

# Wheelchair Control System based Eye Gaze

Samar Gamal Amer<sup>1</sup>, Sanaa A. kamh<sup>3</sup>  
Marwa A. Elshahed<sup>4</sup>  
Department of Physics  
Faculty of Women  
Ain Shames University, Cairo, Egypt

Rabie A. Ramadan<sup>2</sup>  
Computer Engineering Department  
Faculty of Engineering  
Cairo University  
Cairo, Egypt

**Abstract**—The inability to control the limbs is the main reason that affects the daily activities of the disabled which causes social restrictions and isolation. More studies were performed to help disabilities for easy communication with the outside world and others. Various techniques are designed to help the disabled in carrying out daily activities easily. Among these technologies is the Smart Wheelchair. This research aims to develop a smart eye-controlled wheelchair whose movement depends on eye movement tracking. The proposed Wheelchair is simple in design and easy to use with low cost compared with previous Wheelchairs. The eye movement was detected through a camera fixed on the chair. The user's gaze direction is obtained from the captured image after some processing and analysis. The order is sent to the Arduino Uno board which controls the wheelchair movement. The Wheelchair performance was checked using different volunteers and its accuracy reached 94.4% with a very short response time compared with the other existing chairs.

**Keywords**—Dilip; numpy; gaze ratio; facial landmarks points; deep learning

## I. INTRODUCTION

Wheelchairs have become essential for the elderly and the disabled nowadays. Electric wheelchairs are available on the market and are typically controlled using a joystick [1]. In some cases, many people can be injured unexpectedly during various events, including falls, accidents, violence, or even injuries while playing sports. These events may result in devastating neuromuscular disorders, resulting in significant disabilities as a result of injuries to the spinal cord, which may be caused by a lack of nerve supply, paralyzing every affected part of the human body. As a result, there is an emerging need to assist these people in improving their abilities to carry out their daily routine without the assistance of others. [2]. This was a strong motivation for issuing this research paper, different systems were used to control wheelchairs, such as the voice control system [3], brain control system [4], and eye control system [5]. In the voice control system, the Wheelchair is controlled by the sound so that the user gives voice commands such as forward, right, left, and stop, and the system will recognize the word then send the command to move the Wheelchair accordingly. The Wheelchair controlled by the brain signals depends on capturing the patient's so-called EEG signal to move the chair. The downsides of these systems include a reduced immunity to noise, which might cause the system to become distracted and respond incorrectly.

This research focuses on designing a wheelchair that doesn't depend on any physical movement of the user but tracks the motion of the eye with achieving better accuracy than others also, minimizing the delay response time of the detection. The proposed eye-tracking electric wheelchair differs from others with good validation parameters of the controller which uses simple equations depend on counting the white pixels in both eyes to identify the eye pupil position and direction in the user's face. The main idea of this work depends on capturing the user's image using a webcam. some image processing techniques were performed to determine the location of the face, the eye, and its direction, and then control the Wheelchair via the Arduino board [6]. The face detection technique is a computer technology used to detect the human face from a digital image based on facial landmark points. It is considered as a set of key points on human face images [7]. These points are described in the image by their actual coordinates (x, y). A pictorial image has been used to model the facial landmarks relations, as shown in Fig. 1.

This figure is an example of a facial landmark 68 points model (from 1 to 68). After face detection, the system can detect the eye gaze. Then, the process of detecting the position of the eye sclera is performed by calculating the value of the gaze ratio. The gaze ratio is the ratio of the white pixels between each side in both eyes. This ratio allows determining the area at which the person is looking [8,9,10].

The proposed smart wheelchair contributes to facilitating the lives of the disabled and making their lives easier and reducing their financial burden. Also, improving the accuracy and reducing the time between the user's eye and chair movements than other wheelchairs.

