An Application Based on Artificial Neural Network for Determining Viewpoint Coordinates on a Screen

Bulent Turan[1], Halil Ibrahim Eskikurt[2], Mehmet Serhat Can[3]
[1]*Department of Computer Engineering Faculty of Natural Sciences and Engineering Gaziosmanpasa University,*
Tasliciflik Yerleskesi Tokat, Turkey
[2]*Department of Electrical and Electronic Engineering Faculty of Technology Sakarya University,*
Esentepe Campus 54187 Serdivan Sakarya, Turkey
[3]*Department of Electronics and Automation Zile Vocational School Gaziosmanpasa University,*
Cekerek Way 3.km Zile Tokat, Turkey
bulent.turan@gop.edu.tr

***Abstract*—This study used two different Artificial Neural Networks (ANN) to determine the point on a computer screen that the user is looking at. First, an ANN, called ANN1 was developed to identify the eye region of a laptop user from a webcam image. The computer screen was then divided into 57 × 32 blocks of 24 × 24 pixels. One hundred of these were randomly selected, and 20 images were taken by the integrated webcam while the user was looking at each point. The eye region was found on each image by ANN1. This eye region data was used to train another ANN, called ANN2. Twenty blocks were selected, and 20 different images were used as the test set. The coordinates of the block at which the user was looking were determined by ANN2. The deviations between the actual location coordinates and the location coordinates estimated by ANN2 were small. We conclude that our ANN2 was successfully trained to find the viewpoint of the user.**

***Index Terms*—EyeGaze; eye tracker; artificial neural networks; feature extraction.**

I. INTRODUCTION

New developments in image processing use the eye regions in an image [1] to acquire and use data related to the regions. The main fields in which such data are used include health monitoring [2], person recognition, security systems [3]–[6], staff follow-up systems, and EyeGaze systems. EyeGaze systems are used to identify a location on a screen that the user is looking at [7] and can be used to control a system. Beneficiaries of the EyeGaze systems being developed include paralysed patients and children with dyslexia. In these systems, devices called 'eye trackers' are used to follow the eye movements [7], and recent developments support simple and accurate human–computer interaction [8]. However, current eye trackers require physical contact with the user or beaming of infrared light at the eye [7]. Moreover, in EyeGaze systems, the positions of eyes are not evaluated only for that moment as eyes are constantly in motion.While scanning a scene, the eyes alternate between saccades and fixation [7]–[11], but even in fixation, there are small motions [7]. EyeGaze systems therefore need to record eye motions and to determine view points when the eye makes a saccade [7].

Adding hardware to an EyeGaze system increases the cost and affects the usage of the system. An alternative approach would be to determine points on a screen based on user images. This offers the prospect of easy to use, economical systems, encouraging the uptake of EyeGaze systems. Many studies are being conducted to identify the eye region from an image and to collect data about eye regions [12]–[16]. Most use Head Control or Head Tracking, in which the cursor is controlled by following head motion. In other studies, the cursor is directly controlled by eye movement [17], [18]. In a project called Opengazer supported by the European Commission, the Gatsby Foundation, and Samsung, cursor control was achieved through eye movement, without using an eye tracker [17]. In this system, coordinate determination is broken into three steps. In the first step (feature point selection), the user selects feature points on the face area by using a mouse. In the second step, (calibrating the system) the user directs the gaze at points that appear on the screen to calibrate the system**.** In the final step (tracking), feature points are followed to determine the coordinates of the point being looked at [17].

In another study, Yilmaz [18] demonstrated computer control using EyeGaze direction detection. The face region was first detected by the Adaboost algorithm, following which the eye region was detected by a Support Vector Machine (SVM) [18]. Gaze direction was detected using SVM and grayscale image features [18].

The systems to be controlled by looking have various features and control aims. Although these differences affect the devices that are used, they do not affect the demands made on image processing.

