Computers, Materials & Continua
DOI:10.32604/cmc.2021.015478
Article

Tech Science Press

Eye Gaze Detection Based on Computational Visual Perception and Facial Landmarks

Debajit Datta[1], Pramod Kumar Maurya[1], Kathiravan Srinivasan[2], Chuan-Yu Chang[3,*], Rishav Agarwal[1], Ishita Tuteja[1] and V. Bhavyashri Vedula[1]

[1]School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Vellore, 632014, India
[2]School of Information Technology and Engineering, Vellore Institute of Technology (VIT), Vellore, 632014, India
[3]Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin, 64002, Taiwan
[*]Corresponding Author: Chuan-Yu Chang. Email: chuanyu@yuntech.edu.tw
Received: 23 November 2020; Accepted: 08 February 2021

Abstract: The pandemic situation in 2020 brought about a 'digitized new normal' and created various issues within the current education systems. One of the issues is the monitoring of students during online examination situations. A system to determine the student's eye gazes during an examination can help to eradicate malpractices. In this work, we track the users' eye gazes by incorporating twelve facial landmarks around both eyes in conjunction with computer vision and the HAAR classifier. We aim to implement eye gaze detection by considering facial landmarks with two different Convolutional Neural Network (CNN) models, namely the AlexNet model and the VGG16 model. The proposed system outperforms the traditional eye gaze detection system which only uses computer vision and the HAAR classifier in several evaluation metric scores. The proposed system is accurate without the need for complex hardware. Therefore, it can be implemented in educational institutes for the fair conduct of examinations, as well as in other instances where eye gaze detection is required.

Keywords: Computer vision; convolutional neural network; data integrity; digital examination; eye gaze detection; extraction; information entropy

1 Introduction

The COVID-19 pandemic was one of the worst global biological disasters in 2020 and affected everything around the world significantly. The traditional delivery of education had come to a standstill as the educational institutions had to be shut down due to the virus. As new, digitized education systems came into effect, new problems began to present themselves. While most educational institutions have started conducting online classes, the examinations are still problematic due to the lack of effective administration and invigilation approaches. At present, the online examinations are insufficiently monitored to detect malpractices by the students. A flawed examination system cannot accurately measure the students' abilities or be used to educate the students.

In this work, image processing and computer vision techniques are applied with the HAAR classifier to locate the eyes in an image. Once the eyes have been located, grey-scaling and threshold are applied to detect and extract both pupils from the image. The two Convolutional Neural Network (CNN) models–the AlexNet model and the VGG16 model—are applied to identify facial landmarks while maintaining data integrity. There are 68 facial landmarks in a face with six landmarks around the left eye and six around the right eye. For eye gaze detection, the Euclidean distances are used along with the twelve landmark points around both eyes. The system is validated by comparing the results to those from the traditional approach, which only considers eye detection using the HAAR classifier. The comparison is made using evaluation metrics that are accepted globally, namely the precision, recall, F1 score, specificity, sensitivity, and mutual information entropy scores. The proposed system can be used for online invigilation processes to automatically determine whether or not the students are participating in any malpractices, thus allowing for a fair examination process.

The implementation of this work can provide a smart examination system for both the teachers and students. The work presented here aims to ensure that the education process receives the recognition it deserves even when the world is in turmoil. The proposed eye gaze detection system may be applied to detect malpractices during examinations. It can also be applied where eye gaze detection is required, such as analyzing the user's interest level for an online course by tracking the person's eye gaze. The proposed system can also be applied to recommendation systems [1], where the eye gaze plays a vital role in understanding the users' interest. The eye gaze data can also be combined with web data mining concepts and technologies like clickstream, to determine the most relevant part of a web page and facilitate effective website design. The system also helps in analyzing the effectiveness of a steganographic image by capturing the region of interest for different users.

The following sections describe our work in detail, explaining the implementation along with the experimental results. Section 2 discusses related research works by several researchers. Section 3 briefly discusses the proposed system. Section 4 discusses the execution of each module in detail, as well as the concepts that have been used or proposed. Section 5 discusses the results and observations, along with visualizations through charts and plots for better understanding. Finally, Section 6 provides the conclusions based on the results and proposes future enhancements.

2 Related Works

The work of Qiu et al. [2] dealt with recognizing facial expressions based on the 68 different landmarks on the face. They successfully determined seven expressions, which are neutral, happy, fearful, sad, surprise, disgusted and angry expressions. They used distance vectors over the coordinates to design the algorithm for predicting the expressions accurately. The performance of their method was comparable to that of the known CNN models like the VGG-16 and ResNet models. According to the work of Dhingra et al. [3], non-verbal communication was detected by tracking the eye gaze. Their system detected non-verbal communication at a distance of around 3 meters for 28 testers. Using OpenFace, iView, and SVM, they had created a system that provided decent accuracies. Park et al. [4] used facial landmarks to provide insights into facial expression differences and determine a person smiles spontaneously or fakes a smile. They used Singular Value Decomposition (SVD) on the distances obtained to find the optimal transition to detect a spontaneous smile. The robustness of the system could be optimized further for real-time expression detection.

Su et al. [5] created an eye gaze tracking system that targets the pupil and the corners of the eyes to trace the eye movements. They used the inner corner-pupil center vector calculated in Euclidean distances. The model was based on the Deep Neural Network (DNN), and the ReLu was deployed as the activation layer. The system can be made to provide real-time responses. Iannizzotto et al. [6] provided a remote eye-tracking system to cope with interactive school tasks amid the COVID-19 pandemic. They used OpenCV to implement a video conferencing software in their model. In addition, the system is scaled to capture more faces in a single frame.

Sáiz Manzanares et al. [7] dealt with eye gaze tracking and data mining techniques for a sustainable education system. They used statistical analysis and a hierarchical clustering algorithm based on BIRCH and the Expectation-Maximization algorithm was deployed to test the system. The patterns obtained after data analysis were used to develop the education system.

According to the work of Tran et al. [8], the Interpersonal-Calibrating Eye gaze Encoder (ICE) was effective in extracting the eye gaze movement from a video. The paper used the ICE dynamic clustering algorithm and validated it using an infrared gaze tracker. Nevertheless, due to the errors caused by the clustering algorithm the data was not sampled appropriately and discrepancies were not removed. Dahmani et al. [9] developed a motorized wheelchair with an eye-tracking system. The paper described the CNN as the best choice for gaze estimation. Ujbanyi et al. [10] tracked eye movements to determine where the person is looking at. Their work captured the eye movements by performing human computer interactions-based motoric operations. They focused on how the eye-tracking methodology affected the cognitive processes and did not elaborate on the landmarks used to detect eye movements.

