Swanirman Sunirmit Publications of Research -Special Issue ICRTTEAS January 2024 | [2024-25]
ISSN [Online]: 2583-2654

Eye Movement Controlled Keyboard using AI

[1]Dr K M Palaniswamy, [2]Deepak R, [3]Hari Shakthi N, [4]Shyam Kumar P
Department of Electronics and Communication Engineering, Dr T Thimmaiah Institute of Technology

Abstract: Human-computer interaction is an emerging field that includes various algorithms and specific types of strategies to optimize interaction patterns. Optical technology is one of the greatest techniques of modern science and is used for storing, recording, writing, etc. can be used in as many fields as the need for physical equipment arises, causing many researchers to develop the most useful system with heart and light. We present the development of a digital keyboard adapted to the shape of the eye and the definition of the light of the eye. That means building a device that takes video directly from a digital camera and detects people's faces and eyes. We can follow simple rules to keep the face correct, because the eyes and lips are always in the sample area to facilitate the eye detection method. To test this, we will use a method that includes 68 precise facial points, and the focus area must be on each face, such as the top of the chin, eyebrows, nose, and the outer edge of the face can be called a symptom. In addition, the right blinks and eyes determine the left and right eyes to select keyboard elements to select the desired key from the number keys on the keyboard. The idea of this gadget is to make love without using fingers or hands. Such use is necessary for people who have lost control of their limbs. For each stage of the device, the method is explained, consisting of a wave diagram, and then the results of the implementation are determined.

I INTRODUCTION

Several biometric systems, such as face recognition, eye recognition, and iris recognition, have received considerable attention in the field of video analytics in recent years. These systems are very complex due to important dependencies such as object imaging, orientation, lighting, and camera resolution. However, it is a widely used technology in various applications such as security. The human eye is one of the most important and distinctive parts of the face and provides unique information about humanity. "Eye recognition" is now another option to create a promising set. Many useful interfaces can be created with eye detection and eye movement tracking. Due to the rapid advancement of technology, there is a huge demand for human-computer interaction. Many valuable systems have been created to make people's lives safer and easier. However, he had to create such a system for those who could not work voluntarily and could only act unconsciously. For most people with disabilities, blinking is the only voluntary action. That's why eye-based human interaction systems were created. Eye tracking system uses various image processing techniques such as image processing to capture input data and convert it into digital form, filtering based on eye biometrics, sharpening etc. In digital imaging, various mathematical operations are used to obtain more sophisticated images to perform the following operations. There are many systems and applications based on human eye tracking. There are many types of human-computer interfaces that use human eye movements and gestures. Some interfaces use eye movements to control the mouse cursor, while others track the eyes to check if the driver is drowsy while driving. Many eye tracking methods are used in medicine and optometry to diagnose various diseases. Virtual keyboards and eye tracking are examples of modern technology.

II LITERATURE SURVEY

Choi et al [1] proposed a five-layer CNN-based method for driver gaze region classification and head posture estimation. They created a dataset consisting of various images of male and female drivers, including drivers wearing glasses. Based on the dataset, we built a CNN model that can classify the

ISSN [Online]: 2583-2654

driver's 9 gaze regions and estimate the head position. Each view area represents a different area inside the car, such as left mirror, right mirror, rear view mirror, steering wheel, gear, center, left windshield, right windshield. A version of the AlexNet architecture was used in this study, and experimental results showed that this technique produces an accurate gaze region detection classification accuracy of 95%. The design of the personalized driver's view area classification system has been significantly improved.However, generalized systems design still lags behind. A generalized system should be able to classify gaze regions for different subjects and different viewpoints.