This study aimed to determine the coordinates of an image

Manuscript received 11 July, 2015; accepted 1 February, 2016.

on the computer screen that the user was looking at by processing images taken by an integrated webcam. Because no eye tracker was used, the approach differs from that of most EyeGaze systems.

II. MATERIAL

The study was implemented on a laptop computer with a 15.6 inch screen of 1366 × 768 resolution, a 2.13 GHz dual core Intel P7450 processor with 2.00 GB of RAM and a 1.3 Megapixel integrated webcam.

The screen of the laptop was divided into 57 × 32 (blocks), each of 24 × 24 pixels. From these blocks, 100 were selected for training and 20 were used to create a test set. A screen shot of the blocks is given in Fig. 1. The size of block is selected 24 × 24 pixels. For 15.6 inch screen, used in this study the button sizes are 30 × 18 pixels for minimize, 26 × 18 pixels for maximize, and 48 × 18 pixels for shutdown.

Users were asked to look at certain locations, and the integrated webcam captured 20 images from each user, giving a total of 2000 images for each user. A training database was created from these 2000 of the user images (480 × 640 × 3). A testing database was also created from 400 of the user images taken from 20 test locations. Areas of 48 × 160 belonging to an eye region were identified in each image for use in the training set and test data. As the number of pixels in each eye region was large, the image was reduced in size to 6 × 20 using the Matlab 'imresize' command. To allow the pixel values to be used as input to an artificial neural network (ANN2) they were converted into vector matrices with a size of 1 × 120. Because eye regions can start at many different coordinates in a user image, the starting coordinates of the eye region in each user image were determined and used as input values to the ANN2 with their pixel values, providing 122 data points for each image.

The images of the user were taken from a distance of 40–60 cm (the typical distance between the user and the screen), with an average distance of 50 cm, under identical light intensities and gaze angles to the screen.

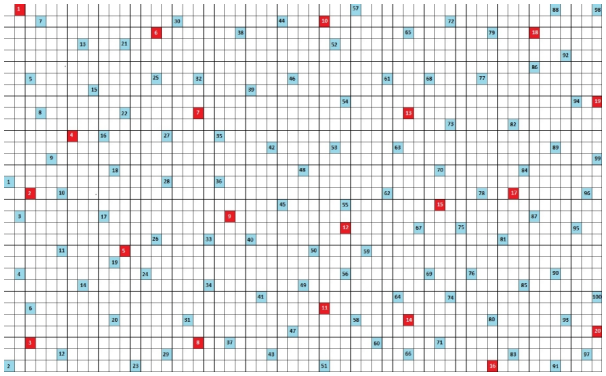


Fig. 1. Screen display of creating locations of training and test: a) locations with blue color are for training, b) locations with red color are for test.

A total of 11 users participated in the study. A MATLAB GUI interface was created to allow the images to be captured from the participants' own computers, making the study free from constraints of time and place. For all images, computers with the same screen resolution and size were used.

III. METHODS

The study followed the flowchart given in Fig. 2.

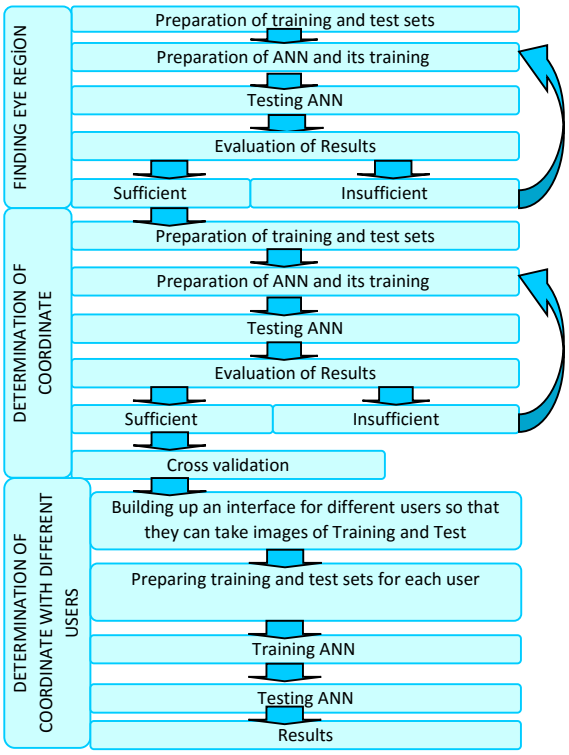


Fig. 2. Flowchart of the study.