Zhu et al. [11] investigated a gaze tracking system with superior accuracies. They developed new algorithms to identify the inner eye corner and the focal point of the iris with sub-pixel precision. Their method established a sub-pixel feature tracking algorithm that upgraded the precision of head pose estimation. However, their work was only accurate for high-resolution images. According to Chennamma et al. [12], there are four different oculographic methods: Electro-oculography, scleral search coils, infrared oculography, and video oculography. Depending on the hardware used for tracking the eye movements, video oculography were further classified into a single camera eye tracker and a multi-camera eye tracker. Moreover, there are two gaze estimation methods, namely feature-based gaze estimation and appearance-based gaze estimation. The feature-based approaches are further categorized into model-based and interpolation-based approaches. Their work mainly highlighted the need for standardizing the metrics of eye movements. Datta et al. [13,14] provided a better way of implementing the Convolutional Neural Network (CNN) models for image classification. The traditional method of implementing the CNNs was time-consuming; moreover, the performance and hardware utilization could be further improved. By utilizing parallel computation with Ray, the authors reduced operating time and increased efficiency.

3 Proposed Work

Over the years, several advancements and research have been done in computer vision and image processing. With the growing number of technologies and scientific knowledge base, there have been many research works to predict the eye gaze direction. The traditional eye gaze detection method was based purely on image processing, which provided decent evaluation. Infrared was often used for eye gaze detection, but it is considered to be harmful in several research works [15,16]. In this paper, we propose an approach that is more effective than the traditional eye

gaze detection method. The proposed method calculates the distances between facial landmarks to provide more accurate eye gaze detection.

The architecture diagram in Fig. 1 shows the different eye gaze detection steps described in this work. Image processing and computer vision techniques, along with the HAAR classifier, are used to detect and mark around the pupils with a red circle, referred to as a blob. The twelve facial landmarks around the eyes are also detected and extracted from the dataset to train the Convolutional Neural Network (CNN) models, namely the AlexNet and VGG16 models. The reason multiple CNN models were chosen is to evaluate the accuracies of the models. Next, the output gaze is predicted from the live video capture data using the Euclidean distances. This work compares the proposed system's evaluation metric scores to that of the traditional system which uses only image processing and computer vision techniques to predict the eye gaze.

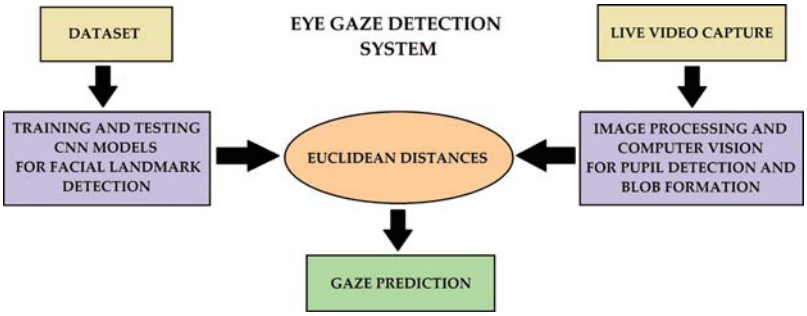

Figure 1: Architecture of the eye gaze detection system

4 Implementation

4.1 Dataset, Hardware and Background Description

The dataset for the proposed eye gaze detection system has been used to locate the twelve facial landmarks around the eyes. The facial landmark dataset consists of the 68 facial landmarks with their x and y coordinates in Comma Separated Value (CSV) format and a total of 5770 colored images of different faces.

The work has been carried out on an HP Spectre×360 Convertible 15-ch0xx workstation, with an x64-based Intel® Core™ i7-8550U processor. Additionally, the system configuration also includes 16 GB RAM, a 64-bit operating system, as well as a touch and pen support.

The environment in which this work has been implemented is illuminated by a white Phillips light source with a Light Emitting Diode (LED) tube light.

4.2 Facial Landmark Detection Using the CNN

A human can quickly identify a face or other parts of the face like the eyes or nose in an image or video, but a computer needs more information to do the same. To identify a face in an image, face detection methods are used to determine the human face's location in the image [17,18]. The location is often returned as a bounding box or coordinates values. To find smaller features such as the eyes and lips, facial landmarks are used to localize and extract the required coordinates of the facial parts from the bounding box. There are 68 landmarks on the face representing the face's salient features such as the eyes (both left and right), eyebrows (both left and right), nose, mouth, and jawline. These landmarks represent the points that are considered

to be essential in the localization and extraction of facial features. Each of these 68 landmarks can be represented uniquely by index numbers from 1 to 68. Each of these landmarks has unique coordinates represented as *(x, y)* values. The Convolutional Neural Network (CNN) is a branch of Deep Neural Networks used for image recognition and classification [19,20]. In the CNN, an image is taken as an input; it is further assigned numerical weight and bias values to enhance certain image features. At the end, the images are classified into particular groups based on the probabilistic values. The process is carried out by passing the image into a convolution layer, pooling, and flattening the obtained output through a fully connected layer. In the convolution layers of CNN, the image RGB values are updated [14] according to Eq. (1).

$$G[m,n] = (f * h)[m,n] = \sum_{j} \sum_{k} h[j,k] f[m-j, n-k] \tag{1}$$

In Eq. (1), f is the input image and h represents the filter. The dimensions of the resultant matrix are m by n. For image classifications, the CNNs are usually preferred over other neural networks due to their feature extraction ability and minimum pre-processing requirement. The AlexNet and VGG16 architectures are used to train the model for facial landmark detection.

4.3 Evaluation of the Models

The CNN models used for detecting the facial landmarks are evaluated based on several performance evaluation metrics, namely the precision score as shown in Eq. (2), the recall and sensitivity scores as shown in Eq. (3), the F1 score as shown in Eq. (4), the specificity score as shown in Eq. (5).

$$Precision, P = \frac{TP}{TP + FP} \tag{2}$$

$$Recall, R = Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ Score = \frac{2 * P * R}{P + R} \tag{4}$$

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

In these equations, TP, TN, FP and FN respectively represent the true positive, true negative, false positive and false negative values. A true positive represents that the predicted and the actual classifications are both positive; a true negative represents that the predicted and the actual classifications are both negative. A false positive is where the predicted value is positive but the actual value is negative, and vice versa for a false negative. For all of these evaluation metrics, higher scores values are better than lower score values.

4.4 Pupil Detection Using Computer Vision and Image Processing

In this work, image processing techniques are implemented to detect the pupil in a live video stream which is composed of a series of images. Since this sub module's main goal is to extract the pupil from an image, the pupil needs to be detected first. For pupil detection, the first step is to identify the face and the eyes. The HAAR classifiers are used for these identifications. The HAAR classifiers are XML files for facial feature detection. After the eyes are detected, pupil

extraction is performed followed by thresholding. After thresholding, the pupils are extracted and the blobs are drawn around the pupils. The proposed system is implemented using Python in conjunction with the OpenCV library.

For eye detection using OpenCV, the HAAR classifiers are used to detect the objects in the given image or video. OpenCV has built-in classifiers, haarcascade_frontalface_default.xml and haarcascade_eye.xml, for face and eye detection, respectively. The classifiers perform the detection process based on certain properties [21,22]. These HAAR classifiers are in-built in OpenCV and are trained using many positive (images containing faces or eyes) and negative (images without faces or eyes) datasets. If a face is detected using the HAAR face classifier, an ROI (Region of Interest) is created for the face and eye detection is applied to the ROI. The ROI is specified by coordinates which represent the bounding regions of the face. For eye detection, the surrounding areas also need to be considered, such as areas containing the eyelids, eyelashes, and so on. Therefore, accurate demarcation is required before eye detection.

