Proceedings of the International Conference on Automation, Computing and Renewable Systems (ICACRS 2022)
IEEE Xplore Part Number: CFP22CB5-ART: ISBN: 978-1-6654-6084-2

2022 International Conference on Automation, Computing and Renewable Systems (ICACRS) | 978-1-6654-6084-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICACRS55517.2022.10029276

Eye-Move: An Eye Gaze Typing Application with OpenCV and Dlib Library

Abhaya V
Dept of CSE
BMS College of Engineering
Bangalore, India
abhayavd3@gmail.com

Akshay S Bharadwaj
Dept of CSE
BMS College of Engineering
Bangalore, India
akshaysbharadwaj2k@gmail.com

Chandan C Bagan
Dept of CSE
BMS College of Engineering
Bangalore, India
chandancbagan@gmail.com

Dhanraj K
Dept of CSE
BMS College of Engineering
Bangalore, India
dhanrajkrishnamurthy@gmail.com

Dr. Shyamala G
Assistant Professor
Dept of CSE, BMS College of Engineering,
Bangalore, India
shyamala.cse@bmsce.ac.in

***Abstract*—People who are unable to type on a computer or a mobile phone due to inadequacies produced by their hands, such as osteoarthritis, carpal tunnel syndrome, trigger finger, Ganglion cysts, and other disorders, can benefit from eye vision technology. There are currently a number of commercial and non-commercial eye-tracking solutions available, including model-based and appearance-based methods; however, some of these solutions are expensive or unreliable in real-world situations, and others require explicit user calibration, which can be time-consuming. As a result, research into deep learning-based eye-tracking systems have switched to improving these systems. Recent eye-tracking research has focused on the development of deep learning-based eye-tracking algorithms that don't require explicit user calibration. Because of the recent emergence of deep learning, gaze estimation models based on convolutional neural networks (CNNs) are becoming more significant and common. In our research, the proposed system will provide a new user -friendly keyboard interface for the user. User can see the keyboard layout and can type the text by the movement of his or her eyes. The user can enhance his typing rate by making use of a word prediction engine. Word prediction is an assistive technology tool that suggests words while typing.**

***Keywords*—CNN, dlib, Eye-typing, Image interface, python, facial landmark, keyboard, camera, OpenCV, nltk, GUI, Motor neuron disability, Text-to-speech, gTTS**

I. MOTIVATION

In learning, the computer becomes an extension of the human body, and the most fundamental thing one could do is typing. However, some people with serious motor disabilities find it difficult to communicate by typing with their hands.

Typing with eye movement is being used as an assistive technology for people who are physically impaired. Vision-based text entry systems are intended to assist disabled people communicate with text using only their eyes.

The aim of this project is to give the potential to the people who are unable to type on a computer or a mobile phone due to disabilities induced in their hands by conditions like osteoarthritis, Carpal Tunnel Syndrome, Trigger Finger, Ganglion Cysts etc., through the use of eye gaze technology.

The model will be beneficial to both handicapped people as well as people without any disabilities as mentioned above. Our suggested approach will be designed to have better precision and more flexibility to text entry errors than currently available systems. Eye tracking has been approached in a variety of ways, but some of them are unreliable in real-world situations. Bad image quality and unpredictable lighting conditions plague some of these approaches. Convolution Neural Networks (CNN) based eye-gaze estimation models are becoming increasingly popular.

The system is easily expanded to support diverse features such as word prediction and various language models and could also be integrated with home automation systems.

II. INTRODUCTION

Many persons with disabilities can be found in our immediate surroundings. Some people are capable of being self-sufficient and doing their responsibilities on their own, while others are unable to speak. People with such disorders, also known as motor neuron illnesses, find it difficult to be self-sufficient in their daily life and hence require regular assistance and care.

A system that interacts with them and can cater to their needs can be provided to help these people obtain more independence and undertake a portion of their daily duties on their own. This system is designed for Human Computer Interaction (HCI), which is defined as direct communication between a person and a computer system. In order for this to happen, the computer system must accept human input in any

form. However, in order to accomplish the interaction, we primarily focus on the user's eyesight and eye-gaze monitoring.