A. Finding the Eye Region

Webcam images were taken of the laptop users looking at the screen under different light intensities, in different environments and from different perspectives. The 48 × 160 blocks that located the eye region were manually identified in these images. The data were converted into 1 × 120 vector matrices in MATLAB and used as the ANN1 training database.

Images used in the training set, and samples of the eye regions identified from these images, are shown in Fig. 3.

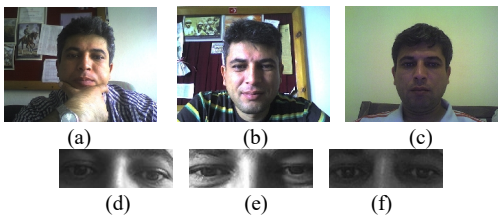
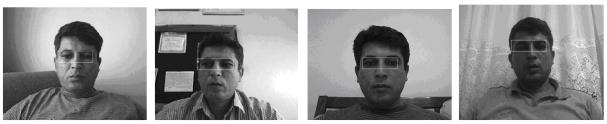


Fig. 3. The samples obtained from different environments and with different light intensity in (a), (b), and (c). The eye regions obtained from these images for the training set in (d), (e), and (f).

A feedforward backpropagation ANN model was used, consisting of one hidden layer and one output layer. The hidden layer comprised 150 neurons, while the output layer comprised one neuron. The network parameters were determined from the test results using the logarithmic sigmoid as the activation function in the hidden layer and the tangent sigmoid as the activation function in the output layer. A 'trainscg' parameter was used in algorithm learning. The learning and momentum coefficients were set at 0.6 and

0.8, respectively. During the training period, a total of 642 images were used, 107 of which correctly represented the eye region. For each input, an attribute vector was used to compose the network from 120 values. Each output was given the value 1 for a correct input and −1 for an incorrect input.



(a) (b) (c) (d)

Fig. 4. The samples for the test images applied by the ANN in (a), (b), (c), and (d).

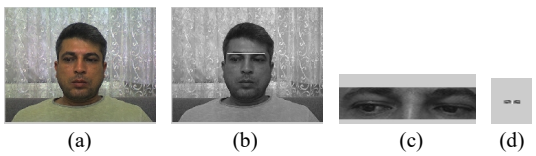
After training, the ANN1 was applied to the test set, achieving a success rate of 100 %. During the repeated testing stage, it was observed that the success rate varied from 90 % to 100 %. The network which obtained the best result was selected. Images obtained by the ANN1 from the test set are shown in Fig. 4.

B. Coordinate Detection

An image including the training and test locations was used as a background and 20 images were taken where the ANN1 had detected the eye region during the first stage. This procedure was repeated for each location. The images used for each user were captured when viewing the area with a fixed gaze. When the eyes saccade between two points, it is impossible to predict where the gaze will terminate. For this reason, images of saccades were not used. The images were taken as soon as the user focused on the key location.

When preparing the training set, the eye regions were first obtained from the image. The image of the eye region was converted to a 1 × 120 sized vector, and the initial coordinates of the eye region were added. For each image, a total of 122 data points were obtained. The test database was prepared in the same way as the training database.

Images used in the preparation of the training set, and the eye regions identified from these images, are shown in Fig. 5.



(a) (b) (c) (d)

Fig. 5. Image taken for the use in the training set (a); Image where the eye region was found (b); Eye region (c); and Eye region which size was minimized in Matlab (d).

