System to Detect the Relative Distance between User and Screen

Namit Jain[1], Parcham Gupta[2], Ruchi Goel[3]

[1,2]*B.Tech. Student, Department of Computer Science and Engineering, MAIT, Delhi, India*
[3]*Assistant Professor, Department of Computer Science and Engineering, MAIT, Delhi, India*

***

Abstract - *Due to the COVID pandemic, there has been a rise in the use of digital platforms both in the domestic and commercial space which has led to a significant increase in the screen time for the masses as everything from school classes to professional conferences is being held in a digital way. The increased screen time has started to cause multiple health ailments like neck fatigue, back pain, eye strain etc. The major reason behind these being the inappropriate posture and distance maintained between eyes and the screen. Our system aims at indicating and notifying the user whether he is sitting at the proper distance as well as in a correct posture while sitting for long continuous hours in front of the display. Our approach is designed with the aim to tell the user if he is sitting at the proper distance and in the correct posture while sitting in front of the screen for long periods of time. The system takes an initial image of the user and poses it against the facial measurements extracted from the dataset. This enables the program to check if the facial dimensions of the live-image from the camera are more inclined towards the correct seating position and distance. Some sort of notifications and warnings will be given to the user if the distance is either too close or too far (which might cause eye strain and headache) or the sitting posture is not upright (which could lead to body fatigue).*

Key Words: Relative distance estimation, facial features, computer vision, real-time, screen distance, single camera

1.INTRODUCTION

The COVID-19 pandemic has forced everyone to carry on professional as well as personal work from their home. This means that the already excessive screen time has increased manifolds. We were already glued to our screens, but now it has become a necessity. We have classes, meetings, conferences, interviews, competitions, etc., all being conducted digitally that we can attend from the comfort of our homes. According to Agnes Wong **[1]**, online learning has undeniable benefits but looking at the flip-side, prolonged screen use can also prove to be adversarial. Even professionals are working right from their homes sipping their daily caffeine quota and attending to the most important of corporate chores.

This entire scenario has caused a huge increase in the eye strain caused due to working for long digital sessions. Teenagers and adults alike, spend a lot of time browsing, social media, entertainment, etc and are just about engaged with some sort of display for the majority portion of the day. In India, Bahkir and Grandee **[2]** surveyed more than 400 respondents having an average age of about 27 years. An increase in the screen time was reported by about 90% of the respondents, especially during the lockdown. It was observed that the screen time of more than 90% of respondents had nearly doubled during that period. The report also stated that most of the respondents had experienced at least one symptom related to digital device usage since the lockdown was declared. Further, half of the respondents who already experienced these symptoms reported an increase in the intensity and frequency of the symptoms.

In other nations as well, the relationship between computer screen exposure time and their visual as well as general symptoms was analysed and compared among students in different grades and universities. Liu, Ayi and Li **[3]** found that on average, the exposure time to computer screens for all the students was (5.20±3.06) hours/day and the mean exposure time increased with grade. Detailed analysis showed that the longer the exposure time was, the more serious the visual and general symptoms became. The frequently occurring symptoms were eyesight decline, vision blurring, eye dryness, fatigue and neck pain.

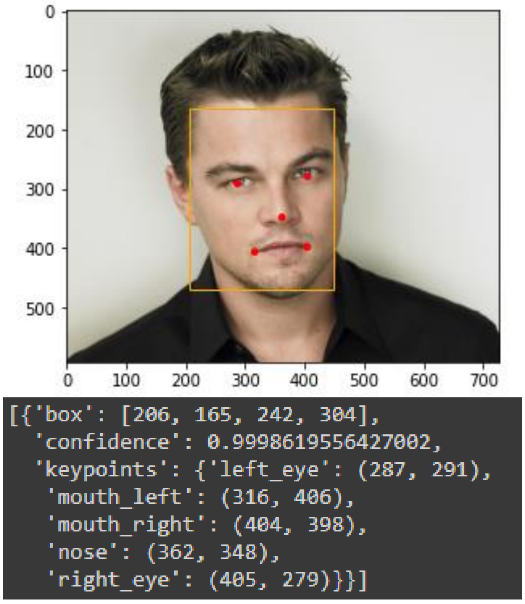
Our system aims to counter these ailments by using computer vision. The system detects if a person is sitting at the right position and at the appropriate distance from the screen and notifies the person accordingly. We also plan to execute some preventive tasks if the user provides suitable permissions.

