Grenze International Journal of Engineering and Technology, January Issue

Online Proctoring using Eye Gaze Tracking and Head Pose Estimation

Dr. Swathika R[1], Radha N[2] and H Varsha[3]
[1-3]Dept. of Information Technology Sri Sivasubramaniya Nadar College of Engineering Kalavakkam, Chennai, India
Email: swathikar@ssn.edu.in, radhan@ssn.edu.in, varsha2011056@ssn.edu.in

***Abstract*—With the emergence of COVID-19, education systems have switched over to e-learning. Online examinations are currently held by utilizing webcams and supervisors, but limitations arise with the growing number of students. Our system detects the examinee's gaze direction and estimates the examinee's head position in real-time for online proctoring. The eye region obtained using facial landmarks is used in the subsequent stages to classify the eye gaze direction. The eye gaze space is reduced to three classes: left, right and center. A CNN is employed in this work for the classification of eye gaze direction and achieved an accuracy of 97%. The head pose is estimated by projecting the camera coordinates onto the image plane. The head position along with the eye gaze direction is used to identify unfair practices of the examinee. The application saves the person's data when his VFOA (visual focus of attention) is off the screen for the examiner to manually check and mark whether the examinee's action was attempted malpractice or just a momentary lapse in concentration. Thus, our application successfully identifies suspicious activity and alerts the examinee.**

***Index Terms*— Eye gaze tracking, Head pose estimation, Online proctoring, CNN, Gaze direction.**

I. INTRODUCTION

The COVID-19 pandemic has caused disruption in many spheres of our lives. Physical interaction for education, work, and leisure has been regulated by governments around the world to minimize human infection rates. In response to the COVID-19 crisis, most governments around the world are closing educational institutions and moving their activity to online and remote modalities impacting over 89% of the world's student population.

Due to its adaptability, accessibility, and user-friendliness, e-learning has become more and more popular. Exams are an essential part of every educational program, including online education programs. The danger of cheating exists in every exam, thus its identification and prevention are crucial. For educational qualifications to remain valuable to society, they must demonstrate actual learning. When it comes to online exams, the research community has significant challenges with the use of proctoring processes. One popular method for proctoring online tests is online human monitoring.

The primary drawback is that it is quite expensive because numerous workers are needed to oversee the students. Methods and technology that offer reliable tools to identify unfair, immoral, and unlawful behaviour during exams must be developed. The current body of knowledge in this area is minimal, with the majority of the software coming from for- profit companies that offer constrained and "non-open" software tools.

This project intends to develop an online proctoring system that continuously monitors physical locations without

Grenze ID: 01.GIJET.11.1.429

the need for a human proctor by utilising computer vision and deep learning techniques.
The system employs biometric approaches including face recognition using the HOG face detector and the OpenCV face recognition algorithm. The user's face is captured from the camera and the images of the eyes are extracted. The eye images are stored in a dataset that is used to train two CNN models, one for each eye, that classifies the eye gaze direction of the user. The real-time images of the face are used to determine the head position of the examinee with the help of HOG face recognition and OpenCV tools. Both eye gaze direction and the estimated head pose are used together to detect unfair behaviour during the exam.
The primary goal of this project is to deliver an application that detects unfair behaviour in an online examination. The application can be used extensively in educational institutions as well as in corporate sector wherever exams have been conducted.

1. The application can isolate a human face in a video feed.
2. The application can determine where the examinee's eyes are directed.
3. The application can determine where the examinee's head is with relation to the X, Y, and Z axes.
4. The application can operate in real time.

The following events that could happen during the test are detected by the application:

1. The presence of a second person,
2. Malpractice based on eye movement and head pose, and
3. The absence of the examinee.

II. Literature Review

The literature review is split into two main categories. In the first category, the literature related to eye gaze direction concepts is discussed. In the second category, the literature related to head pose estimation is discussed briefly here.

