter, butter worth low pass filter.

2.2.6 Pre- Processing

The image obtained on the camera is not that clear. To make clear and usable various functi ons are applied on input image. It can be made blur or sharp accordingly. It can also be mad e blur if it is very sharp or sharp if it very blur.

2.2.7 Selective Red,Green,Blue

Depending on the color components (Red, Green and Blue values)we filter the pixels of the image. Threshold is decided in prior for these. The range is decided based on these pixels.

2.2.8 (Red,Green,Blue) to (Hue ,Saturation, Value) Conversion

The model stands Hue, Saturation and value. Color type is presented by hue. Saturation tell s how pure the color is. Intensity scale represents how bright the color is.

2.3 Algorithm

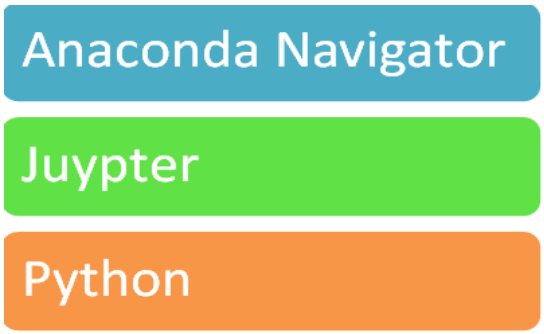
1) A keyboard is created on pur screens in blue color.And the movement of the pen is seen t hrough webcam.
2) Video made while typing is also captured through webcam.
3) Movement of pen while pressing different keys is captured at regular intervals by camera .

4) For clear capturing we do some preprocessing on images.For this we can either make the picture sharp or blur.

5) At the end the characters typed by the user can be seen at real time

Chapter 3: Hardware and Software Details

3.1 Tools Used



Anaconda Navigator is a graphical user interface whichisincluded in Anaconda distribution which allows us to launch applications and easily managevariousconda packages, environments, and channels without us ing commandline commands. Navigator/User can search for packages on Anaconda.org or in a local Anaco nda Repository. It is available for Windows, macOS, and Linux.

Many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different vers ions.

The commandline program conda is both a package manager and an environment manager helps data scientists ensure that each version of each package has all the dependencies it requires andworks correctly.

.

Various applications are available by default in Navigator:

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- Spyder
- PyCharm
- VSCode

- Glueviz
- Orange 3 App
- RStudio
- Anaconda Prompt (Windows only)
- Anaconda PowerShell (Windows only)

Advanced conda users can also build their own Navigator applications.
The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and exec ute your code.

We can also use Jupyter Notebooks the same way. Jupyter Notebooks is becoming popular syste m which combines our code, descriptive text, output, images, in a single notebook file which is ed ited, viewed, and used in a web browser.

The Jupyter Notebook

The Jupyter Notebook is an open-
source web application that allows users to create and share documents that contain live code, equ ations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Chapter 4: Implementation Work Details

4.1 Working

The webcam/camera on our laptop will capture the live video feedback of the user with the help of the keyboard and it will pass that feedback in the code and then it will pre-process the picture i.e. either blur or sharpen it according to the need. Then it will apply diff erent processing strategies, for example, threshold, and segmentation and will produce an o utput.

Python programming language will be used for preparing photo. Further steps like convertin g image into binary form, applying edge detection algorithms will be applied later.

Steps followed during image processing are:

1) Detection of Keyboard:

Color differentiation is used for the detection of corner points of the keyboard. The endpoin ts are black in color, these points could be recognized as we threshold the given image. The n the virtual keyboard's location can be detected.

2) Detection of Hand:

In the beginning a large collection of hand pictures is being made. Then according to the us ers hand pictures of are observed. These observations are same for all the pictures that we v erify. There is no effect of hands and veins on the detection on hands or pen on camera. But we can also use various techniques for improvement in detection. At the the hand comes w hite and remaining part comes in black.

4.2 Applications

1) It can be used for computer machines:

The virtual keyboard applications can work in normal computer. Users can perform various operations like writing of some data or entering passwords for private data by using this app

lication.And we do not need to worry about problems like keystoke.