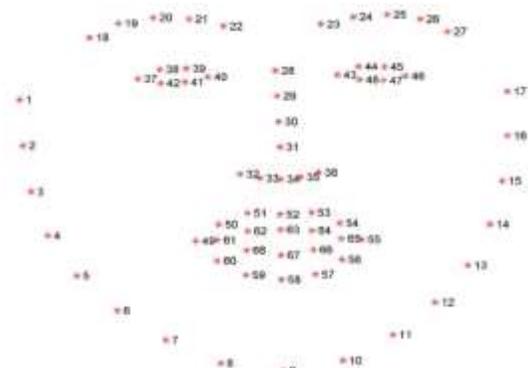


Fig. 1. Facial Landmark Points.

## II. RELATED WORK

The technology that allows extensive use of eye-tracking principles includes sectors in the automotive industry, medical science, exhaustion simulation, automobile simulators, cognitive tests, computer vision, behavior recognition, etc. Over a while, the importance of eye recognition and monitoring in industrial applications grew. This importance of eye-tracking applications leads to more effective and durable designs required in many modern appliances. An extensive review of the literature relating to the eye-tracking system has been carried out in healthcare applications.

The authors in [11] uses machine learning to extract the iris by segmentation algorithm. All experiments were implemented using the MATLAB software, the algorithm accuracy reached to 86% which is not satisfactory, also the authors mentioned that the speed of detection increased but did not mention its value.

A wheelchair is designed in [12], which is based on eye movement. This chair consists of three main parts: imaging processing module, wheelchair-controlled module, SMS manager module, and appliance-controlled module. The imaging processing module consists of a camera installed with glasses. The captured image is transferred to the Raspberry Pi microcontroller will be processed using OpenCV to derive the direction of the two-dimensional eyeball. The eyeball movement will be transmitted wirelessly to a unit controlled by a wheelchair to control movement. But the Calibration time is not calculated. Their proposed chair is relatively high cost and lack of components. A microcontroller called Raspberry Pi were used which may cost up to 70\$, But our proposed system used the Arduino Uno, which doesn't exceed 10 \$ and it reaches the same result.

In [13], the researcher designed a wheelchair that moves by eye-tracking, first the camera capture the user image and detect the eye position by edge detection technique (Laplacian), Then detect the pupil direction by segmentation algorithm (Haar cascade algorithm) then, controlling the wheelchair DC motor with the help of a PID controller. The accuracy of detection was 90%. But Calibration time is not calculated, also, the Laplacian technique is unacceptably sensitive to noise. Haar cascade is an algorithm that depends on edge or line detection so it's suitable for face and eye detection but not suitable to detect eye movement.

The main idea in [14] is that the Wheelchair is controlled by the eyeball movement with the aid of a webcam, which further undergoes multiple image processing techniques. Continuous image is captured to detect the eye pupil's location; then, the Haar cascade algorithm is applied with the wheelchair image processing techniques. The DC motor is mounted on the wheels for quick wheelchair travel. The ultrasonic sensor is mounted on the Wheelchair to locate any obstacles in the direction of its movements and prevents the movement of the Wheelchair as per sensor order. The chair accuracy was 91% which needs improvement. Also, Haar cascade is multiple weak classifiers as mentioned before in this application and the wheelchair move in 4 direction, doesn't move in 360° as our proposed wheelchair move in 360° in the direction of the eyes.

In [15], a wheelchair-based eye-tracking system is presented, which is based on record a video to user's face using an infrared camera, the video is then fed into the proposed algorithm that of six stages: (1) eye area extraction, (2) iris boundary detection, (3) keyframe detection, (4) pupil localization, (5) deviation estimation, and (6) evaluation of strabismus. A database was created that included cover test data from both strabismus and normal subjects. The results of the experiments show that our proposed method can accurately evaluate strabismus deviation. The accuracy was over 91%, in the horizontal direction, and it was over 86% in the vertical direction. The only drawback is that measuring accuracy can improve.

The authors in [16] propose a system that enables a person to have a control wheelchair across the eye gaze. The system consists of a wheelchair, eye-tracking glasses, and a depth camera, Ambient Space Engineering, Portable Computer Sit Flexible Stand for Maximum comfort, and a safety off switch to turn off the system when needed. The author uses the CNN algorithm to capture the eye and calculate its direction. The accuracy of their study reached 92% and the calibration time reached 36 s. CNN algorithm is not preferred in face and eye detections as it is complex. Also, from the results that the calibration time is very long 36 s.