Vora et al [2] took a step forward by designing a generalized CNN-based method for gaze detection. This technique consists of two units: a pre- treatment unit and a fine-tuning unit. The preprocessing unit consists of three preprocessing strategies that are used to extract the subimages (from the raw input image) that are most relevant for classification. The first strategy extracts the whole face images for training and the second strategy extracts the upper half of the face for training. A third strategy added background to the driver's face by expanding the bounding box of the face in different directions. The images were extracted and sent to the fine tuning unit for training. The fine- tuning unit consists of two pre-trained models: AlexNet and VGG16. Overall, the authors built six different CNN- based gaze classification models based on different combinations of the above pre-processing strategies and pre- trained models. During the training, a Pre-processing methods and a pre- trained model were selected and trained. The experimental results show that the combination of VGG16 images and profile provides the highest classification accuracy with 93.36%. Combining AlexNet with profile images achieved an accuracy of 88.91%. The authors attribute the poor performance of AlexNet to the large kernel size (11 × 11) and step 4 used in the first convolutional layer of the AlexNet model. This large size and pitch of the kernel cannot effectively record the small movements of the eyelid and pupil. This is very important because the field of view changes with the slightest movement of the eyelid or pupil. VGG-16 model consists of convolution layers with kernel size 3x3 and step 1. The small kernel and step sizes allow the network to capture small eye movements for efficient gaze estimation. In another study [39], the same authors introduced a similar gaze classification approach using four pre- trained networks: AlexNet, VGG-16, ResNet50, and SqueezeNet. But instead of using 3 pre-purification units and 2 fine-tuning units, we used 4 pre-purification units and 4 fine-tuning units.

Lupu et al [3] proposed a communication system for people with disabilities based on a specially designed device consisting of a web camera attached to an eyeglass frame for image capture and processing. Eye movements are detected by the device and spontaneous blinks are associated with pictograms or keyword selections that reflect the patient's needs. The drawback of this system is that the image processing algorithms are of low quality, they cannot recognize the obtained images accurately, and they are not resistant to light intensity. Later, we proposed a mouse eye tracking system using video glasses and a new eye tracking algorithm based on the adaptive binary segmentation threshold of the obtained images to improve the reliability of the communication system.

Garbin et al [4] introduced large datasets for training models for VR systems. The dataset consists of eye images captured using a head-mounted VR system with two synchronized eyes facing the camera. The images in the dataset are taken from the eye areas collected from 152 people and are divided into four subsets. The first subset contains 12,759 labeled images and the second subset contains 252,690 unlabeled images. The third subset consists of 91,200 randomly selected frames from the video sequence.

Zhang et al [5] proposed a calibration- free, appearance-based method that allows users to enter input text simply by blinking an on-screen keyboard.

We divided human gaze into 9 directions: top left, top, top right, left, middle, right, bottom left, bottom, and bottom right. We also created a dataset consisting of multiple images of 25 people with different lighting conditions, eye appearance, location and time. The images were taken using mobile phones, webcams and digital cameras. They implemented a number of data augmentation methods to improve the model's robustness to image resolution, lighting, slight head rotation, and skin tone. They divided the eye conditions into 10 groups of 9 directions (mentioned above) and 1 eye-closed condition. Based on the dataset, we built a CNN gaze classification model that can learn 10 eye states from 2 eye images. The model was evaluated and the accuracy of 95.01% was obtained.

Groman et al [6] Several methods have been tested in the past to track eye movements, but three eye tracking methods are the most popular and widely used in several applications. These are video oculography (VOG), electrooculography (EOG) and infrared pupil corneal irradiation (IR-PCR). An eye-computer interface is a system that allows you to perform functional operations according to your eyes. This system does not require muscle movement. The eyelid management system distinguishes between voluntary and involuntary eyelids and describes voluntary eyelids or sequences. Vision-based glare detection methods can be broadly divided into two categories. Vision- based glare detection methods can be broadly divided into two categories. Active glare detection: This type of technology relies on special light and uses the reflective properties of the eye. Light enters the eye from an object and is reflected by the retina. This type of blind spot detection provides more accurate results and is a fast and reliable method. The passiveflash detection method does not use an additional light source and the speed of the flash is determined from a set of images in the visible spectrum.