4.5 Gaze Prediction Using Euclidean Distances

The Euclidean distance calculates the segment's length between any two points in space. The calculation for the Euclidean distance is shown in Eq. (6).

$$dist\,(p,q) = \sqrt[2]{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{6}$$

As shown in Eq. (6), the distance between any two points on a plane $p(x1, y1)$ and $q(x2, y2)$ is calculated as $dist(p, q)$.

The conceptual representation of the eye can be seen in Fig. 2. Once the blob is formed using OpenCV, the center of the blob is calculated as the center of the pupil, $ep(x_{ep}, y_{ep})$, as shown in Fig. 2. The eye-landmarks are detected using facial landmark detection. Each eye has six landmark points $e1(x_{e1}, y_{e1})$, $e2(x_{e2}, y_{e2})$, $e3(x_{e3}, y_{ec3})$, $e4(x_{e4}, y_{e4})$, $e5(x_{e5}, y_{e5})$ and $e6(x_{e6}, y_{e6})$ around it. An assumed center $ec(x_{ec}, y_{ec})$ is determined based on the six landmark locations according to Eqs. (7) and (8), as shown in Fig. 2.

$$x_{ec} = \frac{x_{e1} + x_{e2} + x_{e3} + x_{e4} + x_{e5} + x_{e6}}{6} \tag{7}$$

$$y_{ec} = \frac{y_{e1} + y_{e2} + y_{e3} + y_{e4} + y_{e5} + y_{e6}}{6} \tag{8}$$

In Eqs. (7) and (8), x_{ec} is the x-coordinate and y_{ec} is the y-coordinate. After the assumed center is calculated and the pupil's center is located, the projections of ep on the axes, ep_x and ep_y, are considered. As shown in Fig. 2a threshold is decided. In this work, the threshold is selected as 40%. The threshold determines the part of the eye region that is considered as the center, the distances from ec to $e1$, $e4$, et, and eb are calculated using Eq. (6), as $dist(e1, ec)$, $dist(e4, ec)$, $dist(et, ec)$, and $dist(eb, ec)$, respectively. The x coordinate of et can be calculated using Eq. (9), the y coordinate of et can be calculated using Eq. (10), the x coordinate of eb can be calculated using Eq. (11), and the y coordinate of eb can be calculated using Eq. (12).

$$x_{et} = \frac{x_{e2} + x_{e3}}{2} \tag{9}$$

$$y_{et} = \frac{y_{e2} + y_{e3}}{2} \tag{10}$$

$$x_{eb} = \frac{x_{e5} + x_{e6}}{2} \tag{11}$$

$$y_{eb} = \frac{y_{e5} + y_{e6}}{2} \tag{12}$$

In Eqs. (9)–(12), *et* and *eb* are the assumed top and bottom points of the eye as shown in Fig. 2. Once these values are calculated, the system calculates the distances from *ec* to the projections on the axes to detect the eye gaze using Euclidean Distances, $dist(ep_x, ec_x)$, and $dist(ep_y, ec_y)$. The obtained distances are compared with the distances based on the obtained threshold and the distances from the assumed center of the eye, *ec*. The following scenarios are considered:

(1) $ep_x > ec_x$ and $dist(ep_x, ec_x) <= 40\%$ of $dist(et, ec)$
(2) $ep_x < ec_x$ and $dist(ep_x, ec_x) <= 40\%$ of $dist(eb, ec)$
(3) $ep_y > ec_y$ and $dist(ep_y, ec_y) <= 40\%$ of $dist(e4, ec)$
(4) $ep_y < ec_y$ and $dist(ep_y, ec_y) <= 40\%$ of $dist(e1, ec)$
(5) $ep_x > ec_x$ and $dist(ep_x, ec_x) > 40\%$ of $dist(et, ec)$
(6) $ep_x < ec_x$ and $dist(ep_x, ec_x) > 40\%$ of $dist(eb, ec)$
(7) $ep_y > ec_y$ and $dist(ep_y, ec_y) > 40\%$ of $dist(e4, ec)$
(8) $ep_y < ec_y$ and $dist(ep_y, ec_y) > 40\%$ of $dist(e4, ec)$

If any one of conditions 1 to 4 holds true, the eye gaze is considered to be at the "center", otherwise, if 5 and 7 are true, the gaze is considered to be at the "top right," if 5 and 8 are true, the gaze is considered to be at the "top left", if 6 and 7 are true, the gaze is considered to be at the "bottom right", and if 6 and 8 are true, then the eye gaze is considered to be at the "bottom left."

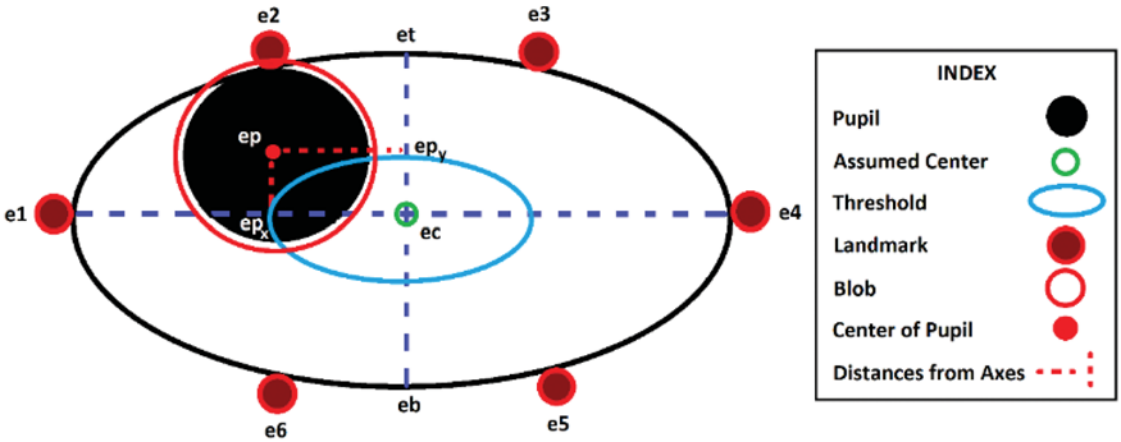

Figure 2: Conceptual representation of the eye in the eye gaze detection

5 Results and Discussion

The eye gaze tracking system is implemented with image processing functions from OpenCV. Two different convolutional neural networks (CNN) models are used for facial landmark detection, the AlexNet and VGG16 models. There are 68 distinct facial landmarks, and each landmark

is defined by a set of x and y coordinates. In this work, we consider the 12 landmarks around the eyes. The indices allocated to these landmarks are 37 to 48. According to the dataset, the relevant columns range from 73 to 84 for the right eye and 85 to 96 for the left eye since each landmark has its associated x and y values.

