An Intelligent Examination Monitoring System Based on Deep Learning

Sarla More, Puja Gupta, Deepesh Agrawal,Chandra Prakash Singar

Assistant Professor , School of CSIT, Symbiosis University of Applied Sciences Indore

Department of Information Technology, SGSITS, Indore, 452001, India

Abstract: As Artificial Intelligence (AI) and education have merged to provide people with the opportunity to acquire new skills, the smart education system has exploded over the past decade. As the demand for smart proctoring services increases, AI-assisted proctoring solutions are in high demand. By developing a multimodal system, we eliminate the need for a human examiner to be available during the examination. Our system utilized a High definition (HD) camera and live window capture to obtain images. First, all humans in the image are identified, and then the head's feature points are calculated to determine its distance from other human heads. Facial feature analysis is performed to infer the candidate's expression. The immediate environment of the examinee can be taken up, including a cell phone, a piece of paper, or the mere existence of another person. Furthermore, our system monitors the examinee's mouth opening and facial deception. The combination of such models produces a smart rule-based prediction system that can figure out the probability that there was examination deception. We conducted a thorough evaluation of our system and determined that it performed well, with a precision of 0.95.

Keywords: Facial features, Head pose, Eye gaze, Mouth aspect ratio, Eye Net.

1 Introduction

The integrity of the testing process, the fairness of the tests, and the efficacy with which they are carried out are all directly impacted by the presence of exam observers. Exam observers are responsible for observing applicants while they are taking examinations, providing a secure atmosphere, and respecting the norms and regulations that have been established by the body that is in charge of administering the exams. Other names for exam observers include invigilators and proctors. Their presence is necessary in order to forestall dishonesty and unethical behavior, which is crucial for the preservation of the evaluation procedure's credibility and validity.

The following are some of the primary roles of an exam observer:

- Enforcement of the Rules: The exam observer is responsible for enforcing the regulations and ensuring that all applicants obey the guidelines and rules that have been given for the test. This involves checking the identity of the candidates, ensuring that they have the appropriate materials, and rigidly enforcing regulations governing communication and behavior while the test is being administered.

- Security Measures: It is their responsibility to ensure that adequate security measures are in place during the examination to forestall the use of any unauthorized items or equipment. In this regard, it may be necessary to keep an eye on the applicants to detect any instances of plagiarism or cheating.

- Management of Exam Materials: The exam observer is responsible for distributing test papers or credentials to login for electronic examinations, collecting complete answer sheets or the submissions, and ensuring that any supplies are properly accounted for and managed in a secure manner.

- Assistance for Candidates: Exam observers are required to maintain an objective and impartial posture throughout the test; nonetheless, they are ready to assist candidates with procedural issues and give explanations about the examination instructions.

- Timekeeping: It refers to keeping track of the length of the test and notifying the remaining time at regular intervals. This gives applicants the opportunity to efficiently manage their time within the time constraints of the exam.

- Handling Emergencies: The observer is responsible for taking fast action in the event that the test is interrupted by any interruptions or technological difficulties in order to remedy the situation while minimizing any effect on the applicants.

- Reporting Incidents: If any violations or irregularities are seen during the test, the observer is responsible for documenting them and reporting them to the authority in charge of the exam for further inquiry.

- Establishing a Peaceable Atmosphere: The examination room should be kept calm and accommodating in order to guarantee that applicants are able to focus on performing to the best of their ability.

Exam observers provide a critical function in the test process, helping to ensure that it is both credible and conducted fairly. Students, exam panels, and other stakeholders gain trust in the process as a result of their participation, since it ensures that all students are examined in standardized and equal settings. Exam observers make a crucial contribution to the overall performance and integrity of the evaluation system by preserving standards of honesty and impartiality throughout the testing process.

Our objective is to provide a secured automated exam proctoring solution to ensure the integrity & authenticity of examinations. Exam proctoring eliminates the need to have human proctors/invigilators and ensures that there is no impersonation or cheating. In this paper, we propose an interface that will capture an image from a webcam and process the image to find visual cues of aberrant behavior [1]. The very first step is to detect the human face in the captured image. After trying multiple face detection algorithms mentioned in Kirti Dang et al. [15], we determined that a deep neural network-based algorithm was most appropriate for the case as it works well in occlusion, extreme face angles, & different scales. After face detection, head pose is estimated in pitch yaw form by projecting a 2D image onto 3D using world coordinates as a reference. The other approach would be to calculate the optic flow of the facial features [20]. Papers [18] and [17] recognize whether the mouth is open or closed using a radial basis network (RBF NN) as well as a k-nearest neighbor (KNN) clustering algorithm. Which we calculated by determining the Mouth Aspect-Ratio using facial features. We utilized extended Mask RCNN[24] for identifying the distance between students. In this paper, unlike In-Ho Choi et al. [20], which uses the pupil's center to determine gaze direction, a deep neural network model called EyeNet [6] is used.