The proposed system is based on the deep Learning base method which has better accuracy in all face parts detection and an easy method to extract eyes Using the dlib and his Face Landmark points.

## III. PROPOSED SYSTEM ARCHITECTURE

The proposed system is simple and has low-cost components, as shown in Fig. 2. It is based on taking the user's face image. Getting the eye's location and then determining the pupil's direction which has a different value. The obtained values are transmitted to the Arduino board connected to a wheelchair to control its moving directions. The designed Wheelchair consists of an integrated circuit called L293d and a motor of the type of DC that works from 3 to 6 V.

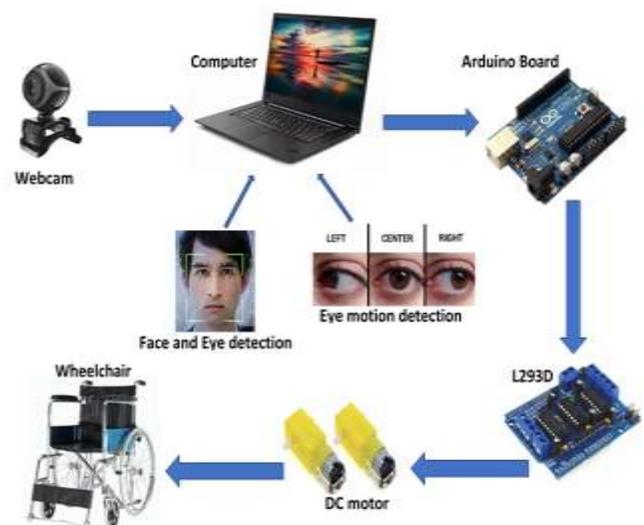


Fig. 2. Proposed System Architecture.

Face and eye detection are determined; then calculate the gaze ratio, which determines the eye direction is (left or right or center). The Arduino Uno is used with an L293D integrated circuit, which is a typical Motor driver or Motor Driver IC. L293D is a 16-pin IC that can control two DC motors simultaneously in any direction and then control in Wheelchair. 5 adult volunteers aged 15 to 50 years old are examined in this study.

This study presents a Technique depend on an algorithm is called "One Millisecond Face Alignment with an Ensemble of Regression Trees" developed by two Swedish Computer Vision researchers Kazemi and Sullivan in 2014[17,18], This detector is built in the dlib library and it detects facial landmarks very quickly and accurately and this algorithm is based on dividing the face into 68 points, Beside the pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face, This occurs by calling a frontal face detector from dlib library. This is a pre-trained detector based on Histogram of Oriented Gradients (HOG) features and a Linear SVM object detector [19]. And the dataset on which the dlib facial landmark was trained in the shape predictor 68 face landmarks" dataset. The facial landmarks with dlib technique don't need any preprocessing or filters but need a high-efficiency camera with strong lenses and high focus, The more efficient the camera, the higher accuracy of eye-tracking.

#### IV. METHOD

This work used a computational efficient technique to precisely predict the position of face landmarks. A cascade of regressors is used in our proposed strategy.

##### A. The Cascade of Regressors

To begin we introduce some notation. Let  $x_i \in \mathbb{R}^2$  be the x, y-coordinates of the  $i^{\text{th}}$  facial landmark in an image I. Then the vector  $S = (x_1^T, x_2^T, \dots, x_p^T)^T \in \mathbb{R}^{2p}$  denotes the coordinates of all the p facial landmarks in I. In this paper, the vector S is referred to as the shape. And  $S^{\wedge}(t)$  is used to denote our current estimate of S. Each regressor,  $r_t(\cdot, \cdot)$ , in the cascade predicts an update vector from the image and  $S^{\wedge}(t)$  that is added to the current shape estimate  $S^{\wedge}(t)$  to improve the estimate:

$$S^{\wedge}(t+1) = S^{\wedge}(t) + r_t(I, S^{\wedge}(t)) \quad (1)$$

The regressor  $r_t$  makes its predictions based on features, such as pixel intensity values, computed from I and indexed relative to the current shape estimate  $S^{\wedge}(t)$ [16]. We will explain how this indexing is performed in detail.