A. Eye Gaze Direction

To help academics improve the design of more effective and efficient deep learning-based eye-tracking models, Akinyelu, A.A. et al. [1] undertook an assessment of contemporary CNN-based gaze estimation techniques. Additionally, they discussed how different gaze estimation models performed as well as helpful information on a variety of large-scale datasets, network architectures like AlexNet, ResNet, SegNet, SqueezeNet, and Own architecture, among others, as well as pre-trained models that can be used to create better deep learning-based gaze estimation models.
Kanade, P. et al. [2] proposed a CNN-based approach for driver gaze classification in the vehicular environment. Face, left eye, and right eye images were extracted from the input image based on the ROI (Region-of-Interest) identified by facial landmarks from the facial attribute tracker for driver gaze classification. They fine-tuned the extracted cropped images of the face, left eye, and right eye with a pre-trained CNN model using the theVGG-face network to obtain the appropriate gaze features.
Datta, D. et al. [3] proposed an eye-gaze detection system by considering facial landmarks with two different Convolutional Neural Network (CNN) models, namely the AlexNet model and the VGG16 model. In addition to the application of image processing and computer vision techniques. Other evaluation metrics, namely the precision, recall, F1 score, specificity, sensitivity, and mutual information were used. The proposed system outperformed traditional eye gaze detectors, which only use image processing and computer vision methods. The gaze detection system was able to determine pupil positions classed as center, top-left, top-right, bottom-left, and bottom-right.
A five-layered CNN-based model for driver gaze zone description and head-pose estimation was developed by Choi,
I.H. et al. [4]. Based on the dataset, they created a CNN model that used a multi-GPU platform to identify nine driver gaze zones and approximation of head pose. To operate in real- time, the network parameters were sent to a GPU inside a Windows-based PC. Their approach surpassed the most advanced gaze zone classification methods based on traditional computer vision techniques, with an average accuracy rate of categorization reaching 95%.
Aunsri, N. [5] proposed a new set of features for natural head-pose eye gaze estimation for a low-cost eye gazing system, which could be used by and accessible to most people. They have tested the efficiency of the features with several machine learning methods, such as k-NN, naive Bayes with kernel density estimation, Random Forest, ANN, and SVM. The gaze position was estimated in 15 regions of a computer screen using ANN with 2 hidden layers and a recognition rate of 93.94% was achieved.

B. Head Pose Estimation

Ruiz, N. et al. [6] compared pose estimated from landmarks using two different landmark detectors, FAN and Dlib, for a landmark-free head pose estimation. The model was trained on the synthetic 300W-LP (300 Faces-In-The- Wild across Large Poses) dataset and was tested on two datasets - AFLW2000 and BIWI. They combined ResNet50 with a multi-loss architecture. Each loss contains a binned pose classification and regression, corresponding to yaw, pitch, and roll individually. With binned classification, their method obtained a robust neighborhood prediction of the pose.

Lee, S. et al. [7] addressed the problem of head pose estimation with two degrees of freedom (pitch and yaw) using a low- resolution image.

Yu, Y. et al. [8] presented a method for robust head pose estimation by augmenting a 3DMM face model with a set of 3D points extracted from a reconstruction of the person's face using a KinectFusion methodology. They reconstructed online head 3D using a KinectFusion methodology, and the result was combined with the 3DMM face model. The experiment results showed that their framework achieved not only accurate but also robust performance even when extreme head poses were presented and extended to estimate facial expressions by directly incorporating expression blend shapes.

Patacchiola, M. et al. [9] introduced the use of dropout and adaptive gradient methods for head pose estimation with CNNs. They used the Viola-Jones object detection framework as a face detector and two CNNs of type B trained on the AFLW (Annotated Facial Landmarks in the Wild) dataset as head pose estimators. They investigated the use of dropout and adaptive gradient methods giving a contribution to their ongoing validation. Their results showed that joining CNNs and adaptive gradient methods led to the state-of-the-art unconstrained head pose estimation.

Shao X. et al. [10] compared the different methods to estimate the head pose of a subject. These methods were based on convolutional neural networks, random forests, manifold learning, depth cameras, and others.