The system works by detecting the face of the person using face recognition and detection systems. After the face is detected, and if a new user is recognized the system will ask to create a new profile, otherwise, we reload the recognized user's profile. Later, we detect the facial measurements and predict if the user is sitting at the appropriate distance from the screen and at an appropriate posture.

We use positioning and measurements of the major facial feature landmarks like eyes, nose, mouth etc. and other computations to predict if a user is sitting at the appropriate distance or not.

The system works as follows:

1. The new user's image is clicked and checked for facial measurements using results obtained from the MTCNN model **[4]**
2. These results are then stored and later used to find the scaling factors. **Fig. 1** shows the face detected with the facial landmarks alongside the exact pixel positions returned by MTCNN.



```
[{'box': [206, 165, 242, 304],
  'confidence': 0.9998619556427002,
  'keypoints': {'left_eye': (287, 291),
   'mouth_left': (316, 406),
   'mouth_right': (404, 398),
   'nose': (362, 348),
   'right_eye': (405, 279)}}]
```

Fig. -1: Result obtained from MTCNN detector

3. The user's relative position is then checked using the above information.
4. Suitable alerts are generated based on the user's positioning.

We also plan to give notifications and provide suitable reminders regarding recommended postural changes based on the person's sitting position. Further, the system can be integrated with different professional apps to perform suitable actions based on the person's sitting position and distance from the display. Also, we plan to check and provide customized feedback for the kids.

The paper is further divided into 5 sections: II-Literature Review, III-Dataset Used, IV-Proposed architecture (IVA-Face Detection, IVB-Standard Images, IVC-Relative Distance Estimation) and V-Conclusion and Future Scope.

2. LITERATURE REVIEW

According to Hossain & Mukit **[5]**, an estimation method based on feature detection can be used to calculate distance between camera and face. They also described the detection of face, other facial features and iris in an image sequence. This information can then be used to formulate an algorithm to determine the distance of face from the camera by using the distance between the centroid of the iris. An architecture for estimating edge and circular iris was presented. It was based on AdaBoost algorithm using Haar features. Canny and Hugo transform was used to determine the iris and to calculate the distance between the centroid of the iris. Later, distance estimations were made by using Pythagoras and similarity of triangles.

According to Dong et al. **[6]**, face distance can be estimated by using a single camera. A method based on monocular vision was proposed. The proposed system consisted of three major steps: feature regions were extracted and located in the face; pixel area of the characteristic triangle was calculated; and in the last step, the measurement formula was constructed using the pinhole camera calibration and area mapping.

According to Eastwood-Sutherland, C., & Gale, T. J. **[7]**, a quantitative metric for computer screen interaction can be obtained by using a wireless camera-based system. The relative position of the eyes with respect to the computer screen was monitored by using a stereo-camera based vision system. Infra-red LED based target markers were also used to increase the accuracy of the system. The use of multiple markers enabled a special LabVIEW program to detect the position and orientation of the head.

According to Dandil, E., & Çevik, K. K. **[8]**, stereo vision is the only way to develop computer vision systems that function similar to human vision. They used the face image distances to calculate the distance between objects and object dimensions. Disparity maps were used to evaluate the distance of the face to the screen. Initially, the proposed system involved extraction and detection of disparity maps and face regions respectively. The actual distance measurements involved calculation of shifts between frames obtained from stereo cameras. Later on, the distance values proposed by the system were compared with the actual distance values to analyse the performance and accuracy of the proposed system.

According to Kumar, M. S. et al. **[9]**, the depth between the front camera of modern devices and the user can be determined by using the monocular cameras possessed by such devices. Modern mobile devices like phones and tablets having front cameras facing the user and rear stereo cameras were considered for the proposed system. The depth thus calculated, was then used to calculate the factor for zooming the content on the display for better viewing or reading experience. A supervised learning algorithm was proposed to find the distance information using the facial landmark values obtained from the front camera of the device. This also reduced the error due to relative motion between the user and the device. The new user's face was registered via the rear stereo cameras and then, the depth analysis was done by the front camera using the trained Back Propagation Neural Network.

According to Rodríguez-Quiñonez, J. C et al. **[10]**, the stereo vision systems can be improved using optimized 3D measuring techniques. Stereo vision involves the use of two cameras, each viewing from a different angle and capturing images. The stereo pair images were used to detect the corner feature points which are then triangulated based on stereo correspondences. Analysis of object shapes, measurement of distances and angles, etc. can be done accurately using the 3D scanners along with the stereo vision technique. The proposed system focused on enhancing the stereo vision system by implementing the intensity pattern match method for distance measurement in real-time applications.