1) *Learning each regressor in the cascade:* Assume we have training data  $(I_1, S_1), \dots, (I_n, S_n)$ . To learn the first regression function  $r_0$  in the cascade we create from our training data triplets of a face image, an initial shape estimate is obtained by,

$$(I_{\pi_i}, s_i^{\wedge 0}, \Delta s_i^{\wedge 0}) \text{ where } \pi_i \in \{1, \dots, n\} \quad (2)$$

$$s_i^{\wedge 0} \in \{S_1, \dots, S_n\} \setminus S_{\pi_i} \quad (3)$$

$$\text{And } \Delta s_i^{\wedge 0} = S_{\pi_i} - s_i^{\wedge 0} \quad (4)$$

For  $i = 1, \dots, N$ ,  $N = nR$  where R is the number of initializations used per image I, And by algorithm 1. The next regressor  $r_i$  in the cascade by setting (with  $t = 0$ ) is obtained by:-

$$s_i^{\wedge(t+1)} = s_i^{\wedge(t)} + r_i(I_{\pi_i}, s_i^{\wedge(t)}) \quad (5)$$

$$\Delta s_i^{\wedge(t+1)} = S_{\pi_i} - s_i^{\wedge(t+1)} \quad (6)$$

This process is iterated until a cascade of T regressors  $r_0, r_1, \dots, r_{T-1}$  are learned which when combined give a sufficient level of accuracy.

##### 2) Learning $r_i$ in the cascade:

Have training data  $\{(I_{\pi_i}, s_i^{\wedge(t)}, \Delta s_i^{\wedge(t)})\}_{i=1, N}$  and the learning rate

(shrinkage factor)  $0 < v < 1$

Initialize

$$f_0(I, S^{\wedge(t)}) = \arg \min \sum_{i=1}^N \|\Delta s_i^{\wedge(t)} - \gamma\|^2$$

$\gamma \in \mathbb{R}^{2p}$

2. for  $k = 1, \dots, K$ :

(a) Set for  $i = 1, \dots, N$

$$r_{ik} = \Delta s_i^{\wedge(t)} - f_{k-1}(I_{\pi_i}, s_i^{\wedge(t)})$$

(b) Fit a regression tree to the targets  $r_{ik}$  giving a weak regression function  $g_k(I, S^{\wedge(t)})$ .

(c) Update

$$f_k(I, S^{\wedge(t)}) = f_{k-1}(I, S^{\wedge(t)}) + v g_k(I, S^{\wedge(t)})$$

3. Output  $r_i(I, S^{\wedge(t)}) = f_k(I, S^{\wedge(t)})$

##### B. Tree based Regressor

Now we'll go through the most important implementation details for training each regression tree.

Shape invariant split tests

We make a decision based on thresholding the difference between the intensities of two pixels at each split node in the regression tree. When defined in the coordinate system of the mean shape, the pixels used in the test are at positions u and v. Let  $k_u$  be the index of the facial landmark in the mean shape that is closest to u and define its offset from u as:

$$\delta x_u = u - x^{-k_u} \quad (7)$$

Then for a shape,  $S_i$  defined in image  $I_i$ , the mean shape image is given by

$$u^- = x_{i, k_u} + \frac{1}{s_i} R_i^t \delta x_u \quad (8)$$

Where  $s_i$  and  $R_i$  are the scale and rotation matrix of the similarity transform which transforms  $S_i$  to  $S^-$ , the mean shape. The scale and rotation are found to minimize.

$$\sum_{j=1}^p \|x'_j - (s_i R_i x_{i,j} + t_i)\|^2 \quad (9)$$

The sum of squares between the mean shape's facial landmark points,  $x_j^-$ 's, and those of the warped shape.  $V'$  is similarly defined. Each split is formally a decision involving three parameters  $\theta = (\tau, u, v)$  and is applied to each training and test example as

$$h(I_{\pi i}, s_i^{\hat{t}}, \theta) = \begin{cases} 1 & I_{\pi i}(u') - I_{\pi i}(v') > \tau \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Where  $u'$  and  $v'$  are defined using the scale and rotation matrix which best warp  $s_i^{\hat{t}}$  to  $S^-$  according to equation (7).