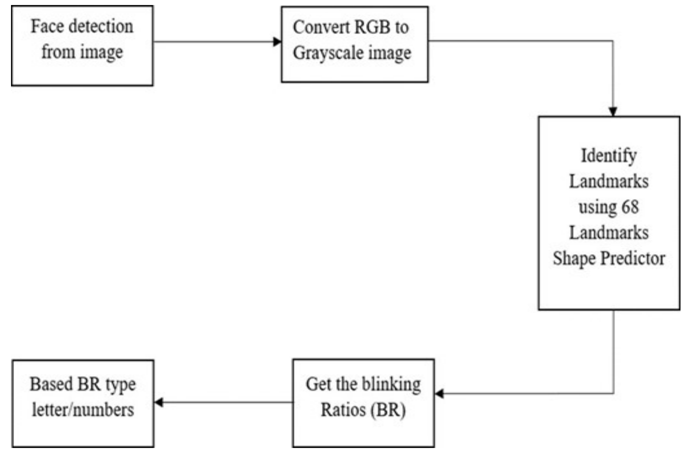

Fig 1: Block Diagram of the proposed method

Face Detection

A computer vision technique called face detection aids in identifying and visualizing human faces in digital pictures. This technique is a particular use of object detection technology for the identification of significant items of a certain class in digital photos and movies. Thanks to technological advances, facial recognition is becoming increasingly important, especially in areas such as photography, security and marketing. OpenCV uses machine learning algorithms to detect faces in images. Because faces are so complex, a simple test cannot tell whether a face has been found. Instead, there are thousands of small patterns and matching features. Algorithms divide the problem of dividing the surface into thousands of small dimensions, each of which is easy to solve. This problem is also called classifiers. For something like a face, there may be 6000 or more classifiers, all of which must match (with errors) for detection. But therein lies the problem. To detect a face, the algorithm starts at the top left of the image and goes through the small data, looking at each point and saying "This is a face? ... This is a face? ... This is a face?" Because there are more than 6,000 trials in each block, each block can have millions of calculations. Algorithms can have 30-50 steps or cascade and only determine the state after all steps have been completed.

Convert RGB to Grayscale Image

After the face detection then the video or image detected is converted from RGB to Grayscale. This conversion process is done by importing the cv2 module (import cv2). Then, we need to create an object of class video capture, which will allow us to get the frames from a webcam. As input of the constructor, we need to pass the identifier of the camera we want to use, as a number. If we only have a webcam attached to the computer, we can pass the value 0. Then, we will need to obtain the frames of our camera one by one and convert them to gray. This means we should write that logic inside an infinite loop, so we keep obtaining frames, converting them to gray and displaying them.

Identify 68 Landmarks using Shape Predictor

The 68 landmarks are predicted using Dlib. There we can see that points from 1 to 68. But sometimes we don't need all 68 feature points, we can customize those points according to our requirements.There are mostly two steps to detect face landmarks in an image which are given below:

Face Detection: Face detection is the first methods which locate a human face and return a value in x,y,w,h which is a rectangle.

Face landmarks: After getting the location of a face in an image, then we have to through points inside of that rectangle.

Now to draw landmarks on the face of the detected rectangle. After loading our image, we resize it, convert it to grayscale, and pass the grayscale image to the Dlib's face detection, rects contains the bounding boxes of the detected faces and the (x,y) coordinates of each bounding box. We are passing the landmarks values and image to the face Points. We are passing landmarks and image as a parameter to a method called draw Points which access the coordinates(x,y) of the ith landmarks points using the part(i).x and part(i).y. All landmarks points are saved in a numpy array and then pass these points to in-built cv2.polylines method to draw the lines on the face using the startpoint and endpoint parameters. Dlib's 68-face landmark model shows how we can access the face features like eyes, eyebrows, nose, etc.

