Virtual keyboard Explanation
FALL SEMESTER 2023

Author : Riteshkumar Kandane

04 December 2023

Methodology:

“My purpose was to to write on a text file from distance”

I used a media pipe and open cv to detect hands and a list containing frequently changing position of nodes of hand. Then I made a class button whose instance is defined by position on image, keyword(to be printed on screen), and size. It also contains the recOfKey() function which returns the coordinates of the rectangle including a button on image. Then I made separate lists of Alphanumeric buttons and special buttons(i.e enter, caps lock, backspace). Then I draw all the buttons on the image.

Next task was to check which button was pressed and the respective action to be accomplished in code. For the Alphanumeric button I just need to add the respective key of the button in text. Then in the GUI window I used two buttons and a text widget. Open button opens a text file into text widget, while save button saves file and clear text widget.

CLASSES:

HandTrackingModule:

This class is used to detect hands and to get a list containing frequently changing position of nodes of hand. It also draws nodes of hands and lines joining the nodes.

button class:

Instance of this class is defined by position on image, keyword(to be printed on screen), and size. It also contains the recOfKey() function which returns the coordinates of the rectangle including a button on image.

```
def __init__(self, pos, text, size=[85,85]) :     #need to add double underscores
    self.pos = pos
    self.size = size
    self.text = text
  def recOfKey(self):
    x ,y = self.pos
    w, h = self.size
    return [self.pos,[x+w,(y+h)]]
```

FUNCTIONS:

IsInRectangle:

This Function checks a point. if a point lies in the rectangle then it returns true and if not then returns false.

```
if (rec[1][0]>point[0]>rec[0][0] and rec[1][1]>point[1]>rec[0][1]):
    return True
  else:
    return False
```

IsPressed:

This function checks the coordinates of fingertip and thumb tips. If these both points are in the rectangle of any button then it returns true in other cases it returns false.

```
if isInRectangle(b.recOfKey(),lmList1[8][:2]) and
isInRectangle(b.recOfKey(),lmList1[4][:2]):
    return True
else:
    return False
```

Unpressed:

This function is reciprocal of is Pressed(). It is to avoid repetition of pressed keys.

```
if isInRectangle(b.recOfKey(),lmList1[8][:2]) and
isInRectangle(b.recOfKey(),lmList1[4][:2]):
    return False
else:
   return True
```

DrawAll:

This function draws all alphanumeric buttons on an image and returns an updated image.

```
for button in buttonList:
    x, y = button.pos
    w, h = button.size
    cv2.rectangle(img,button.pos, (x + w,(y + h)), (255, 255, 0),2)
    cv2.putText(img, button.text, (x + 20,(y + 65)),
            cv2.FONT_HERSHEY_PLAIN, 4, (255, 0, 255), 4)
return img
```

DrawSpecial:

This function draws all Special buttons on the image and returns the updated image. It was necessary to draw special buttons separately because they have different font sizes and keys.

```
for button in spacial_keys:
    x, y = button.pos
    w, h = button.size
    cornerRect(img, (button.pos[0], button.pos[1],
            button.size[0], button.size[0]), 20, rt=0)
    cv2.rectangle(img, button.pos, (x + w, (y + h)), (255, 204, 255), 2)
    cv2.putText(img, button.text, (x + 20, (y + 65)),
        cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 255), 4)
return img
```

OpenFile:

This function opens a text file from file dialog in reading mode. Then it puts the opened file's text into a text widget.

```
global newtext
  newtext = ""
  filepath = filedialog.askopenfilename(initialdir="D:\\programs\\4th_year_project\\Virtual Keyboard Document",
                                        title="Open file okay?",
                                        filetypes= (("text files","*.txt"),
                                        ("all files","*.*")))
    file = open(filepath,'r')
    newtext=file.read()
    clip.delete("1.0","end")
    clip.insert(END,newtext)
    file.close()
```

Save:

This function opens a text file from file dialog in append mode. Then it puts the text widget's text into the opened file. IT also clears text widget (clip in my case).

```
def save(newtext):
    try:
        filepath =
filedialog.askopenfilename(initialdir="D:\\programs\\4th_year_project\\Virtua
l Keyboard Document",
                                   title="Open file okay?",
                                   filetypes= (("text files","*.txt"),
                                   ("all files","*.*")))
        file = open(filepath,'a')
        file.write('\n text added')
        file.write(newtext)
        file.close()
    except Exception:
        print("Could not write to file")
        newtext=''
    clip.delete("1.0","end")
```

GUI:

My project's GUI window contains five Buttons widget(File, Save, Exit, Live, Browse), one Text Widget(i.e clip that can be scrolled vertically) and a video label.