The created models are benchmarked with several performance evaluation metrics, namely the precision, recall, F1 score, specificity and the sensitivity scores [23]. The true positive, true negative, false positive and false negative scores are calculated using a confusion matrix. The two CNN models for facial landmark detection are evaluated using these metrics. The models are trained and tested using the same dataset, which has been adapted from the Kaggle platform.

The system is validated based on the five-evaluation metrics. Tab. 1 shows the evaluation metric scores obtained by the AlexNet CNN model for the twenty-four labels, twelve x-axis and twelve y-axis labels, which represent the twelve facial landmarks around both eyes. Tab. 2 shows the various evaluation metric scores obtained by the VGG16 CNN model.

Table 1: Evaluation metric scores of the AlexNet CNN model

Label	Precision	Recall	F1-Score	Specificity	Sensitivity
73	0.717562225	0.811416472	0.761608792	0.743179032	0.811416472
74	0.711017113	0.799303096	0.752579720	0.813571477	0.799303096
75	0.855053989	0.725997235	0.785258342	0.737910224	0.725997235
76	0.798551410	0.728477370	0.761906571	0.792186928	0.728477370
77	0.714153658	0.829278272	0.767422392	0.761239953	0.829278272
78	0.692026399	0.681052731	0.686495714	0.822699293	0.681052731
79	0.857306836	0.773476943	0.813237266	0.742472363	0.773476943
80	0.856205515	0.833140778	0.844515694	0.748351182	0.833140778
81	0.744953942	0.762219430	0.753487793	0.837746938	0.762219430
82	0.790058473	0.672053872	0.726294197	0.734426802	0.672053872
83	0.684057553	0.679573170	0.681807988	0.733502436	0.679573170
84	0.681351175	0.815410911	0.742377413	0.722432706	0.815410911
85	0.811484764	0.848149141	0.829411961	0.804952065	0.848149141
86	0.779038135	0.677486294	0.724722014	0.702174114	0.677486294
87	0.770787186	0.694470107	0.730641181	0.746307148	0.694470107
88	0.784990193	0.784431198	0.784710596	0.684482508	0.784431198
89	0.739136247	0.773420951	0.755890038	0.853802587	0.773420951
90	0.850102683	0.770637338	0.808421906	0.814135282	0.770637338
91	0.699447937	0.849046833	0.767021068	0.690431673	0.849046833
92	0.707333320	0.774463942	0.739378005	0.756411173	0.774463942
93	0.767906637	0.727391893	0.747100397	0.713192002	0.727391893
94	0.828142026	0.676443418	0.744645278	0.848524704	0.676443418
95	0.825983717	0.833009272	0.829481618	0.715881064	0.833009272
96	0.685241393	0.701316829	0.693185924	0.800200211	0.701316829

The values in the tables are plotted using the Python library matplotlib for better visualization and insights into the inherent pattern of the results.

Table 2: Evaluation metric scores of the VGG16 CNN model

Label	Precision	Recall	F1-score	Specificity	Sensitivity
73	0.840173185	0.859437757	0.849696292	0.751910922	0.859437757
74	0.729921232	0.763873812	0.746511667	0.886822345	0.763873812
75	0.793356873	0.758943082	0.775768508	0.845916328	0.758943082
76	0.816741018	0.860641426	0.838116741	0.725223295	0.860641426
77	0.769738117	0.895913564	0.828046857	0.780446281	0.895913564
78	0.865676922	0.872141454	0.868897164	0.788532870	0.872141454
79	0.837540478	0.794106688	0.815245488	0.846806231	0.794106688
80	0.899704356	0.868857047	0.884011682	0.781056553	0.868857047
81	0.833438053	0.743710083	0.786021642	0.819588918	0.743710083
82	0.848700303	0.889776732	0.868753244	0.762694238	0.889776732
83	0.853125709	0.781587381	0.815791215	0.771867038	0.781587381
84	0.734068860	0.882020136	0.801272105	0.790558188	0.882020136
85	0.773391255	0.852308141	0.810934253	0.878940096	0.852308141
86	0.783518052	0.888087567	0.832532067	0.813805832	0.888087567
87	0.721974099	0.753350505	0.737328654	0.784651758	0.753350505
88	0.720784512	0.883174827	0.793759195	0.807746934	0.883174827
89	0.748865008	0.890039094	0.813371731	0.843960654	0.890039094
90	0.808155491	0.745915407	0.775789101	0.865443502	0.745915407
91	0.847564862	0.823486311	0.835352110	0.755350522	0.823486311
92	0.732644249	0.795227154	0.762653977	0.893290636	0.795227154
93	0.761928369	0.758566128	0.760243531	0.861134840	0.758566128
94	0.847773429	0.797238924	0.821729971	0.830346337	0.797238924
95	0.734934071	0.749501551	0.742146332	0.838393404	0.749501551
96	0.886380720	0.726413530	0.798463843	0.882930523	0.726413530

The plots in Fig. 3 visualize the comparison of the different evaluation metric scores between the AlexNet model and the VGG16 model. From Fig. 3a, it can be inferred that for most of the class labels, the classification for the VGG16 has better precision than that of the AlexNet. The assessment of recall (sensitivity) scores between the AlexNet model and the VGG16 model can be observed in Fig. 3b. The two scores are always equal, thus they are plotted in the same figure. From Fig. 3b, it can be seen that for most of the times, the classification has better recall and better sensitivity scores for the VGG16 than those of the AlexNet, irrespective of the class labels. The F1 scores between the AlexNet model and the VGG16 model are visualized in the plot shown in Fig. 3c. It can be seen from Fig. 3c that for most of the class labels, the classification for the VGG16 has better F1 score values than that of the AlexNet. The specificity scores of the AlexNet model and the VGG16 model are compared and plotted in Fig. 3d. Again, it can also be observed that the VGG16 has better specificity scores compared to that of the AlexNet for most of the class labels.

The evaluation metric scores of the AlexNet and the VGG16 models are respectively aggregated and plotted in Figs. 4a and 4b for better visualization. It can be seen from Figs. 4a and 4b that the scores for the two models fluctuate for the different class labels. Since the VGG16 model has comparatively better results, we use the VGG16 CNN model in this work.