From a modelling perspective, our approach is similar to that proposed by Datta, D. et al. [3]. The models perform a 3- class classification where the classes are – left, right and center. Indi, C. S. et al. [11] classified malpractices performed by an examinee by using two classifiers where one classifier was used when the examinee's eyes were visible and one where the eyes were not visible. When the eyes were not visible, only the head position was considered but, in our approach, the final classification result is obtained by combining the gaze direction and head pose.

III. System Design and Implementation

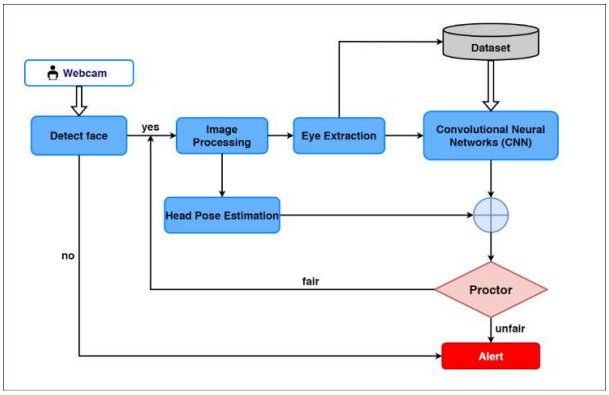


Fig. 1. System Architecture

This section describes the components of the system and their implementation in detail. Fig. 1. represents the block diagram of the system.

A. Input Video Acquisition

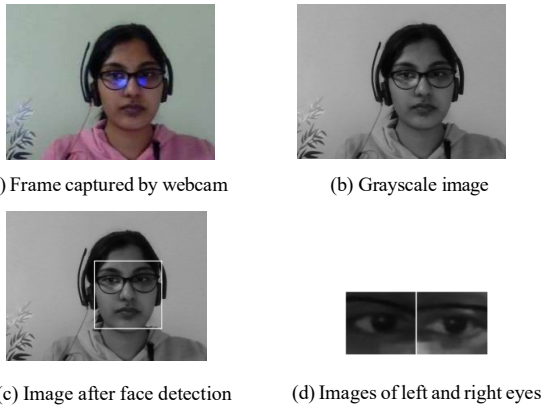
The data was collected from the in-built camera present in the laptop. Using an in-built camera allowed for the acquisition of images under real-world conditions and not typical laboratory conditions.

Test subjects were asked to pay attention to the lighting during image acquisition. The live feed from webcam is captured using OpenCV library. A video capture object is created for the camera. An infinite loop is created to read the image using the created object.

B. Face Detection

Face detection is based on the HOG (Histogram of Oriented Gradients) feature descriptor with a linear SVM machine learning algorithm as studied by Dalal N. et al. [12].

In the initial stage of face detection, we convert our input image into grayscale because we don't need an RGB image to find faces. In our proposed system, we used the HOG (Histograms of Oriented Gradients) method to detect the faces. We used the HOG frontal face detector method using Dlib and the OpenCV library for face detection. This can be seen in Fig. 2.(a, b, c).



(a) Frame captured by webcam (b) Grayscale image

(c) Image after face detection (d) Images of left and right eyes

Fig. 2. Steps involved in Face Detection and Eye Extraction

C. Eye Extraction

Facial landmarks are used to localize and represent salient regions of the face. The pre-trained facial landmark detector inside the Dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face. This was studied by King, D. E. et al. [13].

The face is detected first, and then the eyes are detected in the face image. This is done using the 68 facial landmarks on the face. The landmarks are obtained using the shape predictor method present in the Dlib library. The Region of Interest (ROI) can be accessed by python indexing:

- The right eye can be accessed through points [36, 42].
- The left eye can be accessed through points [42, 48].

The last step before using the data to train the CNN is to resize the images. For proper operation, the neural network requires that each of the input images should be of the same size. Therefore, all eye images were rescaled to 56 × 64 pixels using methods available in the OpenCV library as seen in Fig. 2.(d).