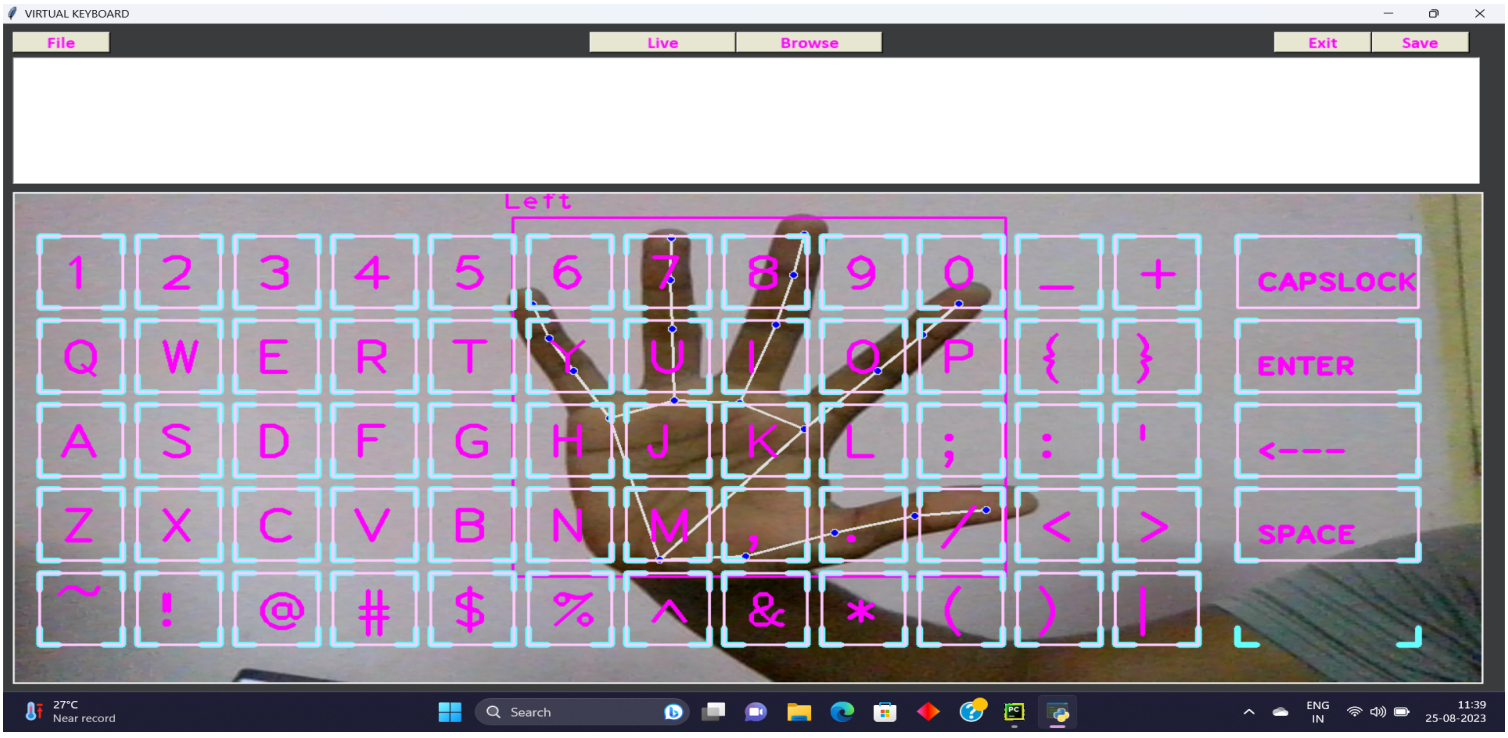
File: It opens a text file in the text widget to edit.

Save: It is to save a text file, wherever you want to save.

Live: It is to shift the webcam to live video.

Browse: It is to shift video feed to saved video selected from file dialog.

Exit: It is to exit the whole program.



Main Program:

In the main function I defined a window of tkinter. Then I added OPEN and SAVE buttons on two top extremes placed by hit and trial method. Both buttons are associated with OpenFile and Save functions respectively. Below these buttons there is a text widget. It can be scrolled vertically.

While True:

This loop is continuously reading images from the webcam then drawing all buttons on it then checks pressed buttons and takes action accordingly. It also checks whether the button is unpressed or still pressed. Incase if button is kept on pressed it will not type. To unpress button , bring out your finger or thumb tip from the rectangle of Keys.

It changes clip(text widget) and ‘newtext’(variable string) accordingly. Finally it forms an array of images and puts it into a video label. Then updates the window.