Gaze Controller Keyboard
Computer Vision

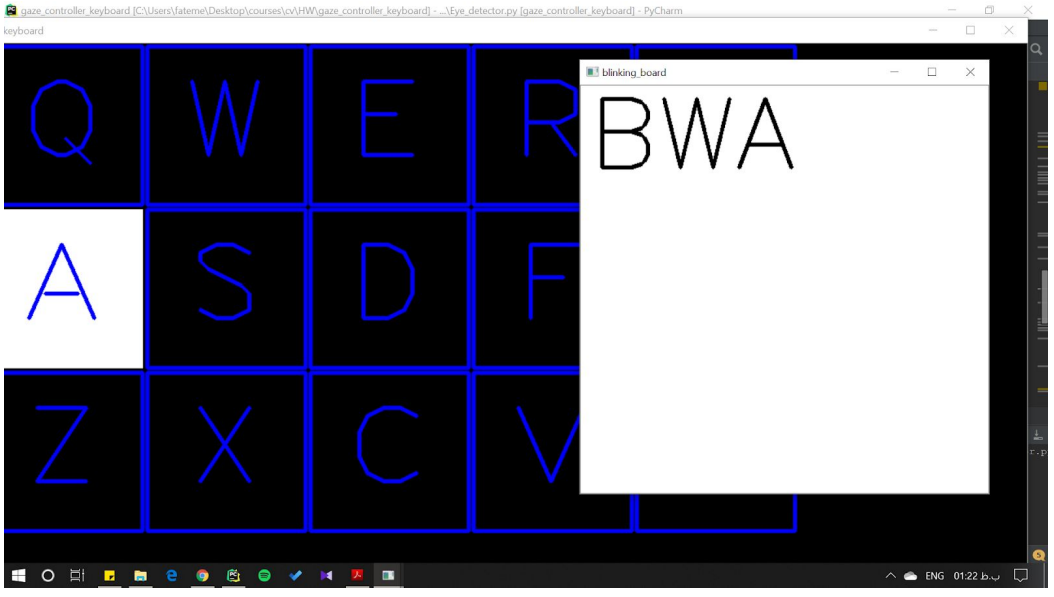
- First of all the polygon of the left eye is taken from the frame
  After putting a threshold on the eye we have the eyeball black and the rest is white
  With this black and white portions, we can compare the portions and find out at what point the person is looking at.
  While we have the thresholded of the eye (grayscale) we can simply count the white pixels.

- I took the ½ part of the eye (left and right part) and then counted the 1(white) pixels to see the sclera position in the eye. By getting the ratio of left side white pixels to the right side we can see which part the eyeball is looking at. Checking one eye (left for an example) is enough but checking 2 would be more precise.
  The gaze ratio is the ratio of left_side_white to the right side.
  If gaze_ratio is smaller than 1 we are looking at the right side and if it is greater than 5.5 we are looking at the left side, otherwise we are looking at the center.

- The next step is creating a virtual keyboard: it is in the letter function, inside virtuar_keyvoard.py

- Then we run a frame counter to show and move on letters for each 10 frames

- By blinking for 5 frames a letter is displayed on blinking board
  It is simply done by defining a blinking counter that is increased each time a blink is detected and if the count is equal to 5 the letter will be writing on text to be displayed on the whiteboard.
  Each time the blinking is increasing, the frames count will not increase because we what the letter to stop at that place and not to move forward so as to keep track of it.



- Final Part + Wrap up the previous parts

  frame_to_blink= 6 -> it means that if we keep our eyes closed for 6 frames then we can consider that the key was pressed.
  Frame_active_letter = 9 it means that every 9 frames the next letter with light up.

  The eye_countor_points function draws the red or green line around the eyes
  In fact, it returns the points of eyes and we draw them with
  cv2.polylines()
  At the #eye color part, the lines over the left eye and right eye are colored to red

  Then we detect the blinking of the eyes #Detect blinking with calculating the ratio as it is explained earlier.