D. Dataset

The subject sat centrally facing the screen with the built- in camera at the height of the horizontal axis of the eyes at a distance of about 35 to 50 cm from the monitor. During every session, the subject was asked to look at different directions and one image was captured every two seconds. Two datasets were created for each of the eyes – left eye, right eye. The captured images of left eye and right eye respectively were stored into three different classes – Left, Right, Center. The datasets consisted of images acquired from three participants. Each dataset was split into training and testing data. There are approximately 3500 images in left eye dataset and 3500 images in the right eye dataset. Images from the datasets can be seen in Fig. 3. and Fig. 4.

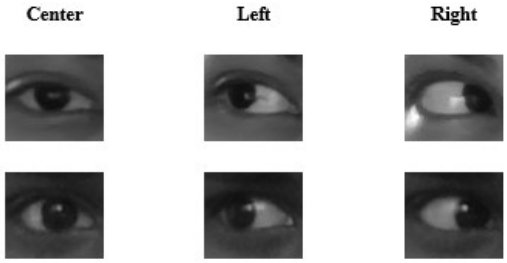


Fig. 3. Examples of images from the left eye dataset

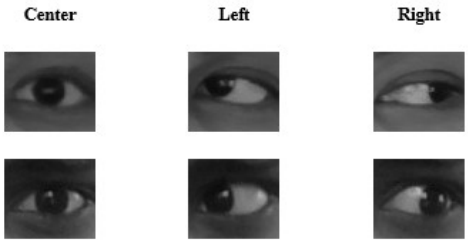


Fig. 4. Examples of images from the right eye dataset

E. Convolutional Neural Networks (CNNs)

The convolutional neural network represents a type of feed-forward neural network which can be used for a variety of machine learning tasks. Even though the training time is huge, the accuracy and robustness of CNNs are better than most of the standard machine learning algorithms. Every network consists of three kinds of components:

Convolutional layer (Convolution): When an input is forwarded into a network, each filter is convolved across the width and height of the input volume. The dot operation is computed between the entries of the filter and the entries from the input using (1). It produces an image with the layers equal to the number of filters.

$$[i,j] = (m * k)[i,j] = \sum_a \sum_b k[a,b]m[i-a, j-b] \quad (1)$$

where m represents the input image and k represents the kernel, a and b represent the row and column of the resultant matrix.

- **Pooling Layer (Subsampling):** The layer's input is an image, and the output is the image reduced in both dimensions. Only Max Pooling layers were used, which reduced the image by representing the area of a given size by one pixel, which is the brightest. There is only one parameter for this layer—the size of the reduction.
- **Fully connected layer (FC):** It is a classic neural network's layer that consists of some number of neurons, and every neuron received a weighted combination of all input values. The fully connected layer performs linear transformation and non-linear transformation on the 1D array. The linear transformation is performed using (2), while the non- linear transformation is performed using (3).

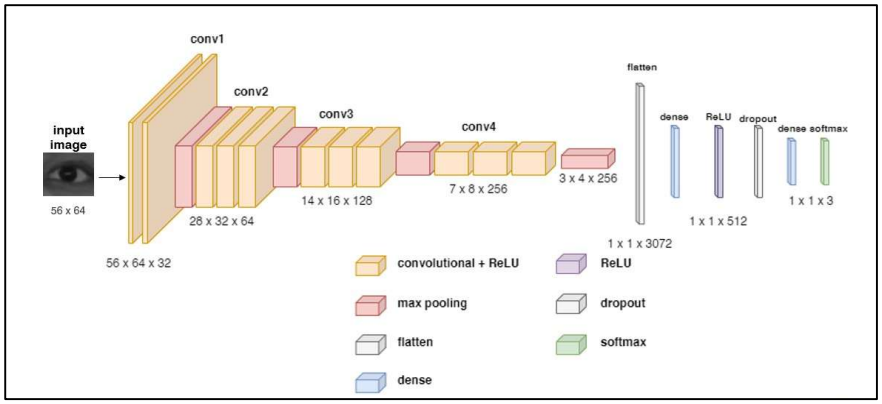


Fig. 5. CNN Architecture

$$Z = W^T . X + b \quad (2)$$

where X refers to the input, b refers to the bias, and
W refers to the learnable parameters (or weights) in the network.

$$output = (Z) \quad (3)$$

where σ refers to the activation function (such as ReLU), and Z refers to the output of the linear transformation operation.