### 1) Choosing the node splits

The training of the regression tree

To train the regression tree we randomly generate a set of candidate splits, that is  $\theta$ 's, at each node. We then greedily choose the  $\theta^*$ , from these candidates, which minimizes the sum of square error.

This corresponds to minimizing  $Q$ , which is the set of indices of the training examples at a node.

$$E(Q, \theta) = \sum_{s \in \{l, r\}} \sum_{i \in Q_{\theta, s}} \|r_i - \mu_{\theta, s}\|^2 \quad (11)$$

where  $Q_{\theta, l}$  is the indices of the examples that are sent to the left node due to the decision induced by  $\theta$ ,  $r_i$  is the vector of all the residuals in the gradient boosting algorithm, computed for an image  $I$  and

$$\mu_{\theta, s} = \frac{1}{\|Q_{\theta, s}\|} \sum_{i \in Q_{\theta, s}} r_i \text{ for } s \in \{l, r\} \quad (12)$$

The optimal split can be found very efficiently because if one rearranges equation (10) and omits the factors not dependent on  $\theta$  then one can see that

$$\arg \min_{\theta} E(Q, \theta) = \arg \max_{\theta} \sum_{s \in \{l, r\}} \|Q_{\theta, s}\| \mu_{\theta, s}^T \mu_{\theta, s} \quad (13)$$

Here we only need to compute  $\mu_{\theta, l}$  when evaluating different  $\theta$ 's, as  $\mu_{\theta, r}$  can be calculated from the average of the targets at the parent node  $\mu$  and  $\mu_{\theta, l}$  as follows:

$$\mu_{\theta, r} = \frac{\|Q\|\mu - \|Q_{\theta, l}\|\mu_{\theta, l}}{Q_{\theta, r}} \quad (14)$$

2) *Feature selection*: The thresholding of the difference in intensity values at a pair of pixels is used to make the decision at each node. Unfortunately, the drawback of using pixel differences is the number of potential splits (feature) candidates is quadratic in the number of pixels in the mean image. This makes it difficult to find good  $\theta$ 's without searching over a very large number of them. However, this limiting factor can be eased, to some extent, by taking the structure of image data into account by the equation:

$$P(u, v) = \alpha e^{-\lambda \|u - v\|} \quad (15)$$

Over the distance between the pixels used in a split to encourage the selection of closer pixel pairs. We found using this simple prior reduces the prediction error on several face datasets.

## V. EXPERIMENT

### A. The Proposed System Flowchart

The proposed system is built based on Python programming language in Pycharm cross-platform. Building the system depends on OpenCV and Dlib libraries, it has been used dlib get frontal face detector in a frame or image. Dlib predictor is a tool that takes in an image containing some objects and outputs a set of point locations that define the pose of the objects. Here the shape predictor 68 face landmarks. The data model is used to create the predictor object.

The implementation of the system can be summarized in the following four points as in Fig. 3:

- The face and the eyes detection
- Calculating the gaze ratio to detect the eye direction
- The wheelchair design.
- The Connection between OpenCV and Arduino.

In this study, we can configure facial landmark points according to our specifications (detect the face and the eyes) as in Fig. 1. For instance, face landmarks that are enough can be from point 1 to point 27. Besides, left eye landmarks can be defined based on points 37-42, and then the right eye landmarks could be defined based on points 43-48.

### B. Calculating the Gaze Ratio to Detect the Eye Direction

It is considered that the eye is divided into two parts white part (sclera) and the color part (pupil). The idea is to find out which of the two parts is more sclera visible as the sclera expresses the direction the eye is looking, as shown in Fig. 4.