  Select_keyboard_menu is true if we want to show the first left and right menu
  Keyboard_selectin_frame counts the number of frames for choosing keyboards
  Rows and cols keep the shape of the frame

  If the keyboard selected is left then it means we want the left part of the keyboard and to use “key_set_1” otherwise key_set_2 will be chosen.

  If the menu is open (True) then we draw the menu with draw_menu()

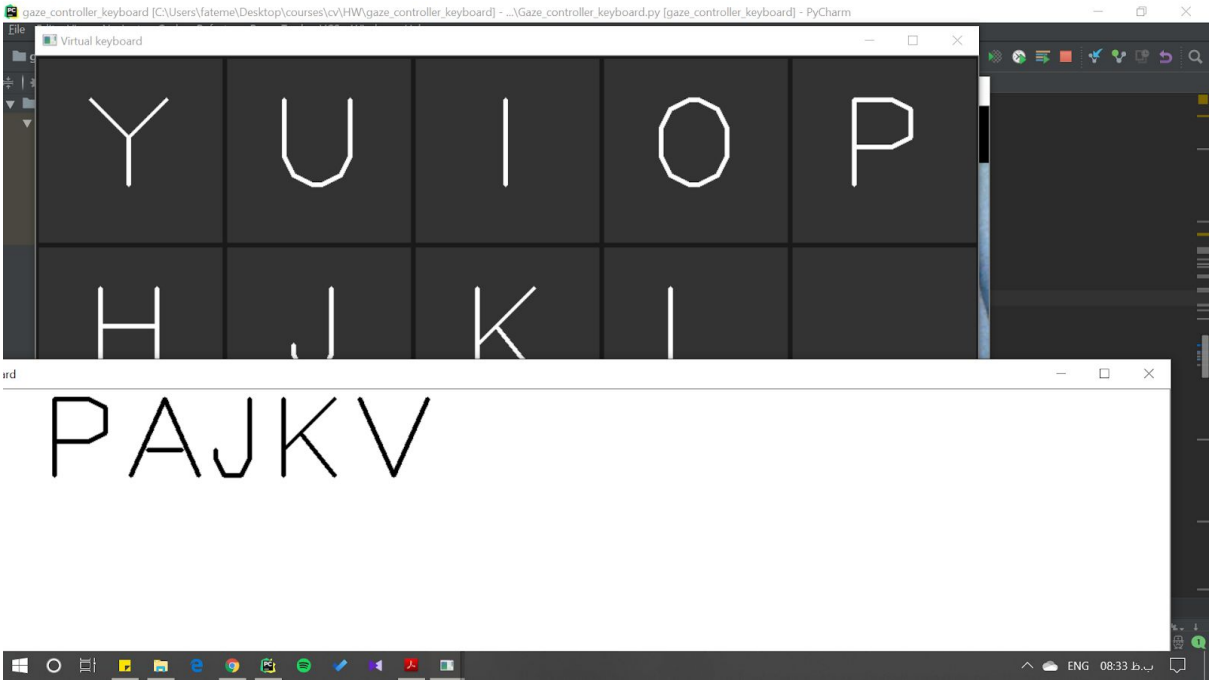
  If the menu is True then we calculate the gaze of the right and left eye as explained before.

  Then if the ratio is below 0.9 it is looking at right we count on the frames on right and if it was 15 then the menu is set to False.

  At each loop according to keyboard_selected which is left or right the keyboard is sent and when the menu is false it is displayed -> each frame is ++ every time and if it gets to 9 it will go to the next.

  Also when the menu is False we detect the blink -> when the ration is more than 5 it means we are blinking
  The lines around eyes turn to red, blinking frames will increase and the frames will decrease one.
  If blinking frames are equal to 6 then the letter will be chosen by adding the current active letter which is updated each frame to the text.
  And the menu is set to True.

  In the end, the text is displayed on the board and images are displayed.


gaze_controller_keyboard [C:\Users\fateme\Desktop\courses\cv\HW\gaze_controller_keyboard] - ...\Gaze_controller_keyboard.py [gaze_controller_keyboard] - PyCharm
File
Virtual keyboard
Y U I O P
H J K L
ard
PAJKV
ENG 08:33 ب.ظ