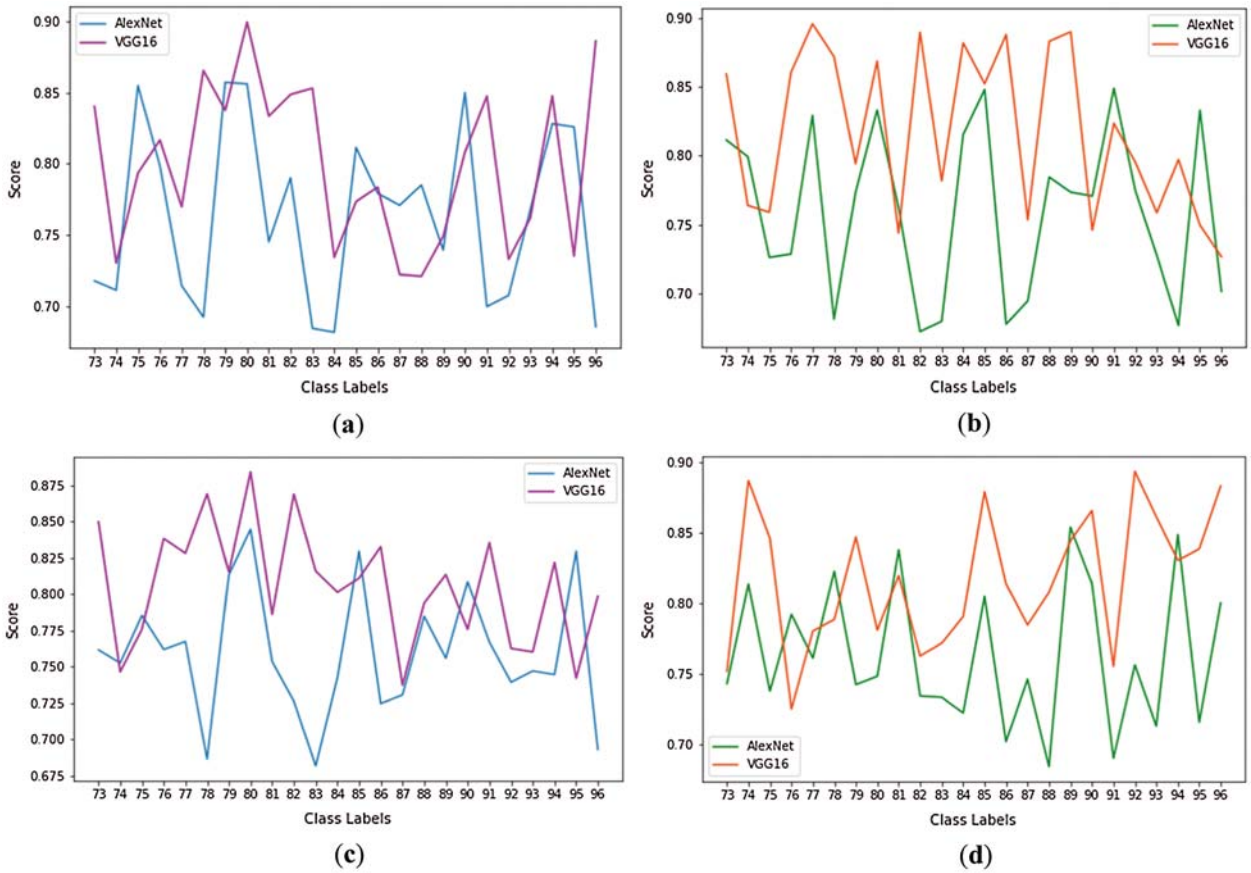

Figure 3: Visualization of the evaluation metric scores that are obtained by the AlexNet model and the VGG16 models. (a) The precision scores; (b) the recall (sensitivity) scores; (c) the F1 scores; (d) the specificity scores

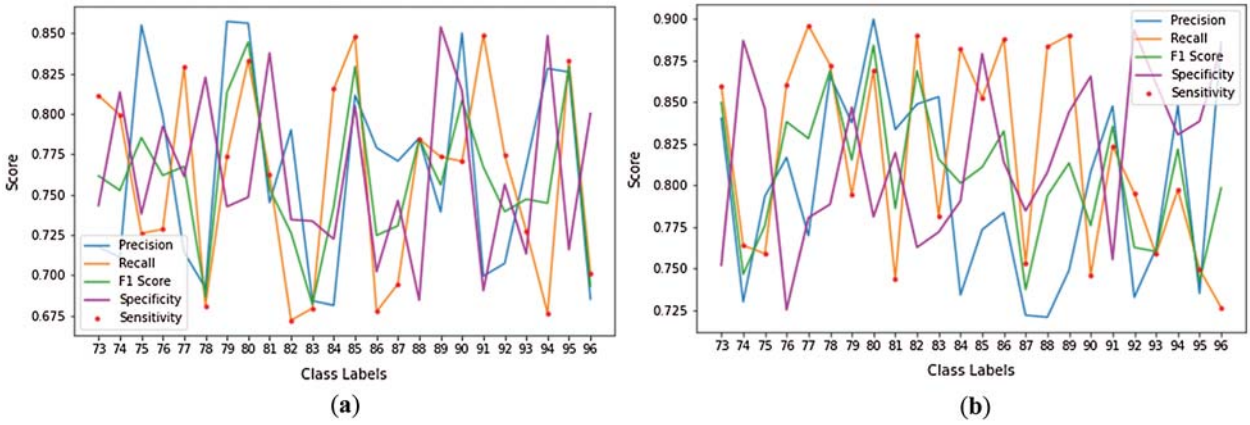

Figure 4: Visualizations of the aggregated evaluation metric scores obtained by the two CNN models: (a) the AlexNet model and (b) the VGG16 model

While this module is dedicated to facial landmark determination using the CNN models, the other module of this system uses image processing, along with computer vision and the HAAR classifier to determine the pupil and extract it from the live videos.

The HAAR classifier implemented in the provided XML files is combined with Python's OpenCV library to determine the face and eyes, as shown in Fig. 5. The HAAR classifier is able to detect the face and eyes with high accuracies. Using Python's OpenCV library, the face image is converted too gray-scale and the face and eyes are detected using the HAAR Classifier. Once the eyes are detected, the gray-scale image undergoes thresholding to enhance the pupils. The blobs, represented as red circles, are drawn around the detected pupils and displayed in the image.

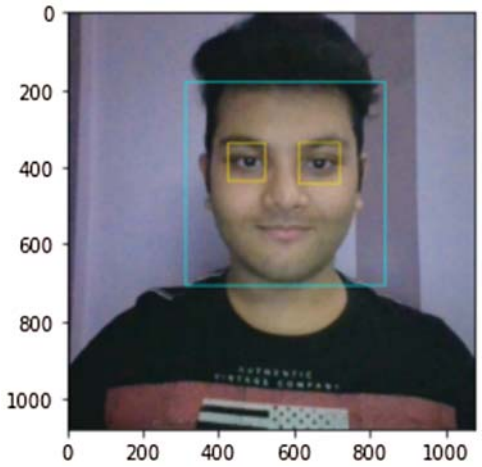

Figure 5: Face and eye detection using the HAAR classifier

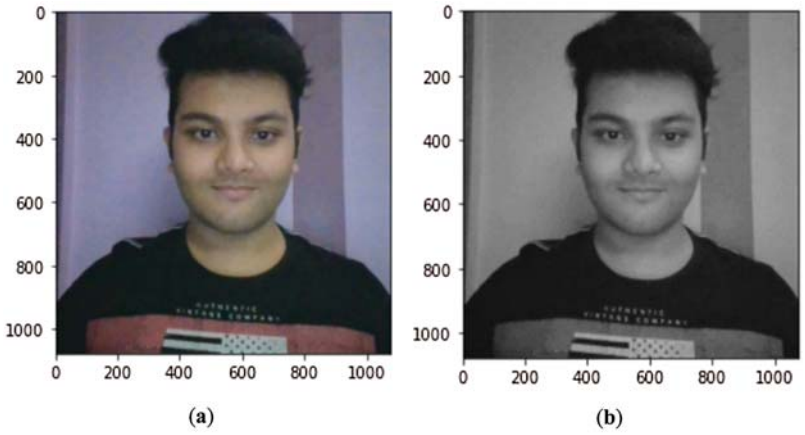

Figure 6: The image processing techniques are implemented for the images from a live video: (a) The colored image of a face and (b) the gray-scale image after image processing