According to A. Saxena et al. **[11]**, monocular visual cues like gradients, texture variations, etc. can be captured and added to stereo vision cues to significantly increase the accuracy of depth estimates. The proposed system incorporated monocular cues together with any other pre-built stereo system. It was observed that unlike stereo cues which are majorly based on the difference between the two images, monocular cues are based on prior knowledge stored in the system and the structure of the image. Thus, integrating both of these into a single system was expected to obtain better depth estimates as compared to any of these cues taken alone.

3. DATASET USED

We have used Labelled Faces in the Wild dataset by Computer Vision Laboratory, University of Massachusetts **[12]**. The database consists of more than thirteen thousand images of faces collected from the web. The dataset was designed for studying the problem of unconstrained face recognition. Images are labelled with the name of the person present in the image. Two or more distinct photos are present for about 1600 people. However, since our system is not focused on facial recognition, we have used the image with higher detection confidence by MTCNN detector in case of multiple images associated with a single name. Each image present in the dataset is centred on a single face and is encoded in the RGB format. All the images are 250x250 pixels.

4. PROPOSED ARCHITECTURE

Working on dataset

Fig. 2 depicts the block diagram for dataset operations

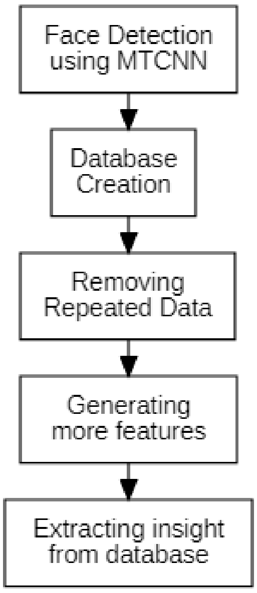


Fig. -2: Block diagram for dataset operations

1. Faces are detected using the MTCNN detector.
2. A database is created using the details returned by MTCNN on the images in the dataset.
3. Multiple records pertaining to the same user are deleted based on detector confidence level.
4. More statistical features like distance between both eyes, distance between eyes and mouth, etc. are added to the database by utilizing currently present features.
5. Mean values of different facial measurements are extracted from the database for further use.

Working of system

Fig. 3 depicts the block diagram for system workflow

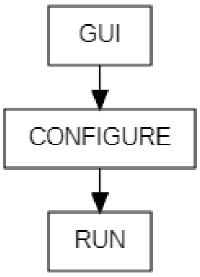


Fig. -3: Block diagram for system operations

A. **GUI**

a. The user details are obtained using the GUI along with basic description of the program. It has buttons to configure and run the code. **Fig. 4** shows a snapshot of the GUI interface.

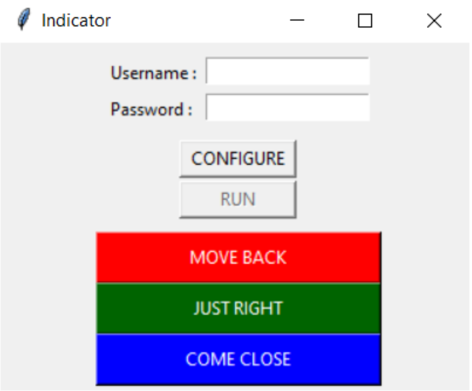


Fig. -4: GUI Layout

B. CONFIGURE

a. The login details are asked from the user. Alongside, an image is also clicked for additional validation.

b. The database is searched for information given by the user:

1. If found, the details are extracted from the database.
2. If not found, MTCNN and face-centred cropping is applied over the image to prepare it for further use. This image along with the details of the user is added to the database.

c. Either way, the details obtained from the database or from the new user are used to configure the system. **Fig. 5** shows the flow of control for the system configuration.