2) It can be used in automated transfer machines:

The user can use this application for entering passwords in automated transfer machines. Fo r this we can use the camera which is installed in the room of the machine and user will be able to type numbers in real time.

3) Protection

This application protects the password of the users from various malware practices. It also makes our password safe and it protects it from theft.

Chapter 5: Source Code

```
import cv2
import numpy as np
cap=cv2.VideoCapture(0)
word=[]
value=[]
while True:
    ret,keyboard=cap.read()
    keyboard=cv2.resize(keyboard,(1500,1000))
    keyboard1=np.zeros((100,500,3),np.uint8)
    h,w=100,100
    th=3

    q=[]
    v=[]
    dic={}
    font_letter=cv2.FONT_HERSHEY_PLAIN

    import string
```

```
a=string.ascii_uppercase[:26]
a=a.split()
for i in range(26):
    v.append(a[0][i])

for a in range(0,1000,100):
    q.append(a)

# To make keyboard outlines
y=-100
for j in range(3):
    x=0
    y+=100
    for i in range(10):
        cv2.rectangle(keyboard,(x+th,y+th),(x+w-th,y+h-th),(255,0,0),th)
        x+=100


x=0
for i in range(6):
```

```
    cv2.rectangle(keyboard,(x+th,300+th),(x+w-th,300+h-th),(255,0,0),th)
    x+=100
    cv2.putText(keyboard,v[i+20],(20+q[i],380),font_letter,5,(255,0,0),4)
    dic[v[i+20]]=[20+q[i],380]
# to enter Text of Keyboard
for k in range(10):
    font_letter=cv2.FONT_HERSHEY_PLAIN
    cv2.putText(keyboard,str(k),(20+q[k],80),font_letter,5,(255,0,0),4)
    dic[k]=[20+q[k],80]
    cv2.putText(keyboard,v[k],(20+q[k],180),font_letter,5,(255,0,0),4)
    dic[v[k]]=[20+q[k],180]
    cv2.putText(keyboard,v[k+10],(20+q[k],280),font_letter,5,(255,0,0),4)
    dic[v[k+10]]=[20+q[k],280]

op=["/","*","-","+",".",",","%","^","@"]

y=-100
for j in range(3):
    y+=100
    x=1050
```

```
        for i in range(3):
            cv2.rectangle(keyboard,(x+th,y+th),(x+w-th,y+h-th),(255,0,0),th)
            x+=100

    for i in range(3):
        cv2.putText(keyboard,op[i],(1080+q[i],80),font_letter,5,(255,0,0),4)
        dic[op[i]]=[1080+q[i],80]

cv2.putText(keyboard,op[i+3],(1080+q[i],180),font_letter,5,(255,0,0),4)
        dic[op[i+3]]=[1080+q[i],180]

cv2.putText(keyboard,op[i+6],(1080+q[i],280),font_letter,5,(255,0,0),4)
        dic[op[i+6]]=[1080+q[i],280]

    # to find key from values in a dictionary.

    def getKeysByValue(dictOfElements, valueToFind):
        listOfKeys = list()
        listOfItems = dictOfElements.items()
        for item  in listOfItems:
```