For training, a feedforward backpropagation ANN model was used, comprising two hidden layers and one output layer. There were 90 neurons in the first hidden layer, nine in the second hidden layer, and two in the output layer. The network parameters were determined from the percentage of successes from the test results. The tangent sigmoid was used as the activation function in all layers, and the 'trainscg' parameter was again applied to the learning algorithm. In the ANN2, the learning and momentum coefficients were 0.7 and 0.3, respectively. For each input, an attribute vector of 122 values was applied to the ANN2 and the location expressed by the input was set as coordinates x (1–57) and y (1–32) in the output. The input and output values were scaled between 0–1 during the training.

After training, the ANN2 was applied to the test database and the area coordinates were determined for each test input. The area coordinates and test areas used during the capture of the test images are shown in Fig. 6.

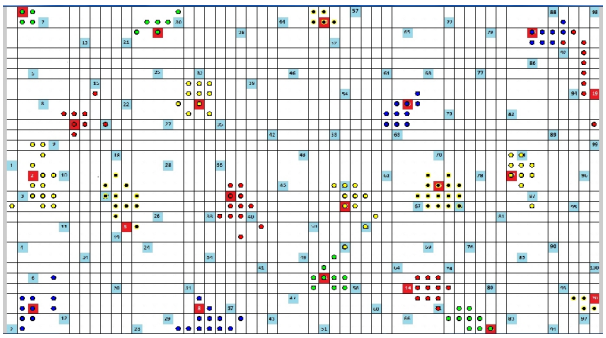


Fig. 6. The result obtained from the test data application of the network trained for the coordinate detection: (a) the red coloured square locations are the test areas, and (b) the round areas, showed with different colours around the test zones are the coordinates estimated by ANN.

A total of 400 test results were obtained, including 20 blocks for each test location. The red blocks show the test locations, and the test results for each location were printed as different coloured circles on the image at the centre of the location which they represented.

C. Coordinate Detection with Different Users

At this stage, to confirm that the ANNs would give comparable results for different users, an interface was developed using MATLAB GUI and the experiment was repeated with different users.

Fig. 7. The interface, prepared for the test subjects and deleting the invalid images before the recording.

In this phase of the study, the goal was to ensure that the ANN1 could correctly identify the eye region of the participant. Images were taken of the participant in different environments, under different light intensities, and when looking at the computer screen from different perspectives. The ANN1 training set was then expanded by adding a further 220 images: 50 images from two of the participants and 20 images from six of the participants. The ANN1 was trained again, now with a total of 1962 data groups (1962 × 122) from 327 user images. At this stage, one of the participants left the study and four new participants were added to the study. The rest of the program was conducted using eight male participants and three female participants. Although the ANN1 had not been trained with the eye regions of the four users who joined the study late, the eye regions of these users were also successfully identified. The image was sent to the screen by analysing it in real-time and finding the eye region of the user. A screen shot of the interface is given in Fig. 7.