The original, colored face image is shown in Fig. 6a and the resultant gray-scaled image is shown in Fig. 6b. Gray-scaling is applied to the colored image to obtain the image in Fig. 6b. The 4-tuple coordinate values obtained for both eyes are used to determine the left and right

eyes after eye detection. The sub-images of the eyes undergo gray-scaling to remove colors and improve the outcomes of thresholding.

Thresholding is required to extract the pupil. Different threshold values are tested to select the most suitable value. The gray-scaled image undergoes thresholding to obtain a segmented image for better feature extraction, as shown in Fig. 7. In Fig. 7, three different threshold values are used: 135, 90 and 45. It can be seen from Fig. 7 that the rightmost image with a threshold value of 45 has the best pupil extraction among the tested values. Once both eyes have been detected, one-fourth of the image is removed from the top to eliminate the eyebrows. The image is eroded to remove unwanted boundary pixels such as the eye corners and the image is dilated to restore the pixels lost in erosion and expand the pupils' features. The salt and pepper noise created by erosion and dilation are removed using a median filter, which also smoothens the image.

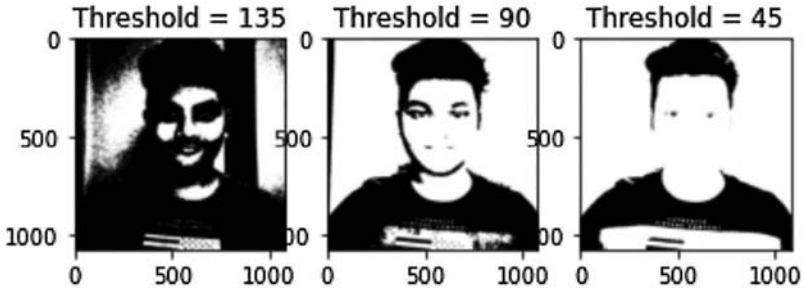

Figure 7: Images obtained using different threshold values

The left and right pupils extracted after thresholding are shown in Figs. 8a and 8c, respectively. The blobs are drawn around the extracted pupils' outline, as shown in Fig. 8b and 8d.

After the blobs have been determined using the CNN models by following the steps described in the previous sections, the Euclidean distances are calculated to perform gaze prediction Image processing algorithms for eye gaze prediction have been investigated in several works [24–30]. In this work, we compare the performances of the traditional eye gaze detection techniques using only image processing techniques with that of the proposed system using the VGG16 CNN model.

The evaluation metric scores obtained by the traditional eye gaze detection method, using only image processing techniques, are shown in Tab. 3. In addition to precision, recall, F1 score, specificity, and specificity, this work also compares the mutual information (MI). The MI is calculated according to Eq. (14), where the entropy function, H, is given by Eq. (13).

$$H(X) = -E[\log(f_x(X))] \tag{13}$$

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{14}$$

In Eq. (13), f_x represents the probability density function of the variable, X. $E[.]$ denotes the expected value function, which is negated to provide the entropy. In Eq. (14), $H(X, Y)$ is the total entropy of the joint variables (X, Y). The mutual information is denoted by $I(X; Y)$ for the variables X and Y. A lower MI value is preferred since it denotes higher independence of the variables.

The evaluation metric scores obtained by the proposed method are shown in Tab. 4. Comparing Tabs. 3 and 4, it can be seen that the scores obtained by the proposed method are higher than the scores for the traditional method.

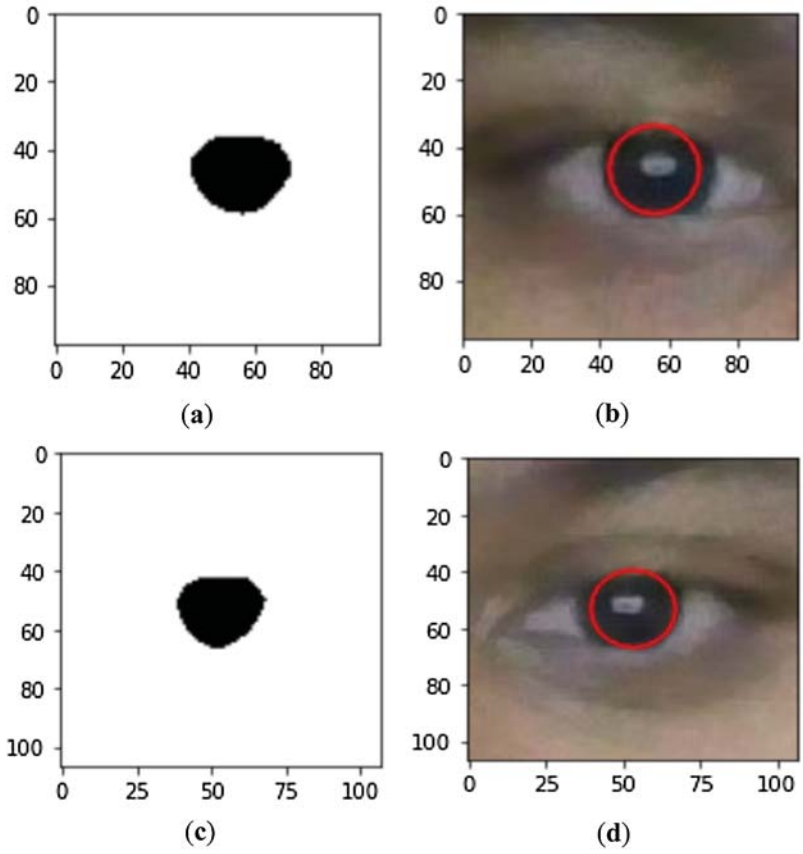

Figure 8: Pupil extraction and blob formation using HAAR classifier: (a) Left pupil extraction after thresholding, (b) blob formation on the left pupil, (c) right pupil extraction after thresholding and (d) blob formation on the right pupil

Table 3: Evaluation metric scores for eye gaze detection using the traditional method

Class label	Precision	Recall	F1 score	Specificity	Mutual information
Center	0.731442	0.704274	0.717601	0.735063	0.612488
Top, left	0.743069	0.709025	0.725648	0.716284	0.611334
Top, right	0.741575	0.703534	0.722054	0.710029	0.513245
Bottom, left	0.736568	0.721158	0.728782	0.694965	0.570149
Bottom, right	0.741648	0.693808	0.716931	0.731043	0.639762