This eye-gaze tracking can be accomplished with a variety of tools, which are explained in the following sections. The user's eye input can be taken using a tool that can take in a continuous video stream and a tool that can locate the eye region. This information can be fed into a model that categorizes eye-gaze data according to the direction of the gaze on the monitor screen. This model is used to generate eye coordinates, which can subsequently be used to determine the desired interaction by analyzing the key the user intends to press on the computer screen.

The system will use a video-based eye typing arrangement using a camera to capture the eye movements of the user.

The user has to look at a letter on the screen and the letter gets printed. The system consists of eye typing implementation and the second part is text entry into the system. Users can express their thoughts by writing using eye gaze technology and CNN algorithms.

We also bring in functionalities of text to speech and word prediction by training the system with a varied vocabulary of english words. The word prediction module makes it easier for the user to communicate and the text to speech module is mainly for the person interacting with the user to better understand and more efficiently communicate with the user.

III. LITERATURE SURVEY

The literature survey was conducted over the course of November and half of December 2021. This phase included the survey of over 15 papers published over various journals which consisted of work done on the eye-tracking technologies and papers which could help in the project.

a. The method uses a robust classifier that is classified using data sets to estimate gaze direction. Neural networks are used in the framework to give a low-cost way of interacting with machines (CNN). The use of a CNN model increased the system's durability and accuracy while also strengthening its current state. One idea for improvement is to reduce the ratio of writing to typing to a more acceptable level. Word prediction and numerous language models, as well as integration with home automation systems and text communication, may all be added to the app. [1]

b. For 9-direction gaze estimate, a Convolution Neural Network (CNN) model was developed. The input system was a nine-key T9 system based on the 9-direction gaze, which is commonly seen in candy bar phones. To evaluate the CNN model's output and compare it to that of other models, two test sets were generated, one with known users and the other with unknown users. The option of picking letters on screen was preferred by the testers since it allowed them to rapidly identify their favourite letters on the screen. [9] They had to spend more time remembering the letters' positions when they used the off-screen option, on the other hand. Because they couldn't remember where the letters were, they had to test each direction one at a time. The pace of input rose dramatically when they memorized the letter locations. Users got the erroneous letter when they switched their eyes before blinking to choose the proper letter. [2]

c. The two basic techniques are appearance-based and model-based tactics. The gaze is estimated using eye photographs in the appearance-based technique, whereas the gaze is calculated using 2D/3D models that employ near-infrared devices to recreate cornea reflection in the model-based method. A 15.3 percent test error rate arose as a result of this. CNN-based techniques may become fouled up when a lot of head and eye rotation happens during picture processing. One of the two eyes may disappear from the obtained picture, resulting in a greater error rate in gaze estimation. [3]

d. Eye gaze tracking and gaze-based human-computer interactions are actively being investigated in modern consumer devices. [15] Eye gazing has been utilized in virtual and augmented reality systems to extract human behavioural cues as an input modality and to create immersive user experiences. For estimating the gaze, this system proposed a dual eye channel Convolutional neural network, in which both eyes' images were utilized to estimate the gaze. This technology substitutes a calibration-free approach for the appearance-based method, making the system more user-friendly and cost-effective. [4]

e. This research proposes a unique method for analyzing webcam eye movement. Instead of estimating the mapping role as in the feature-based gaze tracking process, feature points are used to acquire eye movement signals. This study suggests a new method for analyzing webcam eye movement. The feature points were used to gather eye movement signals rather than estimating the mapping position as in the feature-based gaze tracking technique. This study looked at additional signals including relative iris centre displacement and open width variance to evaluate eye movement patterns. The behavior-CNN was trained to extract more expressive eye movement characteristics from eye movement data in order to identify activities. [5]

IV. EXISTING SYSTEM AND DISADVANTAGES

Prior to the emergence of eye-gaze estimation and human-computer interface [14] through the medium of vision, humans and computers interacted directly through the capture of brain signals. This technology is still in use and is noted for its accuracy since the human communicates directly with the computer system using his brain, which is the most efficient method available.

However, a significant amount of effort and cash is required to make this a reality. This is owing to the fact that the human brain must be physically attached to the computer system via wires and connections, and special input ports must be built in order to read brain signals of the human as the input for the computer.