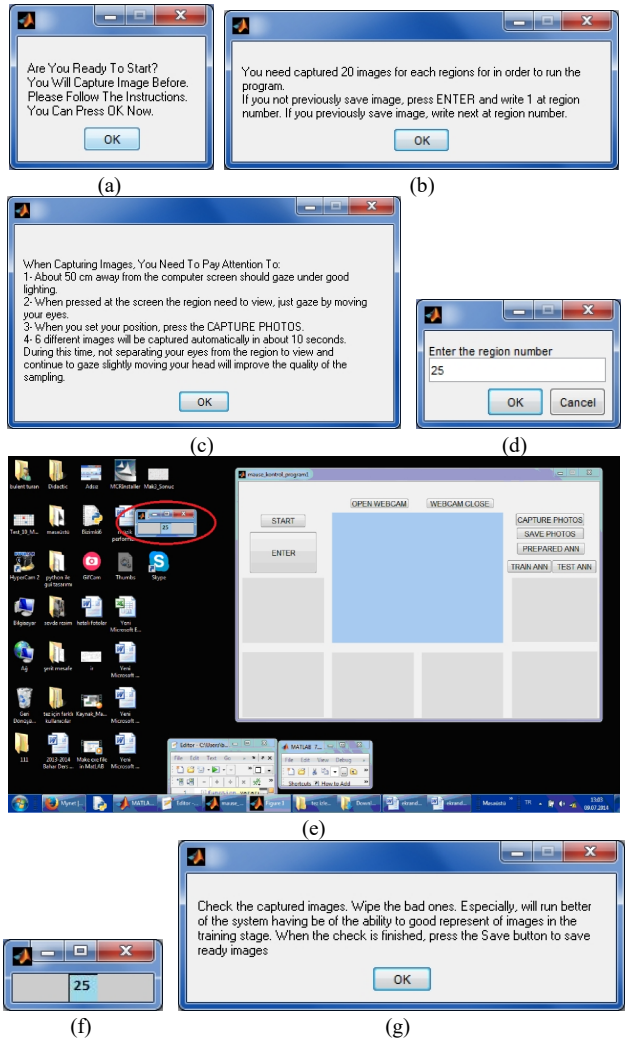


Fig. 8. Windows for directing the interface and displaying the areas on the screen.

This interface was used to guide the participant in the image acquisition process, in an attempt to avoid errors arising from incomplete data acquisition. The process required six real-time images to be taken, and participants were allowed to delete any image in which the eye region was not found, or in which other errors occurred. Deletion was done by clicking on an image in which the eye region was incorrectly identified, as shown in Fig. 7. Images that were saved were recorded by area numbers. When the number of records reached 20 for each location, the program stopped recording. Participants were required to record a total of 2400 images, including 100 for the training location and 20 for the test location. Because the ANN2 was being trained to detect gaze, the test results were negatively affected if images were taken from different perspectives.

The informations about the windows, in Fig. 8 are following:

a) It is seen when pressed “START” button.
b) It is seen when pressed “OK” button on the first window.
c) It gives information about how to take images.
d) It asks to enter the area number desired to take the image.
e) It shows the area which number is entered.
f) It shows the area in size of 24 × 24 related to its coordinates when the area number is entered.
g) It asks to check the captured images before recording.

Fig. 9. Images of the test subjects while looking at the 75th training area.

Images taken from different participants while looking at a given area are shown in Fig. 9.

IV. RESULTS

This study aimed to identify the locations that the user is looking at on a computer screen using a feedforward backpropagation network structure. The coordinates obtained from the ANN2 and the actual coordinates are shown in Fig. 6. It can be seen that the coordinates given by the ANN were very close to the true values.

The accuracy of the obtained coordinates was tested using root-mean-square error (RMSE), mean absolute percent error (MAPE) [19], [20] and mean absolute error (MAE) [21]. The equations for the two-dimensional data are given below:

$$RMSE = \sqrt{\frac{\sum (Y_x - Y_x')^2 + (Y_y - Y_y')^2}{n}}, \quad (1)$$

$$MAPE = \frac{1}{n} \sum \frac{\sqrt{(Y_x - Y_x')^2 + (Y_y - Y_y')^2}}{\sqrt{Y_x^2 + Y_y^2}} \times 100, \quad (2)$$

$$MAE = \frac{1}{n} \sum \sqrt{(Y_x - Y_x')^2 + (Y_y - Y_y')^2}, \quad (3)$$

where n is the number of estimates, Y_x is the real x coordinate, Y_x' is the estimated x coordinate, Y_y is the real y coordinate and Y_y' is the estimated y coordinate.

The values obtained by the ANN2 for each test image were located within a location, and the coordinates of this location were used to predict the location coordinates. The difference between the actual location coordinates and the predicted location coordinates was treated as the error. The error values were RMSE = 2.0069, MAPE = 0.0735 and MAE = 1.6648.