Two CNN models are used to classify each of the eyes – one for the left eye and one for the right eye. The architecture of each of these CNN models remains the same. The images of eyes obtained from the previous stage is used in a multiclass classification framework for predicting the gaze direction using CNN.
The photos in the input stage have dimensions of 56 ×
64. There are thirty-two filters with a 3 × 3 dimension in the first convolutional layer. A rectifier linear unit (ReLU) was used to add non-linearity to the activations after this step. Following the ReLU phase, a max pooling layer is added, which subsamples each output image spatially. Our choice of 2 × 2 max-pooling layers results in a halving of the spatial resolution. There were additionally added three comparable stages with 64, 128 and 268 filters, respectively. A fully connected layer combines the outputs from every convolutional layer. The number of classes in the dataset (i.e., the three directions: left, right, and center) is equal to the number of output nodes. The structure of the network is shown in Fig. 5. The softmax loss was used over classes as the error measure. Cross entropy loss was minimized in the training.

F. Head Pose Estimation

Pose estimation is often referred to as a Perspective-n- Point problem or PnP problem in computer vision. To calculate the 3D pose of the head in an image, the following information is required:

- **2D coordinates:** 2D coordinate points certain points of the face are obtained from the 68 facial landmark points.
- **World Coordinates:** These are the 3D locations of the corresponding 2D feature points.
- **Intrinsic parameters of the camera:** This includes focal length of the camera, the optical center in the image and the radial distortion parameters.

The world coordinates along with the 2D coordinates are considered to obtain the translational and rotational vectors. These vectors along with the intrinsic parameters of the camera are used to project the 3D points on a 2D surface that is our image.
The head pose is defined by three Euler angles around three axes. These angles are pitch (nodding), yaw (shaking) and roll (tilting) as illustrated in Fig. 6. Pitch angle (□) is the head rotation around the horizontal x-axis. Yaw angle (□) measures the rotation around vertical y-axis. Roll angle (□) is the rotation around the z-axis perpendicular to frontal face plane. In our approach, we have used the Yaw angle (□) to find the position of the head.

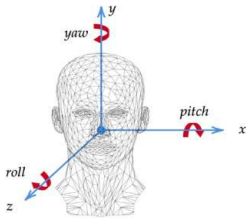


Fig. 6. Euler Angles

The world coordinates of the region-of-interest (ROI) were also taken and the rotational and translational vectors were estimated. Using these values, the 3D camera points were projected on a 2D image plane.
The yaw angle was calculated and if it was found to be greater than a specific angle, it meant the user's head was facing the left side. If the yaw angle was found to be lesser than a specific angle, it meant the user's head was facing the left side.
For ease of predicting whether the user is cheating or not, we used two variables namely head_pose1 and head_pose2 which had different ranges. The ranges were assigned as shown in Table I.

TABLE I. HEAD POSE RANGE

Variable	Left	Right
head_pose1	≥ 36	≤ -36
head_pose2	≥ 15	≤ -15

G. Proctor

The proctor evaluates the user for fair and unfair practices. The proctor considers both the head position and the gaze direction of the eyes to evaluate the user. The gaze of the left and right eye was obtained from the each of the CNN models and stored in two variables namely gaze_left and gaze_right respectively.

The two variables used in head pose estimation were also taken into consideration. The algorithm used by the proctor for evaluation is detailed in Algorithm 1.