As a result, a considerably more cost-effective and straightforward approach was required to enable human-computer interactions that were also more accessible to

everybody. As a result, eyesight was developed as an input for computer systems to communicate with humans. Any individual with complete motor difficulties and the inability to speak or communicate with their surroundings can benefit from using their eyesight. The eyesight approach is also cost effective because it does not require a lot of gear to work well and only requires a monitor screen, making it considerably more economical.

V. TOOLS USED FOR DEVELOPING THE SYSTEM

We need to have a basic understanding of convolutional neural networks, OpenCV, Python and its libraries in order to understand how they can be integrated into an eye gaze typing application.

A. Convolutional Neural Network (ConvNet/CNN) [11] is a Deep Learning approach for assigning importance (learnable weights and biases) to various aspects/objects in an image and separating them. A ConvNet requires significantly less pre-processing than other classification algorithms. With enough training, ConvNets can learn these filters/characteristics, whereas simple techniques require hand-engineering of filters.

B. OpenCV (Open Source Computer Vision Library) is a free software library for computer vision and machine learning. OpenCV was created to provide a standard infrastructure for computer vision applications and to speed up the integration of machine perception into commercial products. Businesses can readily use and change OpenCV because it is BSD-licensed software. Over 2500 optimized algorithms are included in the collection, which comprise a combination of classic and cutting-edge computer vision and machine learning techniques. These algorithms can be used to detect and recognize faces, identify objects, extract 3D models of things from stereo cameras, stitch images together to provide a high-resolution representation of an entire scene, track moving objects, classify human activities in films, and track camera movements, follow eye movements, remove red eyes from flash images, recognize scenery, and create overlay markers.

C. Python is a dynamically semantic object-oriented high-level programming language that is interpreted. Because it features built-in high-level data structures, as well as dynamic type and dynamic binding, it's ideal for Rapid Application Development and as a scripting or glue language for integrating existing components. The readability of Python's succinct, easy-to-learn syntax is prioritized, lowering programme maintenance costs. Python supports modules and packages, allowing for programme modularity and reusability. For all major operating systems, the Python interpreter and its enormous standard library are available for free download and distribution in source or binary form. TensorFlow and Keras are two libraries that aid in the system's development.

D. TensorFlow is a machine learning platform that is totally free and open source. It has a wide, extensible ecosystem of tools, libraries, and community resources that allow academics to improve the state-of-the-art in machine learning while also allowing developers to easily build and deploy machine learning applications.

E. Keras is a human-centric API rather than a machine-centric one. It adheres to best practices for lowering cognitive load, such as providing consistent and simple APIs, limiting the amount of user activities required for common use cases, and showing clear and actionable error messages. It comes with a lot of documentation and development instructions. [10]

VI. METHODOLOGY

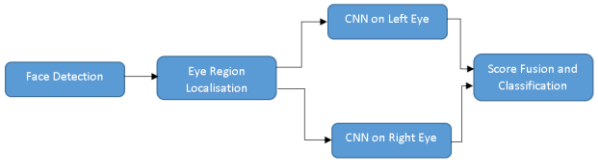

Fig. 1. A brief flow diagram showing the system methodology

The system as is seen in the image can be able to function in the flow as shown in Fig. 1.

- We detect the face of the user using the OpenCV library and the python code. The OpenCV library helps us to take in a constant video input and using the OpenCV library we can detect the face of the user on the screen.

- Using a facial landmark detector (like the dlib library) we can generate a facial landmark. A few example facial landmarks consist of 68 such landmark points which clearly depict the face region of the user. The points are present at the jaw, nose, eyebrows, mouth and eyes of the user. Since we need only the eye input we only choose the points which form the eye region landmarks. There are 6 landmark points around each eye in the facial landmarks generated and hence twelve points in total are needed. These twelve points are therefore used in the eye region localization.

- We use a CNN model for each eye. This is done so as to improve the efficiency of the system. CNN is used to check the eye-gaze direction using a model. This CNN model takes in the localized eye-region points as the input and classifies each eye's direction as either up, down, left or right. These four directions can be easily checked by using coordinates on a two-dimensional plane which can provide us with the required output. [8]Therefore, the eye-gaze direction is given as the output in the form of coordinates. For each eye the output direction is given as output by the CNN classifier as coordinates.

- The coordinates given as output from the CNN model for both eyes are combined together and are used to calculate a final score using python. This final score determines the actual eye-gaze direction of the user

which is hence used to calculate the final direction and hence select the key the user intends to select to interact with the system.