The deviations between the actual location coordinates and the location coordinates estimated by the ANN2 were small. The calculated MAE expected for any location is shown as a black circle in Fig. 10.

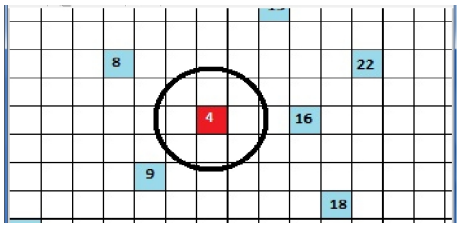


Fig. 10. The average of the results expected to be obtained from the 4th test area.

In the cross validation, one of the performance measures (MAPE) decreased, while two (RMSE and MAE) increased. However, these changes were not significant.

The performance values obtained by repeating the test and training processes of the ANN2 with the images from the participants are given in Table I.

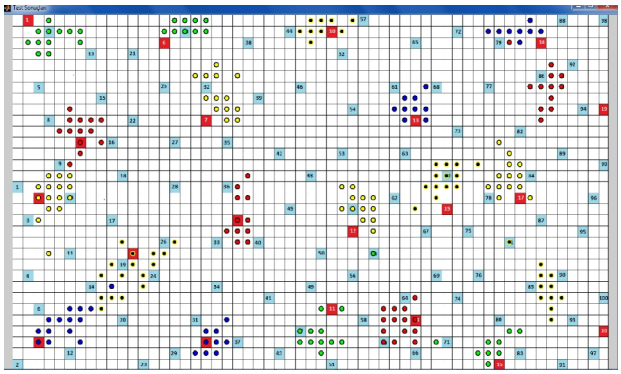
TABLE I. OBTAINED PERFORMANCE MEASUREMENT FROM THE TEST SUBJECTS.

	RMSE	MAPE	MAE
1	5.5245	0.28168	4.1566
2	6.0787	0.39425	5.2363
3	5.527	0.27387	4.6644
4	5.65	0.25946	5.0021
5	3.2202	0.1357	2.7083
6	7.7756	0.38961	7.0399
7	18.3218	0.929	15.7423
8	6.9441	0.35913	6.1379
9	7.823	0.299	6.5481
10	7.5598	0.50591	6.2501
11	5.7171	0.21133	4.9078

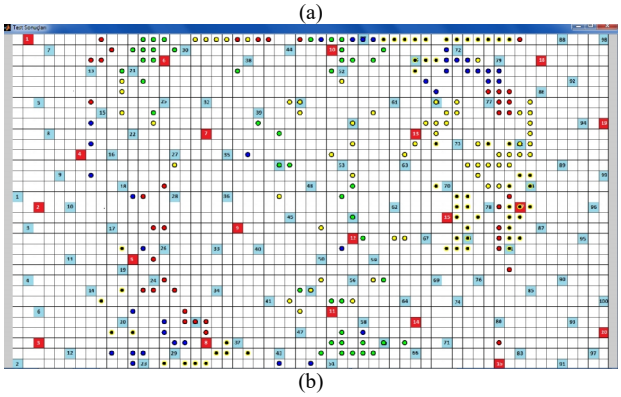
Fig. 11. The images of the researcher and the 5th and 7th test subjects who obtained the best and worst results while looking at the 5th training area (10 pieces).

Cross validation was performed by switching the input data of 20 randomly selected locations in the training database with 20 locations from the test. The results were RMSE = 2,6453, MAPE = 0,0715 and MAE = 2,0070.

Fig. 12. The images of the researcher and the 5th and 7th test subjects who obtained the best and worst results while looking at the 85th training area (10 pieces).



(a)



(b)

Fig. 13. The best (a) and worst (b) results obtained by the test subjects.

Examples of the images used in the study are given in Fig. 11 and Fig. 12. The results when the test images were applied to the ANNs developed using these example images are given in Fig. 13.

V. CONCLUSIONS

EyeGaze systems allow a computer to be used without a keyboard or other input device, by following the eye movements of the user.