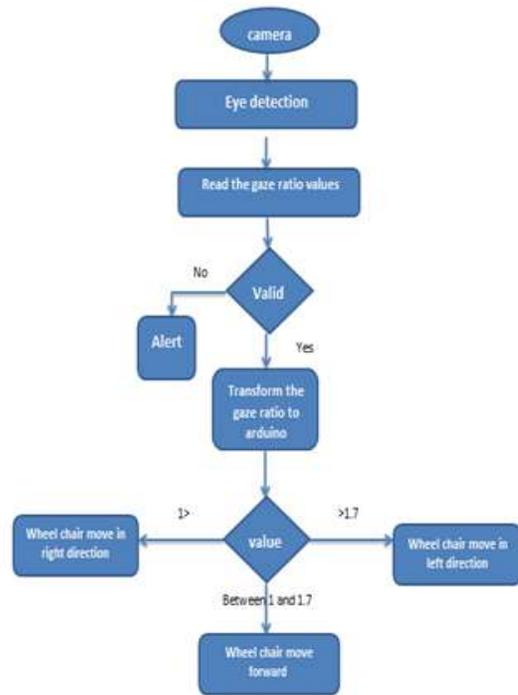


Fig. 3. The Proposed System Flowchart.

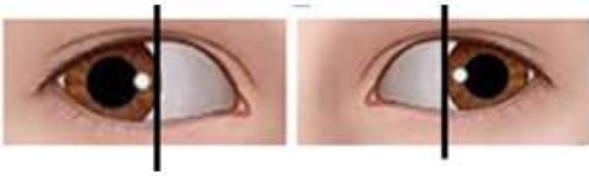


Fig. 4. Facial Detecting Eye Directions.

To detect the location of the sclera, convert the image into grayscale and calculate the number of white pixels on the left side and right side to both eyes. Then, calculate the gaze ratio to the right and left eye, which is the ratio between the left and white pixels and the right side of white pixels to each eye. Then, calculate the final gaze ratio to both eyes, which is the average value of the right and left gaze ratio. The values of the final gaze ratio could be computed using the following equation.

$$Gaze\ Ratio = \begin{cases} < 1, \text{Sclera in right direction} \\ > 1 \text{ and } < 1.7, \text{sclera in center} \\ > 1.7, \text{sclera in left direction} \end{cases} \quad (16)$$

These values are observed for all volunteers.

### C. The Wheelchair Design

The smart Wheelchair consists of three main components, as shown in Fig. 5.

Arduino Uno board.

L293D integrated circuit.

Two DC motors.

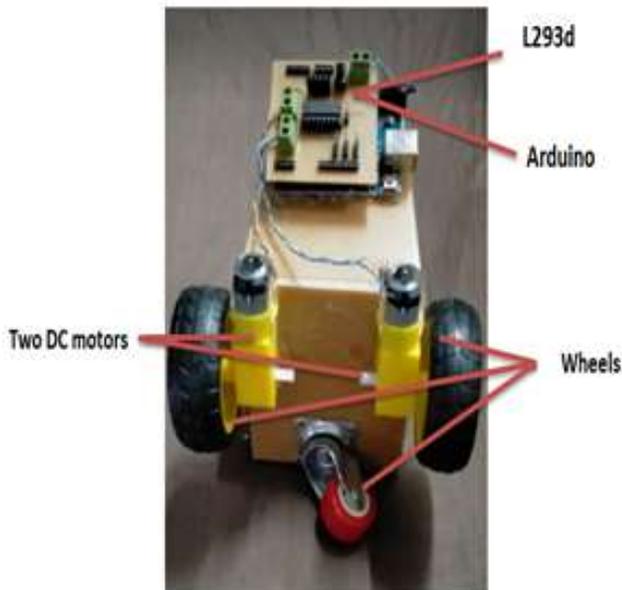


Fig. 5. Wheelchair Back Architecture.

1) *Arduino Uno board*: -is used to revise the eye direction from OpenCV and it is connected to the L293D integrated circuit.