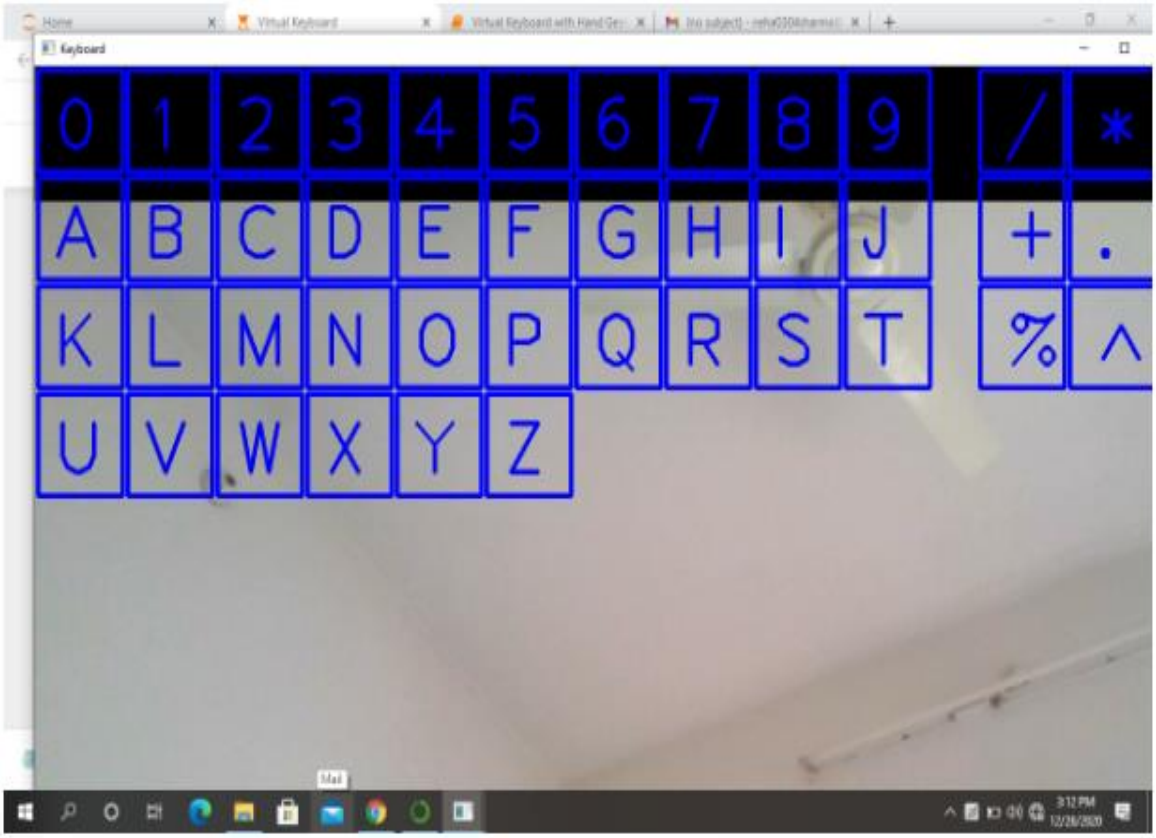
```
        if item[1] == valueToFind:
            listOfKeys.append(item[0])
    return  listOfKeys
# to get the coordinates of mouse click.
def mouse_drawing(g,h):

        #print("Left Click")
        #print(g,h)
        for a,b in dic.values():
            if (g>=a and g<=a+100) and (h>=b and h<=b+100):
                listOfKeys = getKeysByValue(dic, [a,b+100])
                keyboard1=np.zeros((1000,1200,3),np.uint8)

                for key  in listOfKeys:
                    ad=key
                        #print(key,end="")
        return(ad)
hsv = cv2.cvtColor(keyboard, cv2.COLOR_BGR2HSV)

# define range of red color in HSV
```

```
lower_red = np.array([0, 120, 70])
uper_red   = np.array([10,255,255])
# Threshold the HSV image to get only red colors
mask = cv2.inRange(hsv, lower_red,uper_red)

# Bitwise-AND mask and original image
res = cv2.bitwise_and(keyboard,keyboard, mask= mask)
frame=cv2.flip(keyboard,1)
frame=cv2.resize(keyboard,(1300,700))

points=cv2.findNonZero(mask)

try:
  # print("s")
   if points[0][0][0]!=None:
       #print("A")

       g=points[0][0][0]
       #print(g)
       h=points[0][0][1]
```

```
        #print(h)


        a=mouse_drawing(g,h)


        try:
           if a!=None:
              word.append(a)
              if len(word)==30:
                 value.append(a)


cv2.putText(keyboard1,a,(12,20),cv2.FONT_HERSHEY_COMPLEX,1,(0,
0,255),3)
                 print(a,end="")


                 word=[]
        except:
           pass


    except:
```

```
        pass


    cv2.namedWindow("Keyboard")
    cv2.imshow("key",keyboard1)
    cv2.imshow("Keyboard",keyboard)
    if cv2.waitKey(10) & 0xFF==27:
        cv2.destroyAllWindows()
        break

cap.release()
cv2.destroyAllWindows()
```