This study attempted to identify a location on the computer screen based on the gaze of the user, without the

need for dedicated eye tracking devices. The eye region of the user was found from images taken by a webcam, and the pixel values of the eye region were used to train an ANN. The location identified by the ANN was very close to the true location. This was confirmed by statistical testing with values of 7.35 % for MAPE, 2.0069 for RMSE and 1.6648 for MAE. In Fig. 13, differences can be observed between the MAE results, with the lowest MAE result being very close to the MAE result obtained by the researcher. After analysing the user images, it was concluded that the deviations in the MAE results were caused by the head position of the user changing during image capture. Figure 11 and Fig. 12 show the images of the participants with the best and worst MAE result from the 5–85th training areas. It can be seen that the first participant did not change his angle of gaze, whereas the angle of the second participant changed.

These results suggest that the ANN2 developed in this study can be used to control a computer with a virtual keyboard by tracking the gaze of the user.

In previous studies Yilmaz [18] has reported a success rate of 97.2 % in the downward, upward and right and left gaze directions. However, gaze detection yielding viewpoint coordinates is a different matter, and the performance cannot be compared with that of the present study.

Opengazer [17] did not report success rates, so again the results cannot be compared with the present study, but the method used in the study offers certain advantages. Although our method did not require reference points or calibration during operation, the fact that each user had to repeat the training process was a disadvantage.

In the near future, the EyeGaze system will be able to use Webcam images instead of requiring an eye tracker. This will reduce costs by simplifying the use of the EyeGaze system, which will help expand its use.

A factor which adversely affected the results in this study was the continued movement of the eye during moments of fixation. The results could also be improved by increasing the number of locations used in the training set.

Based on our results, the use of data on both the eye region and the head position not only reduces the error rate, but also simplifies the use of the interface. The ANN can achieve improved performance when identifying the eye region of any computer user by developing the training database used.

The software is intended for use with certain screen sizes and resolutions and should therefore be rearranged to allow the users themselves to conduct the training and test procedures if the study is to be repeated for different screen sizes and resolutions.

REFERENCES

[1] S. Adwan, H. Arof, "A new approach for an efficient DTW in face detection through eyes localization", *Electronics and Electrical Engineering,* no. 2, pp. 103–108, 2011. [Online]. Available: http://www.eejournal.ktu.lt/index.php/elt/article/viewFile/154/113

[2] A. Aydemir, A. Uneri, "Nistagmusta gozun hizli fazli hareketlerinin tespit ve analizi (VNG) detection and analysis of quick phase eye movements ın nystagmus (VNG)", *Signal Processing and Communications Applications, IEEE 14th*, Antalya, 2006, pp. 1–4.

[3] R. Bodade, S. Talbar, "Dynamic ıris localisation: a novel approach suitable for fake ıris detection", *Int. Journal of Computer Information Systems and Industrial Management Applications (IJCISIM),* vol. 2, pp. 163–173, 2010.

[4] A. Husam, E. Lahrash, M. J. Nordin, "An enhanced segmentation approach for ıris detection", *European Journal of Scientific Research* vol. 59, no. 2, pp. 179–190, 2011.

[5] J. Greco, D. Kallenborn, M. C. Nechyba, "Statistical pattern recognition of the iris", in *17th annual Florida Conf. Recent Advances in Robotics (FCRAR),* 2004. [Online]. Available: http://www.mil.ufl.edu/publications/fcrar04/fcrar2004_iris.pdf

[6] J. Daugman, "New methods in ıris recognition", *IEEE Trans. on Systems, Man, And Cybernetıcs—Part B: Cybernetıcs,* vol. 37, no. 5, pp. 1167–1175, 2007. [Online]. Available: http://dx.doi.org/10.1109/TSMCB.2007.903540

[7] H. Coskun Gunduz, Goz Hareketlerinin Takibi ve Kaydedilmesi, 2005. [Online]. Available: http://ileriseviye.org/arasayfa.php?inode=eye-tracking.html (in Turkish)