2) *L293D integrated circuit*: -Using this L293D motor driver IC is very simple. The IC operates on the Half H-Bridge principle, a setup that is used to operate motors in both clockwise and anti-clockwise directions is the 1, H bridge. As stated earlier, this IC can run the two motors in any direction at the same time. In this experiment, L293D revise the control order (eye direction) from Arduino and control DC motors' control command [20].

3) *Two DC motors*: - All types of DC motors have some function, either electromechanical or electronic, to change the motor periodically's current path. In the experiment, each wheel in the chair is connected to the dc motor, and each motor receives the control order from the L293D integrated circuit to move the wheels in all directions. As shown in Fig. 6, Wheelchair forward architecture consists of a place to sit and a battery.



Fig. 6. Wheelchair Forward Architecture.

## VI. RESULTS AND ANALYSIS

The performance measure criteria in this study are: -

- 1) Calibration time.
- 2) Accuracy.

According to calibration time: During the testing time, it is very fast in the order of a millisecond. According to accuracy: Accuracy was calculated by counting the gaze detection for 3 persons (P1, P2, and P3). For each person, 30 gazes are tested, 10 for the right eye, 10 for the left eye, and 10 for the blinking eye, as shown in Table I.

Where:

$$Accuracy = \frac{\text{no of gaze detected}}{\text{total no od detection}} \quad (17)$$

TABLE I. THE CALCULATION OF THE SYSTEM PERFORMANCE

S.No	P1	P2	P3
<b>Total Detections No</b>	30	30	30
<b>Total no. of max left Detection</b>	8/10	8/10	10/10
<b>Total no. of max Right Detection</b>	10/10	10/10	10/10
<b>Total no. of max blinking Detection</b>	10/10	10/10	9/10
<b>Total no. of max Detection</b>	28/30	28/30	29/30
<b>Accuracy</b>	93.3%	93.3%	96.6%
<b>Error</b>	6.5%	6.5%	3.4%

The obtained average of accuracy is 94.4%. It is evident from the results that this study provided a noticeable improvement in eye tracking compared to previous studies. First, a Facial landmark with the Dlib technique one of the latest and more accurate techniques that are used in determining the face and eye was issued in 2014, besides it is clear from the results that the calibration time doesn't exceed a millisecond and accuracy is 94.4%, While the best results were previously monitored, the calibration time was 33.82 s and the accuracy was 90%. Table II presents all the proposed system components' costs.

TABLE II. THE COST OF THE MOST IMPORTANT TOOLS USED

Tool	Price
Arduino Board	10 \$
LD293D	2 \$
2 Dc Motor	1 \$
Wheelchair Model Design	5 \$
Total	18\$

## VII. CONCLUSION

Eye-tracking is an interesting field of research, especially to help the completely disabled to communicate better with the community and practice all life activities without anyone's help. For the proposed eye-tracking system, a successful circuit was built between a wheelchair and a webcam which was used to capture the user's face. A face detection algorithm was used using the landmarks of the face and the eyes based on Facial Landmarks points which divide the face into 68 points. The gaze ratio was calculated by tracking the pupil area. We have found that when this ratio is less than 1, the pupil of the eye is in the right direction and when it is greater than 1. The pupil is in the left direction, otherwise, it is in the middle, then these values were passed to the Arduino board, then to an L293D circuit, and then to the wheelchair wheels. The experiment was prepared using Python programming language and OpenCV Library. The experiment achieved better results than other wheelchairs, as the accuracy of the gaze detection was about 94.4%, the time was one microsecond, and the cost didn't exceed 15\$.