2 Literature Survey

Yousef Atoum et al. [1] introduce an interactive analytics system with the ability to provide consistent online examination proctoring and a fully integrated online examination proctoring system equipped with audio and visual sensors to ensure academic integrity. They developed a two-stage hybrid interactive variety of decisions in which an ecosystem of classification models derives middle-level features from raw data and transforms these into high-level features to detect cheating.

Swathi Prathish et al. [2] built a comprehensive inference system capable of assisting instructors in monitoring students taking an online test. They established the ability to perform ground truth abstraction and yaw detection from the system's numerous proposed features. Although these two milestones are important, the ultimate goal is to improve the inference method. This is still being updated, but significant design decisions have already been taken. Currently, their project is limited to a single participant.

Keiko Sakurai et al. [3] discuss the system's use of EOG and RGB-D sensors, as well as an application that makes use of the proposed process. They discovered two characteristics of the current system: it is non-invasive; the EOG system does not require the wearer to wear a mask such as a goggle; it is simply a small electrode that is easy to use; as well as the device calibration is straightforward. Following that, they discovered that the proposed system has a higher level of precision than the current process. Additionally, gaze estimation over a broad range. They are not required to position that device; they only require the head's angle to be determined by template matching.

Song Wang et al. [4] proposed an advanced remote gaze estimation Automatic calibration method that does not require active user participation in estimating subject-specific eye parameters. The calibration-free technique for extracting subject-specific eye parameters results in a more effective solution because calibration errors are eliminated. It makes use of several cameras to assess the corneal centers and optic axes of two eyes, which makes it very expensive.

M. Sai Mounica et al. [5] use Neural Network to determine where the subject is looking at the screen. This allows the use of such a simple webcam, which significantly reduces the cost. However, this model includes calibration, which involves familiarizing the device with the appearance of the eye when the user's eyes are fixed on known positions on the screen. Additionally, it's indeed limited to screen-based gaze detection.

Zhengyang Wu et al. [6] utilize EyeNet, the very first unified deep neural network capable of solving multiple heterogeneous tasks relevant to off-axis camera eye gaze estimation. This approach to multi-task learning capitalizes on the inductive bias inherent in relevant tasks to boost results. Xiabing Liu et al. [7] suggested a technique for estimating head pose using a trained convolutional neural network on synthetic head pictures. Liu formulated the estimation of the head pose as just a regression problem. A convolutional neural network is equipped to recognize but not resolve the regression problem for head features. This technique is validated using both synthetic and real-world evidence. The experiment demonstrates that the procedure increases head pose estimation accuracy.

Zhang Ling et al. [8] An enhanced algorithm based on the appearance method for estimating head posture was suggested, as well as a strong emphasis on System-on-Chip (SoC) FPGAs. The results indicate that the device can estimate the head pose at a rate of approximately 16 frames per second because a photo has a scale resolution of 640 x 480 and precision of approximately 2.7 degrees in absolute error.

Xiangyang Liu et al. [9] presented a novel method for manifold clustering in which several manifolds are constructed, each of which characterizes its corresponding subspace of some topics. Liu begins by constructing a series of n-simples with subjects based on their pose picture similarity. Then, Liu demonstrated a supervised method for creating unique geometric structures for each learned manifold by combining manifold embedding and clustering. Experiments on the standard database show that the approach is resilient to identity variations and achieves a high degree of pose estimation accuracy.

Alejandro Newell et al. [10] present a new convolutional layer network topology to estimate a human's pose. The body's various spatial relationships are better captured when features are processed at all scales and combined. Alejandro demonstrates how repetitive bottom-up, top-down processing combined with intermediate supervision is crucial for network performance improvement. They allow development as a "stacked hourglass" network, referring to the successive phases of pooling and quantization used to generate a final stretch of prediction values.

Zihan Ren et al. [11] presented a tool for detecting and monitoring the human face in real-time. The proposed approach incorporates detection via Convolutional Neural Networks and monitoring via Kalman Filters. Convolutional Neural Networks are used to detect faces in the video, providing a more accurate detection method than conventional detection methods. When the face is significantly deflected or obscured, Kalman Filter tracking can be used to forecast the face's location. The goal is to improve the rate of face detection while also meeting real-time requirements. The system is based on the Caffe framework. The experimental results demonstrate that the system outperforms current techniques in terms of precision and maintains real-time efficiency.