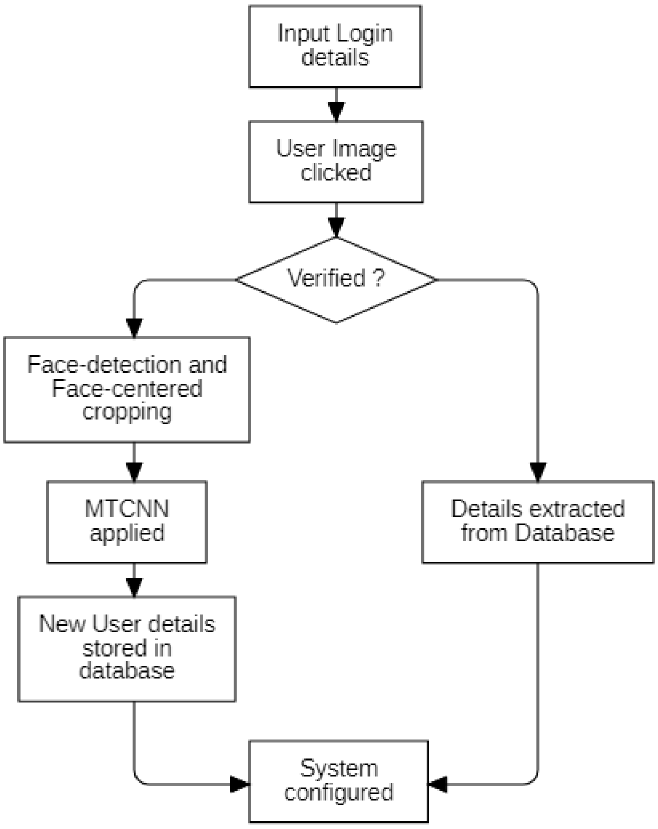

Input Login details
User Image clicked
Verified ?
Face-detection and Face-centered cropping
MTCNN applied
New User details stored in database
Details extracted from Database
System configured

Fig. -5: Block diagram for system configuration

C. RUN

a. Images are clicked after definite intervals of time.

b. These images are analysed for distance analysis using prior obtained facial measurements.

c. The facial measurements along with continuous clicked images are then used to evaluate the user's positioning in front of the display. **Fig. 6** shows the execution path for the proposed system.

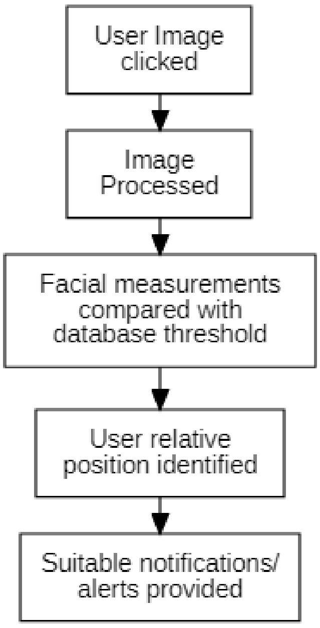

User Image clicked
Image Processed
Facial measurements compared with database threshold
User relative position identified
Suitable notifications/ alerts provided

Fig. -6: Block diagram for system execution

d. Suitable alerts/notifications along with respective instructions are displayed to the user. **Fig. 7** shows the alerts along with the captured user's image as generated by the system.

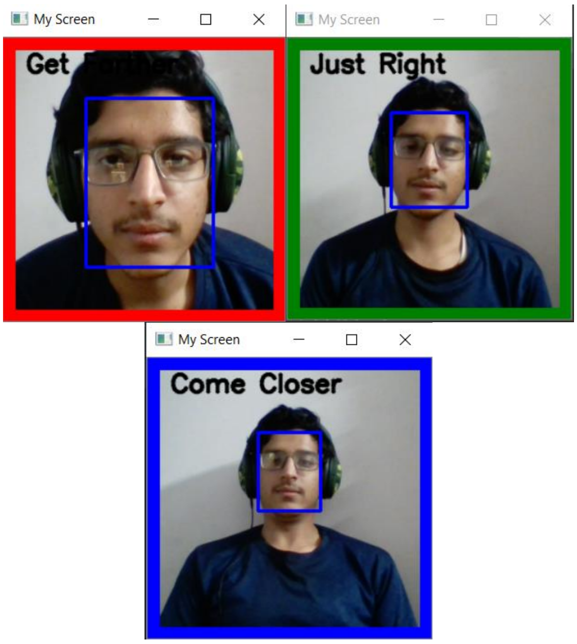


Fig. -7: Notifications provided by the system

4.1 Face Detection

Face detection was primarily required to obtain facial landmark points. Different libraries were tried and tested for better suited results in accordance with the proposed system. We primarily need face recognition, facial feature identification and facial landmark points as per the requirements. Following models were tried but MTCNN suited perfectly for the said requirements.

Haar Cascades: It was proposed by P. Viola & M. Jones in 2001 **[13]**. It works like a simple Convolutional Neural Network and is capable of extracting a lot of features from images. Adaboost is used to select the best features. However, haar cascades require a reasonably large unskewed dataset to train on and are prone to false-positive detections and require extensive parameter tuning when being applied for inference/detection. The accuracy is also not up to the desired level as we need the exact pixels to point to the facial features. Further, a number of haar cascades would need to be used to detect the face and then other facial features respectively. Thus, this model was not chosen for our desired purpose.