Get the Blinking Ratios

This method is very simple, effective and does not require anything like Image processing. Essentially, this ratio provides a definite relationship between the horizontal and vertical dimensions of the eye. Here we have to find out what happens when we close our eyes. It is easy to point out that an eye is blinking when:

The eyeball cannot be seen

Eyelid is closed

Upper and bottom eyelashes connect together

We must remember that all these actions must take place at a certain time (approximately 0.3 - 0.4 seconds). If the movement takes a long time, we say that the eyes are closed. We make two lines in this system to determine the brightness of the eyes. One line is horizontal and the other is vertical. When there is no vertical line, we can say that the eyes are closed or glazed.

Based BR type letter/number After getting the blinking ratios, if the blinking ratios is approximately 0.3 to 0.4 seconds at this point of point of time, the letter with its changed rectangle color is selected as the letter from the virtual keyboard to be displayed. Another plain/empty window is created using the Numpy library to display all the selected keys from the virtual keyboard. Display the video captured by the web-camera, by displaying all the frames one after the other. Also display the window representing the created-virtual keyboard and the window that displays all the selected keys.

ISSN [Online]: 2583-2654

III RESULTS

After running the program, the computer's webcam opens. The first step of the system is face detection. Therefore, we use real-time face detection and face tagging methods with dlib using Haar-based HoG feature description. The results of face and eye detection are shown in the picture.

Fig 2: Eye detection

White Board

A white board is displayed along with the virtual keyboard. This white board shows the characters or letters typed with the help of eye blinking.

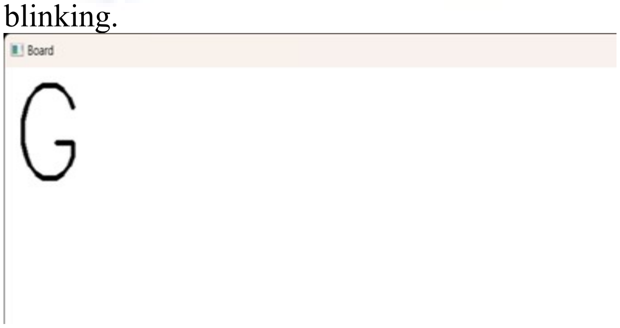

Fig 3 White Space

Keyboard Selection

We calculate the aspect ratio to choose the left or right side of the keyboard. If the aspect ratio is greater than 0.9 (the default value), the left side of the keyboard will be opened, otherwise, it will be opened to the right. In this case, we look to the left or right and close our eyes for a second or more to open thekeyboard. The results of the implementation are shown in figure below.

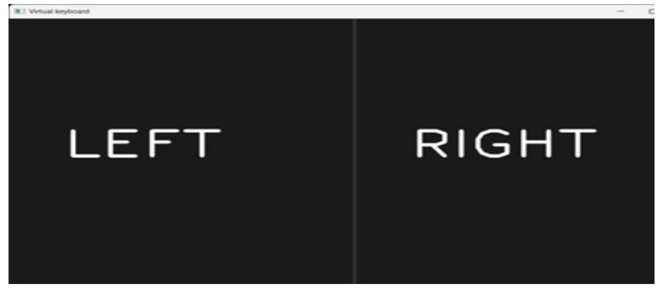

Fig 4: Keyboard Selection

Left Side Keyboard

The gaze ratio is more than 0.9 so the left side of the keyboard is opened and the letter are displayed in the virtual keyboard. The results are shown in below figure.

Fig 5: Left side virtual keyboard

Right Side Keyboard

The aspect ratio is less than 0.9, so the left side of the keyboard is open and the letters are displayed on the virtual keyboard. The results are shown in figure below.