Table 4: Evaluation metric scores for eye gaze detection using the proposed method

Class label	Precision	Recall	F1 score	Specificity	Mutual information
Center	0.823174	0.783230	0.802705	0.833660	0.506921
Top, left	0.822202	0.750120	0.784509	0.840002	0.505113
Top, right	0.842695	0.809641	0.825838	0.865054	0.477070
Bottom, left	0.812432	0.778174	0.794934	0.812065	0.478230
Bottom, right	0.837449	0.746624	0.789433	0.823848	0.481862

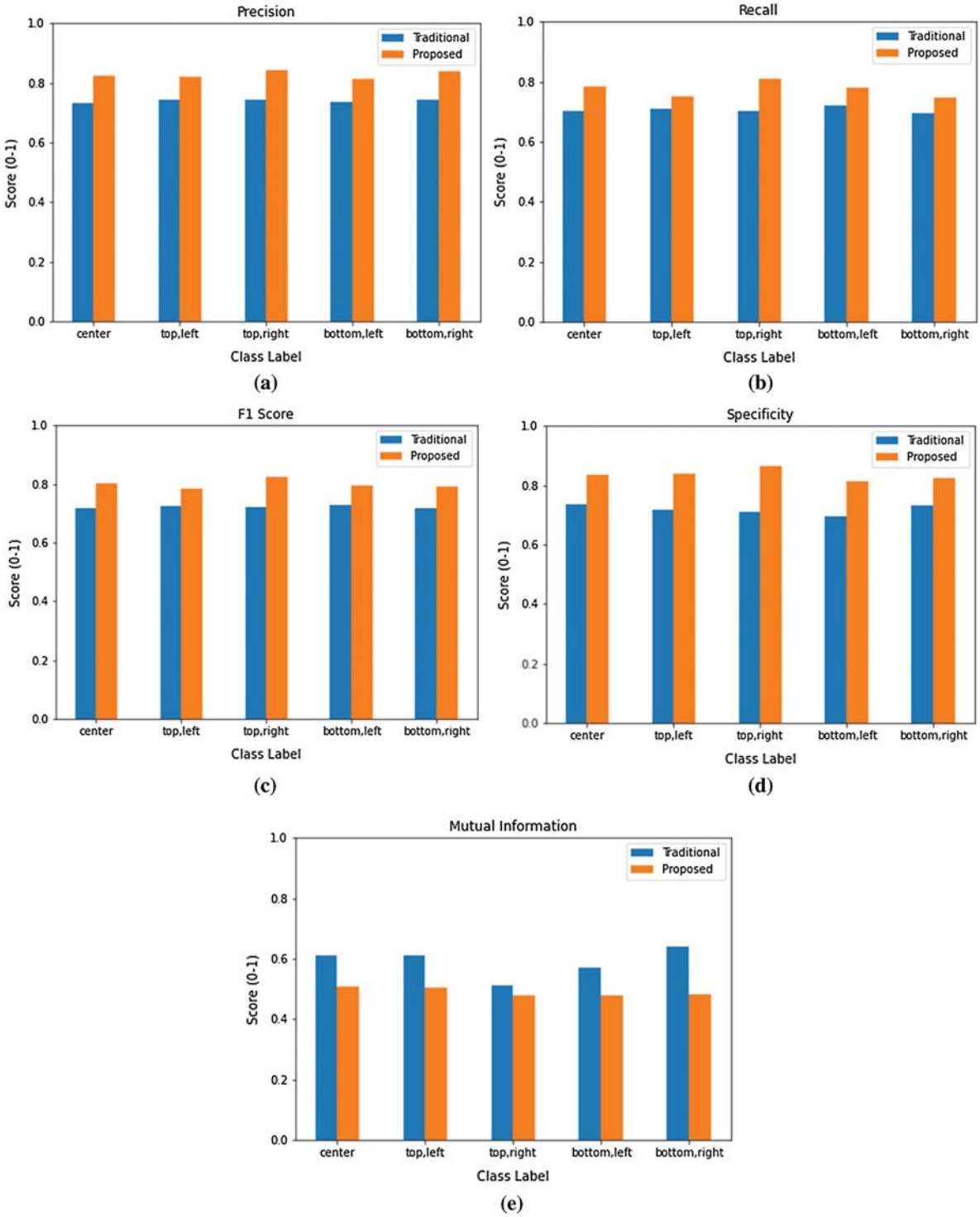

Figure 9: Visualization of the evaluation metrics for the systems implemented using the traditional and the proposed method: (a) the precision score, (b) the recall score, (c) the F1 score, (d) the specificity score, (e) the mutual information score

Figs. 9a–9e show the superior performances of the proposed system over the traditional system. The evaluation scores for precision, recall, F1 score, specificity, and sensitivity of the proposed system are all higher than those of the traditional system. The proposed system's mutual information is lower than that of the traditional system, inferring that the variables are more independent and is more desirable.

As demonstrated by the results, the proposed system outperforms the traditional approach for eye gaze detection. However, this system has not yet taken into account the effects of varied light sources, which may be a topic for investigation in the future. Additionally, the proposed system may also be extended to consider users with spectacles or various expressions involving the eyes, such as squinting or winking.

6 Conclusions

The pandemic caused by the COVID-19 in 2020 gave rise to a new, digitized normal, which drastically changed people's lives. In particular, the education and examination systems around the world were seriously affected. This work has proposed an eye gaze detector to monitoring the user's eye motions, which can be implemented in academic institutions as well as other locations where eye gaze tracking is required.

The system has been implemented by considering specific facial landmarks with the Convolutional Neural Network (CNN) models, in addition to the application of image processing and computer vision techniques. According to various evaluation metrics, namely the precision, recall, F1 score, specificity, sensitivity, and mutual information, the proposed system outperforms traditional eye gaze detectors, which only use image processing and computer vision methods. The gaze detection system is able to determine pupil positions classed as center, top-left, top-right, bottom-left, and bottom-right.

The proposed system facilitates education and conforms to the new digitized normal that might persist indefinitely after 2020. The work also discusses prospective improvements that can be made to increase the evaluation metric scores. In the future, the system can be further optimized towards better performances and more generalized applications, such as adapting to scenarios with various light sources, users with glasses or different facial expressions. The work can be further integrated with researches from other domains for more extended applications.

Funding Statement: This research was partially funded by the "Intelligent Recognition Industry Service Research Center" from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan. Grant Number: N/A and the APC was funded by the aforementioned Project.

Conflicts of Interest: The authors declare that they have no conflicts of interest regarding the present study.

References

[1] D. Datta, T. M. Navamani and R. Deshmukh, "Products and movie recommendation system for social networking sites," *International Journal of Scientific & Technology Research*, vol. 9, no. 10, pp. 262–270, 2020.

[2] Y. Qiu and Y. Wan, "Facial expression recognition based on landmarks," in *Proc. of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conf.*, Chengdu, China, IEEE, pp. 1356–1360, 2019.