Dlib: Dlib was developed by D.E.King in 2009 **[14].** Dlib's frontal face detector works by using the features extracted by Histogram of Oriented Gradients. The next step involves passing these features through a Support Vector Machine. In spite of being a pretty advanced Convolutional Neural Network based face detector, Dlib is still not suitable to work with real time applications. Furthermore, it provides a very detailed mapping of the facial features which is not desired for our application. This will tend to increase computational cost and storage requirements. **Fig. 8** shows the facial feature points considered by Dlib for detection.

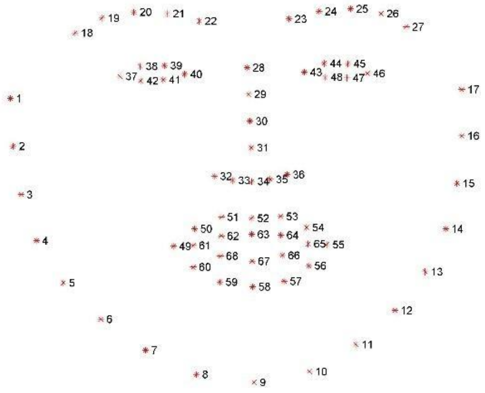


Fig. -8: The 68-points mark-up used by Dlib.

MTCNN: Multi-task Cascaded Convolutional Neural Network (commonly abbreviated as MTCNN) introduced by Zhang, K. et al. **[15]**, suits the purpose of our application. It detects the five key facial points as well alongside the face. It consists of 3 layers of Convolutional Neural Network: candidate windows are produced by a shallow Convolutional Neural Network in the first layer; proposed candidate windows are then refined in the second layer, by using a more complex Convolutional Neural Network; and lastly, the results and output the facial landmark positions are refined by the third Convolutional Neural Network. This model suits our requirements by also being computationally inexpensive. Further MTCNN is capable of producing real-time results on a CPU. Thus, images were fed to the MTCNN model during both the dataset processing (for generating threshold values) and also during the image processing while actual running of the application. The facial landmarks could be accurately used to generate required facial measurements and aspects.

4.2 Standard Images

The system relies on the camera hardware available with the user. Different laptops, webcams, etc. can have different field-of-view, focal length, resolution, etc. Thus, to nullify the hardware disparities as well as provide a standard user image to the system for further computations, the images are cropped by keeping the user face as the centroid. The image is fed to the MTCNN detector which provides the pixel values of the detected face-box. These values are then used to crop the image based on the formula:

x, y: top left pixel position of the detected face-box
w, h: width and height of the detected face-box
dim: lesser of the two dimensions of the image

```
$centre\_x = x + w / 2$
$centre\_y = y + h / 2$
$left = centre\_x - dim / 2$
if $left < 0$ then,
  $left = 0$
$right = left + dim$
if $right >$ horizontal dimension of image then,
  $right =$ horizontal dimension of image
  $left = right - dim$
$top = centre\_y - dim / 2$
if $top < 0$ then,
  $top = 0$
$bottom = top + dim$
if $bottom >$ vertical dimension of image then,
  $bottom =$ vertical dimension of image
  $top = bottom – dim$
```

center_x, center_y: pixel position of centre of the face-box
left, right, top, bottom: boundaries for cropping the image

This not only keeps the face in centre for better prediction of relative distance but also, eliminates the irregularities that might arise due to the off-centred position of the user in front of the system. Detection of irregular facial figures by the system are also avoided by cropping the image centred on the face. We have also used a human bust (upper body with head, neck and shoulders) stencil to highlight the expected user position for the system for best results. This helps to reduce the computational expense as face-cropping hardly cuts any required portion of the face detected.

Further, the dataset used had squared images, which is generally different from what most common hardware produce. So, face-cropping also modulated the images as per the requirements.

4.3 Relative Distance Estimation

We do not plan to predict the actual distance of the user from the display but want to give a rather relative insight about the user's position to ensure optimal and fatigue free positioning. The reason for this being, any person can't be asked or instructed to sit at a measured distance of 2 feet or so from the display but relative positioning in the terms of proximity from the display can be ensured. For this, we have used the T-shaped orientation of facial features like eyes, nose and mouth. **Fig. 9** shows the T-shaped orientation of the human face. This gives us some quantitative measurements to work with. The dataset was evaluated for these measurements. The measurements thus obtained, were treated as threshold values for further calculations.