VII. DESIGN

A. SYSTEM ARCHITECTURE

A formal description and representation of a systematic definition and representation of a system structured in a way that promotes understanding about the system's mechanisms and behaviours is known as an architecture diagram. Figure 2 depicts the overall system, highlighting the primary components that will be constructed as well as their interfaces as part of the system proposed. The system will follow the methodology described in the sections above.

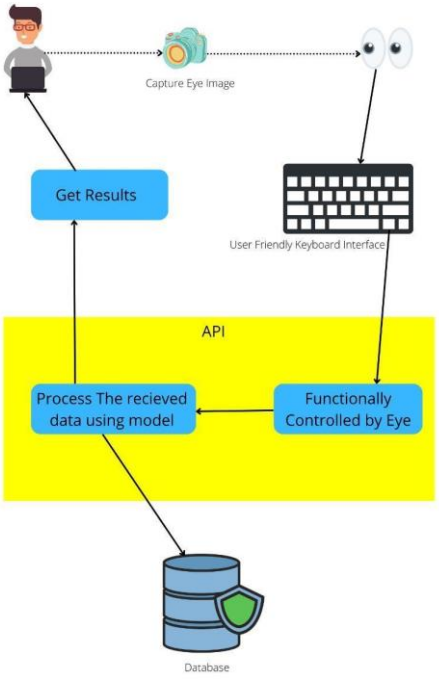

Fig. 2. System Architecture diagram

The steps are as follows:

1) A keyboard interface is supplied to the user.
2) The keyboard layout includes extra features such as home automation that can be managed with the eyes.
3) A camera is used to collect the user's eye image.
4) The image is recorded and submitted to the API, which then sends it to the model.
5) The model processes the received data using the ML library and generates the results.
6) Show the results to the user.

B. ACTIVITY DIAGRAM

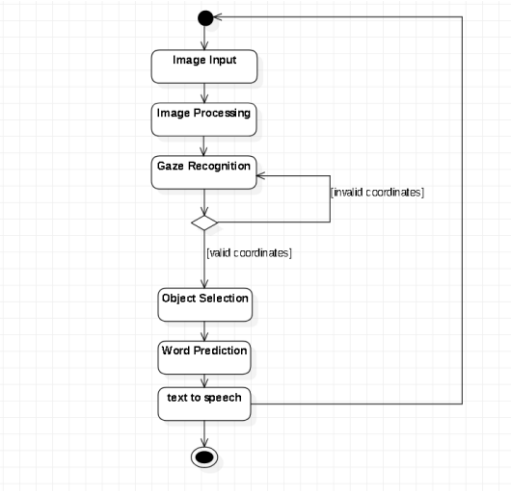

Fig. 3. Activity diagram

The activity diagram depicts the system's numerous actions:

1. Start the application, and the system will take a screenshot of the input image.
2. The image is processed and transmitted to CNN for gaze coordinate estimation.
3. Users can use the eye typing module once the eye gaze is calibrated.
4. Object selection is carried out using gaze coordinates.
5. In eye typing, the user types the text using a word prediction module, and the text is then transformed to speech.

VIII. MODULES IMPLEMENTED

The following modules have been implemented in the proposed system which will help the user to interact with the computer system efficiently.

1. Eye Gaze Recognition module:

   This module is used to establish the precise location of the eye's gaze. This can be accomplished in a variety of ways. The image of the user's eyes is captured using a camera. [7] If the surrounding conditions are poor, the image cannot be accurately captured to detect the gaze. As a result, the lighting conditions must be suitable to capture the image correctly. The image is transferred to the Eye Gaze Recognition Module for processing after it has been recorded in correct lighting conditions. In this module, the collected image is analyzed, and the exact gaze coordinates are calculated. The module then returns coordinates as a consequence.

2. Eye Typing Module:

   This module employs the gaze established in the previous module. The gaze's coordinates are received. The above-mentioned libraries are utilized to aid in the typing of text based on gaze coordinates. The user should be able to see what he is typing, hence the module includes a display interface for the written text. [6]

3. Word Prediction Module:

   The eye typing module makes use of this module to improve its performance. Word prediction is an intelligent word processing technology that can help a variety of people avoid writing breakdowns by minimizing the number of keystrokes required to type words. In this example, a machine learning method is utilized to predict the missing word in order to improve the speed and efficiency of eye typing, and it is trained using various words of the English language vocabulary. nltk library is used to achieve this functionality.