[8] E. Ozcelik, E. Kursun, K. Cagiltay, "Goz Hareketlerini Izleme Yontemiyle Universite Web Sayfalarinin Incelenmesi" *Akademik Bilisim 2006 Bildiriler Kitapcigi,* Denizli, 2006 (in Turkish)

[9] O. Ince, M. Gokturk, "Guvenlik Sistemi Izleyici Personelinin Gorsel Tarama Davranisinin Analizi", *Akademik Bilsim Subat 2009 konferansi*, Urfa, 2009, pp. 699–704. (in Turkish)

[10] R. J. K. Jacob, K. S. Karn, *"Eye tracking in Human-computer interaction and usability research: Ready to deliver the promises", in the Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research, ed. by J. Hyona, R. Radach, and H. Deubel*, Amsterdam: Elsevier Science, 2003, pp. 573–605.

[11] A. Poole, L. J. Ball, *Eye tracking in human-computer interaction and usability research: current status and future prospects,* Encyclopedia of Human Computer Interaction, *Ghaoui, C. (ed.).* Hershey, PA: Idea Group, pp. 211–219.

[12] N. Erdogmus, J. L. Dugelay, "Automatic extraction of facial interest points based on 2D and 3D data", *SPIE 2011, Electronic Imaging Conf. on 3D Image Processing (3DIP) and Applications,* vol. 7864, San Francisco, California January, 2011, pp. 23–27.

[13] P. J. Batista, "Locating facial features using an anthropometric face model for determining the gaze of faces in ımage sequences", *Lecture Notes in Computer Science*, vol. 4633, pp. 839–853, 2007. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74260-9_75

[14] P. Kuo, J. Hannah, "An improved eye feature extraction algorithm based on deformable templates", in *IEEE Proc. Int. Conf. Image*, 2005, vol. 2, pp. 1206–1209. [Online]. Available: http://dx.doi.org/10.1109/icip.2005.1530278

[15] A. Majumder, L. Behera, V. K. Subramanian, "Automatic and robust detection of facial features in frontal face images", *UKSim 13th Int. Conf. Modelling and Simulation*, 2011, pp. 331–336. [Online]. Available: http://dx.doi.org/10.1109/uksim.2011.69

[16] G. Diamantopoulos, "Novel eye feature extraction and tracking for non-visual eye-movement applications", Ph.D. dissertation Submitted Dept. Of Electronic, Electrical and Computer Eng., Univ. Birmingham, 2010.

[17] E. Nel, O. Williams, R. Cipolla, "Opengazer: open-source gaze tracker for ordinary webcams", The Opengazer project is supported by Samsung and the Gatsby Foundation and by the European Commission in the context of the AEGIS project, 2012. [Online]. Available: http://www.inference.phy.cam.ac.uk/opengazer/

[18] M. C. Yilmaz, C. Kose, "Computer control and interaction using eye gaze direction detection", *2014 IEEE 22nd Signal Processing and Communications Applications Conf. (SIU 2014)*, Trabzon, 2014, pp. 1658–1661. [Online]. Available: http://dx.doi.org/10.1109/siu.2014.6830565

[19] V. A. Cho, "Comparison of three different approaches to tourist arrival forecasting", *Tourism Management*, vol. 24, no. 3, pp. 323–330, 2003. [Online]. Available: http://dx.doi.org/10.1016/S0261-5177(02)00068-7

[20] H. Ceylan, M Avan, "Turkiyede'ki Iş Kazalarının Yapay Sinir Agları ile 2025 Yılına Kadar Tahmini", *International Journal of Engineering Research and Development*, vol. 4, no. 1, pp. 46–54, 2012. (in Turkish)

[21] C. J. Willmott, K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance", *Climate Research*, vol. 30, pp. 79–82, 2005. [Online]. Available: http://dx.doi.org/10.3354/cr030079