[3] N. Dhingra, C. Hirt, M. Angst and A. Kunz, "Eye gaze tracking for detecting non-verbal communication in meeting environments," in *Proc. of the 15th Int. Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Switzerland, ETH Zurich, 2020.

[4] S. Park, K. Lee, J. A. Lim, H. Ko, T. Kim *et al.,* "Differences in facial expressions between spontaneous and posed smiles: Automated method by action units and three-dimensional facial landmarks," *Sensors*, vol. 20, pp. 1199, 2020.

[5] M. C. Su, Y. Z. Hsieh, Z. F. Yeh, S. F. Lee and S. S. Lin, "An eye-tracking system based on inner corner-pupil center vector and deep neural network," *Sensors*, vol. 20, pp. 25, 2020.

[6] G. Iannizzotto, A. Nucita, R. A. Fabio, T. Caprì and L. Lo Bello, "Remote eye-tracking for cognitive telerehabilitation and interactive school tasks in times of COVID-19," *Information—An International Interdisciplinary Journal*, vol. 11m, pp. 296, 2020.

[7] M. C. Sáiz Manzanares, J. J. Rodriguez Diez, R. Marticorena Sánchez, M. J. Zaparain Yanez and R. Cerezo Menendez, "Lifelong learning from sustainable education: An analysis with eye tracking and data mining techniques," *Sustainability*, vol. 12, pp. 1970, 2020.

[8] M. Tran, T. Sen, K. Haut, M. R. Ali and M. E. Hoque, "Are you really looking at me? A feature-extraction framework for estimating interpersonal eye gaze from conventional video," *IEEE Transactions on Affective Computing*, vol. 99, pp. 1, 2020.

[9] M. Dahmani, M. E. Chowdhury, A. Khandakar, T. Rahman, K. Al-Jayyousi *et al.,* "An intelligent and low-cost eye-tracking system for motorized wheelchair control," *Sensors*, vol. 20, no. 14, pp. 3936, 2020.

[10] T. Ujbanyi, G. Sziladi, J. Katona and A. Kovari, "Pilot application of eye-tracking to analyse a computer exam test," in *Proc. of the Cognitive Infocommunications, Theory and Applications*, Cham, Springer, pp. 329–347, 2019.

[11] J. Zhu and J. Yang, "Subpixel eye gaze tracking," in *Proc. of the Fifth IEEE Int. Conf. on Automatic Face Gesture Recognition*, Washington, DC, USA, IEEE, pp. 131–136, 2002.

[12] H. R. Chennamma and X. Yuan, "A survey on eye-gaze tracking techniques," arXiv: 1312.6410, 2013.

[13] D. Datta, D. Mittal, N. P. Mathew and J. Sairabanu, "Comparison of performance of parallel computation of CPU cores on CNN model," in *Proc. of the 2020 Int. Conf. on Emerging Trends in Information Technology and Engineering*, Vellore, India, IEEE, pp. 1–8, 2020.

[14] D. Datta and S. B. Jamalmohammed, "Image classification using CNN with multi-core and many-core architecture," In: *Applications of Artificial Intelligence for Smart Technology*, Pennsylvania, United States: IGI Global, pp. 233–266, 2021.

[15] E. Fernández, L. Fajarí, G. Rodríguez, M. Cocera, V. Moner *et al.,* "Reducing the harmful effects of infrared radiation on the skin using Bicosomes incorporating β-carotene," *Skin Pharmacology and Physiology*, vol. 29, no. 4, pp. 169–177, 2016.

[16] A. Bozkurt and B. Onaral, "Safety assessment of near infrared light emitting diodes for diffuse optical measurements," *Biomedical Engineering Online*, vol. 3, no. 1, pp. 1–10, 2004.

[17] K. N. Kim and R. S. Ramakrishna, "Vision-based eye-gaze tracking for human computer interface," *Proc. of the IEEE SMC'99 Conf. Proc. 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics*, vol. 2, pp. 324–329, 1999.

[18] R. Santos, N. Santos, P. M. Jorge and A. Abrantes, "Eye gaze as a human-computer interface," *Procedia Technology*, vol. 17, no. 6, pp. 376–383, 2014.

[19] J. Sanchez-Riera, K. Srinivasan, K. Hua, W. Cheng, M. A. Hossain *et al.,* "Robust RGB-D hand tracking using deep learning priors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2289–2301, 2018.

[20] K. Srinivasan, A. Ankur and A. Sharma, "Super-resolution of magnetic resonance images using deep convolutional neural networks," in *2017 IEEE Int. Conf. on Consumer Electronics Taiwan*, Taipei, pp. 41–42, 2017.

[21] I. J. L. Paul, S. Sasirekha, S. U. Maheswari, K. A. M. Ajith, S. M. Arjun *et al.,* "Eye gaze tracking-based adaptive e-learning for enhancing teaching and learning in virtual classrooms," in *Proc. of the Information and Communication Technology for Competitive Strategies*, Singapore, Springer, pp. 165–176, 2019.

[22] C. H. Morimoto and M. R. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[23] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in *Proc. of the European Conf. on Information Retrieval*, Berlin, Heidelberg, Springer, pp. 345–359, 2005.

[24] V. Sean, F. Cibrian, J. Johnson, H. Pass and L. Boyd, "Toward digital image processing and eye tracking to promote visual attention for people with autism," in *Adjunct Proc. of the 2019 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing and Proc. of the 2019 ACM Int. Symp. on Wearable Computers*, New York, New York, United States. ACM, pp. 194–197, 2019.

[25] D. Datta and J. Dheeba, "Exploration of various attacks and security measures related to the internet of things," *International Journal of Recent Technology and Engineering*, vol. 9, no. 2, pp. 175–184, 2020.

[26] T. Ngo and B. S. Manjunath, "Saccade gaze prediction using a recurrent neural network," in *Proc. of the 2017 IEEE Int. Conf. on Image Processing*, Beijing, China, IEEE, pp. 3435–3439, 2017.

[27] D. Datta, R. Agarwal and P. E. David, "Performance enhancement of customer segmentation using a distributed python framework, ray," *International Journal of Scientific & Technology Research*, vol. 9, no. 11, pp. 130–139, 2020.

[28] D. Datta, P. E. David, D. Mittal and A. Jain, "Neural machine translation using recurrent neural network," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 4, pp. 1395–1400, 2020.

[29] W. Fuhl, D. Geisler, T. Santini, W. Rosenstiel and E. Kasneci, "Evaluation of state-of-the-art pupil detection algorithms on remote eye images," in *Proc. of the 2016 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing: Adjunct*, New York, New York, United States, ACM, pp. 1716–1725, 2016.

[30] H. Mohsin and S. H. Abdullah, "Pupil detection algorithm based on feature extraction for eye gaze," in *Proc. of the 2017 6th Int. Conf. on Information and Communication Technology and Accessibility*, Muscat, Oman, IEEE, pp. 1–4, 2017.