4. Text to Speech Module:

   TTS (text-to-speech) is an assistive technology that reads digital text out loud. It's also known as "read aloud" technology. It can transform words from a computer or other digital device into audio. The user's chosen letter will be converted to speech using the text to speech module in this situation. gTTS and pyttsx3 libraries were used to implement this functionality.

IX. RESULTS

The below images show the eye-typing module including the virtual keyboard, the eye-gaze recognition module with the pointer, the word prediction module and the text to speech module all being integrated and forming a cohesive communication system for the user.

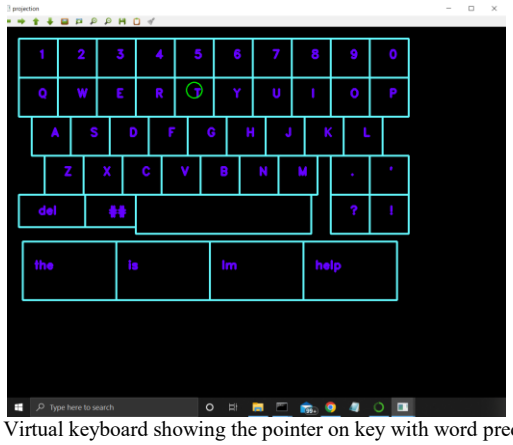

Fig. 4. Virtual keyboard showing the pointer on key with word prediction

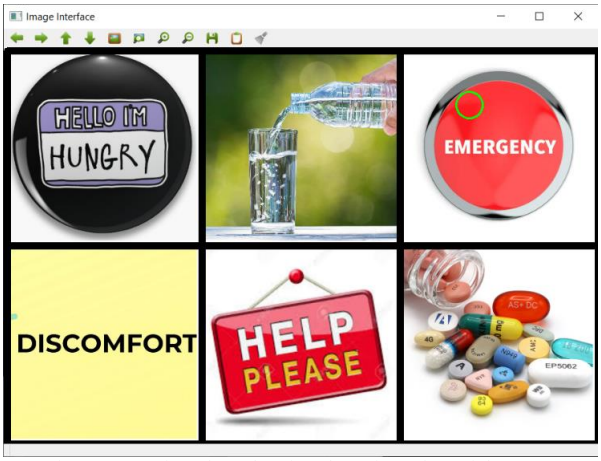

Fig. 5. Image interface showing the pointer for quick access

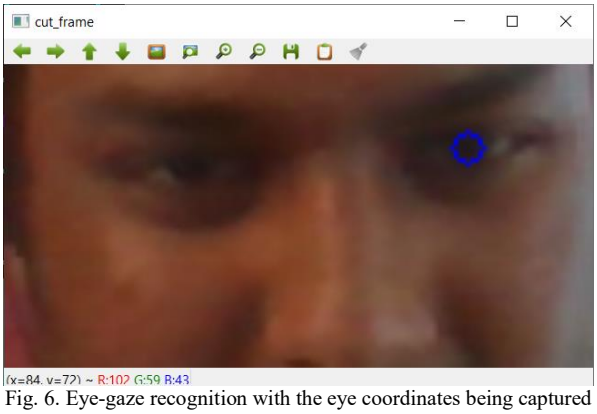

Fig. 6. Eye-gaze recognition with the eye coordinates being captured

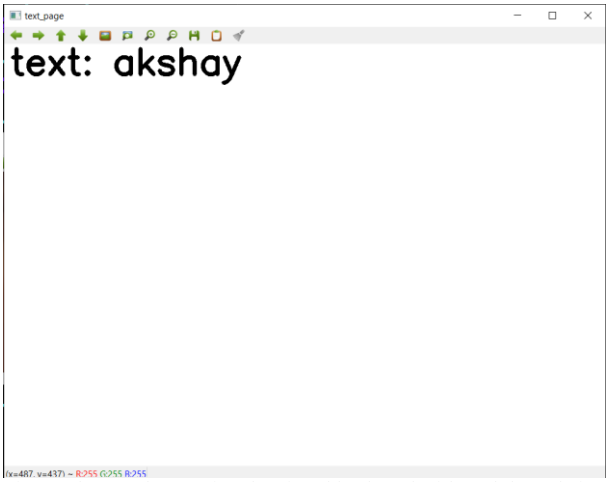

Fig. 7. The typed text using the virtual keyboard with each letter being pronounced using text to speech

The results indicate the successful working of the eye-typing project as the user is able to type the text and the system is able to recognize and communicate the text.

X. CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, our method of eye-gaze typing intends to help out people with motor neuron diseases and other such severe

forms of disability to communicate with people around them and to help them be more independent. We do this using tool such as OpenCV, Python and also using convolution neural networks which can efficiently take in eye input. Using these tools we classify the eye-gaze direction of the user and can select the appropriate key. This approach has been developed in the form of modules like eye-typing, word prediction and also text to speech so as to improve the communication abilities of the user with people around him.

The high-level design incorporating all the sub-systems as mentioned will be incorporated to make our project reliable and efficiently able to serve its purpose by helping disabled people communicate. The eye-typing module has been implemented and the user can now type with eye-gaze. We can hence establish a communication between user and the computer using eye-tracking.

The word prediction module has also been integrated along with the eye-typing module which will give the user a more efficient and easier way of communicating by just clicking on a word trying to be communicated rather than typing it whole.

The Text to speech module will also be incorporated so as to audibly convey the user's message to be communicated by sounding each alphabet once it has been typed by the user using the virtual keyboard.

Further, out of the scope of this project we also intend to include a home automation module which can also help the user to be more in control of his house and therefore, be more independent. We will take this up as a future enhancement that can be made to this project as currently it is out of scope for what we wish to achieve now with this project.

REFERENCES

[1] A. Akinyelu and P. Blignaut, "Convolutional Neural Network-Based Methods for Eye Gaze Estimation: A Survey," in IEEE Access, vol. 8, pp. 142581-142605, 2020, doi: 10.1109/ACCESS.2020.3013540
[2] Zhang, C., Yao, R. & Cai, J. Efficient eye typing with 9-direction gaze estimation. Multimed Tools Appl 77, 19679–19696 (2018)
[3] Ms. Saily Bhagat, Ms. Tanvi Patil, Ms. Varsha Singh, Mr. Abhishek Bhatt, Ms. Snehal Mumbaikar iGaze-Eye Gaze Direction Evaluation to Operate a Virtual Keyboard for Paralyzed People
[4] Z. Li, M. Li, P. Mohapatra, J. Han and S. Chen, "iType: Using eye gaze to enhance typing privacy," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, 2017, pp. 1-9, doi: 10.1109/INFOCOM.2017.8057233
[5] W. Hansen and J. P. Hansen, "Robustifying Eye Interaction," 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), 2006, pp. 152-152, doi: 10.1109/CVPRW.2006.181.
[6] W. Hansen, J. P. Hansen, M. Nielsen, A. S. Johansen and M. B. Stegmann, "Eye typing using Markov and active appearance models," Sixth IEEE Workshop on Applications of Computer Vision, 2002. (WACV 2002). Proceedings., 2002, pp. 132-136, doi: 10.1109/ACV.2002.1182170.
[7] Pholder: An Eye-Gaze assisted reading application on Android
[8] Deep Learning based Eye Gaze Tracking for Automotive Applications: An Auto-Keras Approach
[9] Convolutional Neural Network-Based Methods for Eye Gaze Estimation: A Survey. ANDRONICUS A. AKINYELU AND PIETER BLIGNAUT
[10] Convolutional Neural Network Implementation for Eye-Gaze Estimation on Low-Quality Consumer Imaging Systems Joseph Lemley, Anuradha Kar, Alexandru Drimbarean, Peter Corcoran
[11] Sangeetha, S. K. B. "A survey on Deep Learning Based Eye Gaze Estimation Methods." Journal of Innovative Image Processing (JIIP) 3, no. 03 (2021): 190-207
[12] Sathesh, A. "TYPING EYES: A HUMAN COMPUTER INTERFACE TECHNOLOGY." Journal of Electronics and Informatics 1, no. 2 (2019): 80-88.
[13] Y. Li, X. Xu, N. Mu and L. Chen, "Eye-gaze tracking system by haar cascade classifier," 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), 2016, pp. 564-567, doi: 10.1109/ICIEA.2016.7603648.