Chapter 6: Outputs

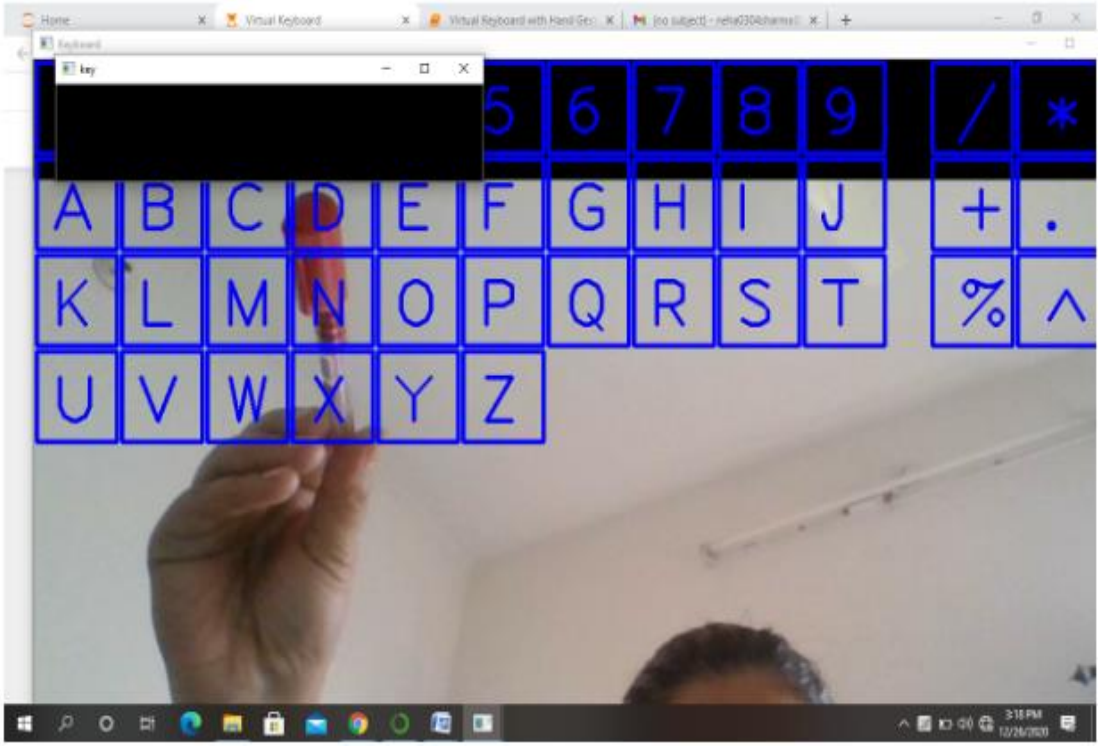


```
    cv2.imshow( key ,keyboard)
    cv2.imshow("Keyboard",keyboard)
    if cv2.waitKey(10) & 0xFF==27:
        cv2.destroyAllWindows()
        break

cap.release()
cv2.destroyAllWindows()
```

```
NXNBYYXXYYYZZYZYXZZLWXTKUWWWXXVVVVWWVWDKKLLYZMXZZTVWY
```

In [2]:
```
cap.release()
```

In [ ]:
```

```



Chapter 7: System Testing

In this project, keyboard layout can be seen on the camera window in blue color.

The images are preprocessed before the webcam gets started and it either sharpens the imag e or blurs it accordingly.

Than the pen used by the user are detected as soon as there is a contact made with any of t he characters in the keyboard.

A blink is made every time we click on any key of the keyboard on another window and th en the keys pressed by the user are obtained as the output.

Chapter 8: Conclusion

So it can be concluded that the virtual keyboard application is not 100% efficient as there m ight be some errors occurring due to low margins between different characters because of w hich more than one or a different character can get typed.