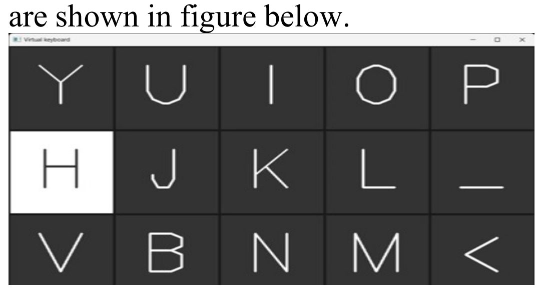

Fig 6: Right side virtual keyboard

Typing using Eye blink

Every key on the virtual keyboard lights up. We fire each key for 10 frames, then the next key. So, when the desired key is lit, we have to close our eyes for about a second and a symbol or letter will appear on the Blackboard.

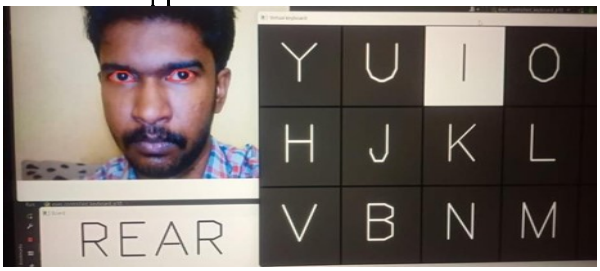

Fig 7: Character typed using Eye blink

IV CONCLUSION

The proposed virtual keyboard has been tested and it shows improved performance in terms of CPM, WPM, and total error rate than the existing models. The virtual keyboard has several prominent features such as play sound when a key is pressed. The proposed virtualkeyboard requires only a webcam for taking commands from the user, which makes it cost-effective. Further research such as testing the keyboard on the increased number of users can be carried out.

REFERENCES

[1] Poole and L. J. Ball, "Eye tracking in human computer interaction and usability research: current status and future prospects," Encyclopedia of human computer interaction, pp. 211–219, 2006.

[2] EyeTrackingUpdate. (2011, Jun.) Eye tracking astigmatism correction. [Online]. Available:http://eyetrackingupdate.com/2011/06/27/eyetracking-astigmatism-correction/.

[3] P. Turaga and R. Chellappa, "Age estimation and face verification across aging using landmarks," IEEE Trans Inf. Forensic Secure, no. 7(6), pp. 1780– 1788, 2012. [Online].Available:https://doi.org/10.1109/TIFS.2012.2213812.CrossRefGoogle Scholar.

[4] T. Devries, K. Biswaranjan, and G. W. Taylor, "Age estimation and face verification across aging using landmarks," Canadian Conference on Computer and Robot Vision, IEEEMontreal, 2014, p. 98–103. [Online]. Available:http://ieeexplore.ieee.org/document/6816830/.

[5] D. Guarin, J. Dusseldorp, and T. A. Hadlock, "A machine learning approach for automated facial measurements in facial palsy," Jama Facial Plast. Surg., no. 20(4), p. 335, 2018. [Online]. Available: https://doi.org/10.1001/jamafacial.2018.0030. CrossRefGoogle Scholar.

[6] S. Anping, X. Guoliang, D. Xueha, S.Jiaxin, X. Gang, and Z. Wu, "Assessment for facial nerve paralysis based on facial asymmetry," Australas Phys. Eng. Sci. Med., no. 40(4), p. 851–860, 2017. [Online].Available: https://doi.org/10.1007/s13246- 017-0597-4.CrossRefGoogle.

[7] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the- wild challenge: The first facial landmark localization challenge." [Online]. Available: http://eprints.nottingham.ac.uk/31434/1/tzimir oICCVW13.

[8] M. L. Mele and S. Federici, "Gaze and eye-tracking solutions for psychological research," Cognitive Processing, 2012.

[9] A. Sharma and P. Abrol, "Eye gaze techniques for human computer interaction: A research survey," International Journal of Computer Application, 2013.

[10] M. Chau and M. Betke, "Real time eye tracking and blink detection with usb cameras," 2005.