ALGORITHM 1: Master Algorithm

```
if head_pose1 == Center then
    if gaze_left == left and gaze_right == left then
        if head_pose2 == Center then
            alert user
        elif head_pose2 == Left then
            alert user
        end
    elif gaze_left == right and gaze_right == right then
        if head_pose2 == Center then
            alert user
        elif head_pose2 == Right then
            alert user
        end
    end
elif head_pose1 == Left then
    alert user
elif head_pose1 == Right then
    alert user
end
```

IV. Experimental Results

A. Performance of Model

The results obtained from the experiments are discussed here. CNNs were trained separately for left and right eyes using data augmentation. Regularization was included while creating the model as to avoid over-fitting effects.

We used the confusion matrix, which is a technique for summarizing the performance of a classification algorithm. There are some key terms for evaluating the confusion matrix, including True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), accuracy, precision, recall, etc.

Classification Report

	precision	recall	f1-score	support
center	0.94	0.98	0.96	243
left	0.99	0.98	0.99	287
right	0.99	0.97	0.98	310
accuracy			0.97	840
macro avg	0.97	0.98	0.97	840
weighted avg	0.98	0.97	0.98	840

Fig. 7. Classification Report – Left Eye Model

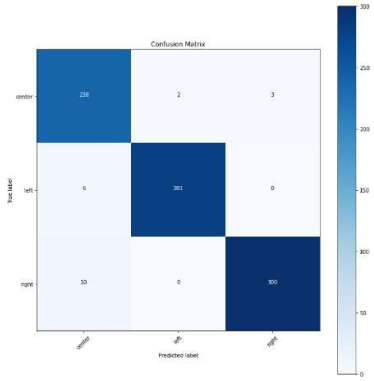


Fig. 8. Confusion Matrix – Left Eye Model

Regarding the evaluation with the original dataset, Fig. 7. shows the F1 Score metric, Precision, Recall and the overall accuracy of the model on left eye. For the CNN on Left Eye, the left direction has the highest average F1

Score equal to 0.99, followed by the right direction, with 0.98. The center direction resulted in 0.96 average F1 Score.
An average accuracy of 97.5% was achieved for the CNN model on left eye. The confusion matrix of the model is shown in Fig. 8.

Classification Report

	precision	recall	f1-score	support
center	1.00	0.93	0.97	259
left	1.00	1.00	1.00	299
right	0.95	1.00	0.97	301
accuracy			0.98	859
macro avg	0.98	0.98	0.98	859
weighted avg	0.98	0.98	0.98	859

Fig. 9. Classification Report – Right Eye Model

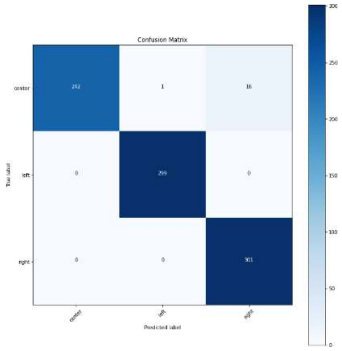


Fig. 10. Confusion Matrix – Right Eye Model

Fig. 9. shows the F1 Score metric, Precision, Recall and the overall accuracy of the model on right eye. An average accuracy of 98% was achieved for the CNN model on right eye. The confusion matrix of the model is shown in Fig. 10.

B. Performance of the System

When the examinee attempts to commit malpractice while taking an assessment, a warning is displayed in the window. Fig. 11. visualizes some of the outputs of the system. Examinees are advised with ten warnings. If any anomalous behaviour or malpractices are detected after ten warnings during the assessment, the examinee will be terminated. All suspected behaviour is recorded in a CSV file and will be reported to the exam coordinator. The frames where the examinee is involved in malpractices are also captured.