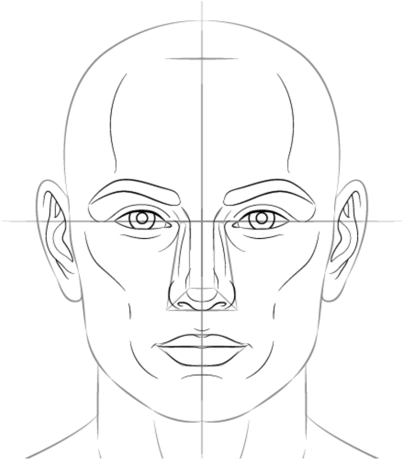


Fig. -9: Basic Structure of Human Face

But to nullify the hardware discrepancies and other factors, the threshold values were scaled by using the initial measured image from the user from expected ideal positioning. Factors are then calculated to scale the facial feature measurements obtained during real-time operation of the system to give accurate positioning estimates. Considering the T-shape also enables us to check for proper neck posture by checking the angle made by the T intersection with respect to the Cartesian axes. Calculating only two lateral values makes the system computationally inexpensive and quite robust as the facial features are naturally aligned in this way.

5. CONCLUSION AND FUTURE SCOPE

In this paper, a new approach for estimating user positioning in front of the display has been proposed. The estimation method is simple but gives reliable and accurate results for real-time applications. The proposed method doesn't focus on explicitly calculating the exact distance but gives a relative measure of closeness to the display which in turn promotes less fatigue and better efficiency. The dataset has been pre-analysed and the system can straight away work by using the threshold values alongside the user's initial camera snap. Therefore, the approach is fast enough for real-time application and reduces the computational cost. The system focuses the distance estimation based on the facial landmarks provided by MTCNN. Later, as we are only dealing with basic landmarks like eyes, nose and mouth, the system can generally work with most webcams. This also ensures handling only Euclidean distances which improves computational efficiency. However, the system relies on the accuracy of the landmarks generated by the MTCNN. The future work should focus on developing a custom algorithm to detect facial landmarks and compute the Euclidean distances simultaneously. Future scope also includes integrating more health norms like proper posture, viewing distance and viewing angles, etc. with the system to promote longer, healthier and efficient working hours.

6. REFERENCES

[1] Wong, A. S. (2021). Prolonged Screen Exposure During COVID-19—The Brain Development and Well-Being Concerns of Our Younger Generation. Frontiers in Public Health, 9.

[2] P Bahkir, F. A., & Grandee, S. S. (2020). Impact of the COVID-19 lockdown on digital device-related ocular health. Indian Journal of Ophthalmology, 68(11), 2378.

[3] LIU, X. T., AYI, B. L., & LI, Y. (2008). Impacts of Computer Screen Exposure on Visual and General Symptoms. Chinese Journal of School Health, 07.

[4] Zhang, N., Luo, J., & Gao, W. (2020, September). Research on Face Detection Technology Based on MTCNN. In 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA) (pp. 154-158). IEEE.

[5] Hossain, M. A., & Mukit, M. (2015, November). A real-time face to camera distance measurement algorithm using object classification. In 2015 International Conference on Computer and Information Engineering (ICCIE) (pp. 107-110). IEEE.

[6] Dong, X., Zhang, F., & Shi, P. (2014). A novel approach for face to camera distance estimation by monocular vision.

[7] Eastwood-Sutherland, C., & Gale, T. J. (2011, August). Eye-screen distance monitoring for computer use. In 2011

Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 2164-2167). IEEE.

[8] DANDIL, E., & ÇEVİK, K. K. (2019, October). Computer Vision based distance measurement system using stereo camera view. In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1-4). IEEE.

[9] Kumar, M. S., Vimala, K. S., & Avinash, N. (2013, September). Face distance estimation from a monocular camera. In 2013 IEEE International Conference on Image Processing (pp. 3532-3536). IEEE.

[10] Rodríguez-Quiñonez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-Lopez, M., Hernandez-Balbuena, D., Rascón, R., & Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. Opto-Electronics Review, 25(1), 24-32.

[11] Saxena, A., Schulte, J., & Ng, A. Y. (2007, January). Depth Estimation Using Monocular and Stereo Cues. In IJCAI (Vol. 7, pp. 2197-2203).

[12] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49, October, 2007.

[13] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). Ieee.

[14] King, D. E. (2009). Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research, 10, 1755-1758.

[15] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499-1503.