We are able to access the keyboard of blue color using camera on our screen. Also we are a ble to locate the fingertip position in the paper keyboard and whether the touch has been ma de or not and the output is generated accordingly.

This report principally shows the practical execution of the virtual keyboard that shows the future generation of human mobile devices as well as the human computer interaction in the creation of virtual world. We are aware that the mobile devices are more in demand now, which requires an easy system for typing. The properties such as lightness and flexibility of the keyboard are more suitable for such devices. It provides users the ability to compress th e data. Users can change the keyboard according to his/her needs which would require very little changes in the program. It can be concluded that virtual keyboard application would m ake typing more easier, secure and reliable for the user.

8.1 Future Scope

For future work we will focus on improving the performance of this virtual keyboard. The futur e improvements are as follows: 1) the virtual keyboard can work in real time, which means this virtual keyboard can be taken in practice; 2) it could also identify two different hands typing si multaneously so that it can get utilized in computers; 3) it can also work in different backgro unds or dim light environments; 4) other function keys such as enter, delete, space, shift, etc., ca n be added on the virtual keyboard; 5)the difference in participants should be considered in the f uture.To get more efficiency people of all ages can be called. More and more people can be inv ited to take part in the experiments, which will give more advice and feedback.

In addition, if some other machine learning technologies like *k*-
map neural network, artificial neural network can be used in this virtual keyboard, the recogniti

on performance might get improved. It is believed that artificial neural networks offer significa nt advantages for dynamic gesture recognition and character recognition.

3 dimensional projection technologies might get known and cheaper as compared to others. At t hat time, the virtual keyboard can combine with 3 dimensional projection technologies to projec t a virtual keyboard.

8.2 Limitations

1) Key size creates error:

As the key sizes are constrained also a little margin between the two keys are present so, th ere is a chance that a user may accidentally touch two keys at the same time or the adjacen t key might get touched which can change the output. Also it leads to slow typing speed.

2) Predetermined number of keys:

As the virtual keyboard is software based application and has very precise key layout which is already known to the program so adding any more symbol or a character will require the program to be changed accordingly the layout.

3) Software so an update can break it:

As virtual keyboard is software based application so, any virus or malware material within t he system can seriously damage the application.

Chapter 9: Bibliography

1. Kölsch, M. and Turk, M. “Keyboards without Keyboards: Survey of Virtual Keyboards”, Workshop on Sensing 0.and for Media-centric Systems, Santa Barbara, CA, June 20-21, 2002.

2. H. Du, T. Oggier, F. Lustenberger and E. Charbon, “A virtual keyboard based on true3D optical ranging,” Proc. British Machine Vision Conference (BMVC), Oxford, pp. 220-229, Sept. 2005.

3. Matsui, N. & Yamamoto, Y. A New Input Method of Computers with One CCD Camera: Virtual Keyboard. Keio University, 2000.

1. Wijewantha N. S.,” Vista Key: A Keyboard Without A Keyboard – A Type Of Virtual Keyboard,” Final year project hesis, Informatics Institute of Technology, Wellawatta, Sri Lanka, April 2004.

2. Adajania, Y., Gosalia, J., Kanade, A., Mehta, H. and Shekokar, N., 2010, November. Virtual keyboard using shadow analysis. In *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on* (pp. 163-165). IEEE.

3. Yin, Y., Li, Q., Xie, L., Yi, S., Novak, E. and Lu, S., 2018. CamK: Camera-Based Keystroke Detection and Localization for Small Mobile Devices. *IEEE Transactions on Mobile Computing*, *17*(10), pp.2236-2251

Chapter 10 : Annexures

URKUND

Document Information

Analyzed document	Minor Project Report (1).pdf (D90633079)
Submitted	12/27/2020 2:09:00 PM
Submitted by	Dr Sanjeev Patwa
Submitter email	sanjeevpatwa.cet@modyuniversity.ac.in
Similarity	0%
Analysis address	sanjeevpatwa.cet.modyun@analysis.urkund.com

Sources included in the report