V. Conclusion and Future Work

We have proposed a model of a proctoring system to prevent harmful behaviour during online exams. This method of eye-gaze estimation brings together our work for head and eye pose estimation, using a low-resolution webcam, without calibration. Our method has been evaluated and its performance was found to be comparable to the state- of-the-art methods. The functionalities of the system include:

- **Eye Gaze Tracking:** The proposed method detects the pupil center and classifies user's gaze direction. Identifying where the examinee is looking throughout an exam is one of the most important features in determining whether or not he is cheating. The left and right eye models achieved an average accuracy of 97.5% and 98% respectively in classifying the eye gaze direction and were found to be comparable to the state-of-the-art systems.
- **Head Pose Estimation:** The movement of the examinee's head can be an indication to engagement in cheating activities.

Future work aims to improve the robustness of the salient feature tracking especially under large head rotations, in order to improve upon the achieved gaze estimation accuracy. With the addition of audio checking, it would be possible to isolate voices sources and determine whether the examinee is cheating using auditory aids. Object detection can also be incorporated to identify if the examinee is using a mobile phone or a book.

Fig. 11. Visualizing the output of the system – Looking at screen, Looking away, No face detected, Multiple faces detected

Another metric that would provide more insight would be the distance of the subject from the screen. It can be precisely measured if the testing condition has more than one lens/camera, enabling depth estimation calculation. Adding these metrics would result in a more accurate model.

This system can be used in conjunction with a secure exam browser to prevent cheating in online exams. However, because the system will not be completely successful in eliminating all forms of cheating, human intervention may be required in such cases.

REFERENCES

[1] Akinyelu, A.A. and Blignaut, P., 2020. Convolutional neural network- based methods for eye gaze estimation: A survey. IEEE Access, 8, pp.142581-142605.

[2] Kanade, P., David, F. and Kanade, S. (2021) "Convolutional Neural Networks (CNN) based Eye-Gaze Tracking System using Machine Learning Algorithm", *European Journal of Electrical Engineering and Computer Science*, 5(2), pp. 36–40. doi: 10.24018/ejece.2021.5.2.314.

[3] Datta, D., Maurya, P.K., Srinivasan, K., Chang, C.Y., Agarwal, R., Tuteja, I. and Vedula, V.B., 2021. Eye Gaze Detection Based on Computational Visual Perception and Facial Landmarks. *CMC- COMPUTERS MATERIALS & CONTINUA*, *68*(2), pp.2545-2561.

[4] Choi, I.H., Hong, S.K. and Kim, Y.G., 2016, January. Real-time categorization of driver's gaze zone using the deep learning techniques. In *2016 International conference on big data and smart computing (BigComp)* (pp. 143-148). IEEE.

[5] Aunsri, N. and Rattarom, S., 2022. Novel eye-based features for head pose-free gaze estimation with web camera: New model and low-cost device. *Ain Shams Engineering Journal*, *13*(5), p.101731.

[6] Ruiz, N., Chong, E. and Rehg, J.M., 2018. Fine-grained head pose estimation without keypoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 2074- 2083).
[7] Lee, S., Saitoh, T. (2018). Head Pose Estimation Using Convolutional Neural Network. In: Kim, K., Kim, H., Baek, N. (eds) IT Convergence and Security 2017. Lecture Notes in Electrical Engineering, vol 449. Springer, Singapore.
[8] Yu, Y., Mora, K.A.F. and Odobez, J.M., 2017, May. Robust and accurate 3D head pose estimation through 3DMM and online head model reconstruction. In *2017 12th ieee international conference on automatic face & gesture recognition (fg 2017)* (pp. 711-718). Ieee.
[9] Patacchiola, M. and Cangelosi, A., 2017. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, *71*, pp.132-143.
[10] Shao, X., Qiang, Z., Lin, H., Dong, Y. and Wang, X., 2020, November. A survey of head pose estimation methods. In 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics) (pp. 787-796). IEEE.
[11] Indi, C.S., Pritham, K.C.S., Acharya, V. and Prakasha, K., 2021. Detection of Malpractice in E-exams by Head Pose and Gaze Estimation. *International Journal of Emerging Technologies in Learning*, *16*(8).
[12] Dalal, N. and Triggs, B., 2005, June. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886- 893). IEEE.
[13] King, D.E., 2009. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, *10*, pp.1755-1758.