## REFERENCES

- [1] S. Kundu, O. Mazumder, P. K. Lenka and S. Bhaumik, "Hand Gesture Recognition Based Omnidirectional Wheelchair Control Using IMU and EMG Sensors", Journal of Intelligent & Robotic Systems, vol. 91, no. 3, pp. 529-241, 2018.
- [2] Shahin, M. K., Tharwat, A., Gaber, T., Hassanien, "A Wheelchair Control System Using Human-Machine Interaction: Single-Modal and Multimodal Approaches". Journal of Intelligent Systems, April-2019.
- [3] Putri Madona, Husna Khairun Nisa, Yusmar Palapa Wijaya and Amnur Akhyan, "The Design Of Wheelchair Systems With Raspberry Pi 3-Based Joystick Analog And Voice Control", Published under licence by IOP Publishing Ltd, Indonesia, November 2019.
- [4] Mamunur Rashid, Norizam Sulaiman, Mahfuzah Mustafa, Sabira Khatun, Bifta Sama Bari, Md jahid Hasan, Mamunur Rashid, "Recent Trends and Open Challenges in EEG Based Brain-Computer Interface Systems", Malaysia, 24 March 2020.
- [5] Dahmani, M., Chowdhury, M. E. H., Khandakar, A., Rahman, T., Al-Jayyousi, K., Hefny, A., Kiranyaz, S., "An Intelligent and Low-Cost Eye-Tracking System for Motorized Wheelchair Control". Sensors, July 2020.
- [6] <https://opencv.org>.
- [7] Fwa Hua Leong, "Deep learning of facial embeddings and facial landmark points for the detection of academic emotions", International Conference on Information and Education Innovations, July 2020.
- [8] Samar Gamal Amer, Sanaa A. kamh, Marwa A. Elshahed, "WRITING BASED EYE GAZE ESTIMATION SYSTEM", Sci.Int.(Lahore), 2020.
- [9] Anagha Dwajan B., Sowmya M S., Bhavani S., Priya Reddy N, Usha M R, "Eye Gaze Controlled Wheelchair", INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT), May 2020.
- [10] Mahendran Subramanian, Pavel Orlo, Suhung Park, "Gaze-contingent decoding of human navigation intention on an autonomous wheelchair platform", International IEEE/EMBS Conference on Neural Engineering (NER), May 2021.
- [11] Rupanagudi, S. R., Bhat, V. G., Gurikar, S. K., Koundinya, S. P., M. S., S. K., R., S., Satyananda, "A Video Processing Based Eye Gaze Recognition Algorithm for Wheelchair Control". 10th International Conference on Dependable Systems, Services and Technologies, 2019.
- [12] Ragini Singh1, Harsha Rani2, Junaid Hasan Khan3, Kumari Komal, "Eyeball Controlled Wheelchair, Journal of Science and Technology", 2020.
- [13] Hadish Habte Tesfamikael, Adam Fray, Israel Mengsteab, Adonay Semere, "Simulation of Eye Tracking Control based Electric Wheelchair Construction by Image Segmentation Algorithm", Journal of Innovative Image Processing, 2021.
- [14] Nutthanan Wanluk, Aniwat Juhong, Sarinporn Visitsattapongse, C. Pintavirooj, "Smart wheelchair based on eye tracking", The 2020 Biomedical Engineering International Conference, 2020.
- [15] Zheng, Y., Fu, H., Li, R., Lo, W.-L., Chi, Z., Feng, D., Wen, D. "Intelligent Evaluation of Strabismus in Videos Based on an Automated Cover Test". Applied Sciences, 2019.
- [16] MOHAMAD A. EID1, NIKOLAS GIAKOU MIDIS, "A Novel Eye-Gaze-Controlled Wheelchair System for Navigating Unknown Environments: Case Study With a Person With ALS", IEEE, 2016.
- [17] V. Kazemi J. Sullivan, "One millisecond face alignment with an ensemble of regression trees", IEEE Conference on Computer Vision and Pattern Recognition, 2014.
- [18] X. Cao, Y. Wei, F. Wen, J. Sun, "Face alignment by explicit shape regression", 2013.
- [19] Zhou, W., Gao, S., Zhang, L., Lou, X., "Histogram of Oriented Gradients Feature Extraction from Raw Bayer Pattern Images", IEEE Transactions on Circuits and Systems, 2020.
- [20] <https://www.instructables.com/Using-Motors-With-L293D-IC/>.