Kewen Yan et al. [12] presented a system for facial recognition focused on Convolutional Neural Networks (CNN). Three convolution layers, two pooling layers, two fully connected layers, and one Softmax regression layer comprise this network. Stochastic gradient descent is being used to develop the feature extractor and classifier, which are capable of automatically extracting and classifying facial features. The Dropout approach is used to address the issue of overfitting. And during training and testing phases, the Convolution Framework for Feature Extraction system (Caffe) is used.

C. Anitha et al. [13] proposed a robust extraction method for the mouth area, as our application involves both facial and mouth regions. They recommend a new methodology for automatically extracting the mouth area. The proposed technique senses and monitors the mouth regardless of whether it is closed or open. They remove skin, complexion, and lip color using the color components.

Masao Shimizu et al. [14] described a detailed study of the precision and process of estimating eye-gaze direction in the human visual system. Extensive measurements indicate that the precision of eye-gaze prediction is 2-4 degrees at a distance of 1 m. They do, however, suggest that the frequency and propensity of point of perspective estimation errors vary. In general, a person with good eyesight makes accurate estimations. Additionally, measurements indicate that human vision could be based on an eyeball model that utilizes the iris or pupil's middle, instead of an iris shape model that utilizes the elliptical iris shape.

Kirti Dang et al. [15] article discussed and evaluated various face detection algorithms, including Viola-Jones, SMQT's SNOW Classification algorithm, Neural Network-Based Face Detection, and Support Vector Machine-Based Image Recognition. All of these facial recognition methods are compared based on their precision and recall values, which are determined using DetEval Software, which uses precise bounding box values around the faces to produce accurate results.

Ioana Bacivarov et al. [16] to model and monitor the lips, we created a computational active appearance model (AAM). The lack of comparison between lips and skin color presents a significant challenge when modeling the mouth area, which is deemed a most deformable facial feature. AAM needs a good initialization point to work accurately. Their approach begins with the extraction of critical information through chrominance analysis. The model's optimum parameters are then calculated using the AAM fitting technique. The model and fitting algorithm are described in detail, and preliminary findings on several databases are summarized.

Najafa Islam et al. [17] built an automated mouth recognition device that can aid in the control of a robotic system capable of self-feeding disabled individuals. To meet this need, they created and tested algorithms that can: 1) detect and monitor an individual's mouth in real-time; and 2) identify the mouth as open or closed. To identify and recognize the mouth's pose, a k-nearest neighbor (KNN) clustering algorithm has been used. The KNN algorithm categorizes image frames using four different methods: a histogram of directed gradients, the Harris-Stephens method, maximally stable external areas, and local binary structures. The findings of this study showed a high accuracy (87%) when three participants without disabilities were cross-validated tenfold. The study demonstrated that perhaps the models can detect a person's mouth posture in near real-time when they are eating a meal supported by a robot in a social environment.

Lim Ee Hui et al. [18] demonstrated a strong exploratory investigation of audio-visual speech recognition (AVSR) systems, owing to the growing number of multimedia applications requiring robust speech recognition systems. The use of image elements in AVSR is explained by the fact that speech is produced in both audio and visual modes and by the requirement for features that are immutable to acoustic perturbations. The AVSR system's output dependent on a robust collection of visual features derived from precise detection and monitoring of the mouth area. As a result, mouth monitoring is critical in AVSR systems. They present an enhanced version of the mouth tracking technique based on region-based convolutional neural networks (RBF NNs) and demonstrate its application to AVSR systems. The parameters of the RBF NN are adjusted using a revised extended Kalman filter (EKF). The simulation results indicate that the proposed approach performs well.

Michal Wlodarczyk et al. [19] established a method for estimating the head pose automatically, which is becoming increasingly necessary in modern computer vision applications. Accurate localization of facial landmark points and subsequent analysis allow the determination of a person's gaze direction or facial expression. These types of solutions are often used in connected vehicles, intelligent environments such as smart rooms, human-computer interfaces, and recognition systems. Surprisingly, the tasks of assessing the yaw or pitch angles of the face and localizing prominent points are approached separately in the literature. They integrate and discuss these two issues in this job. To begin, a thorough examination of the work of another researcher is presented. On this basis, they discuss the selected methods in light of their application to non-cooperative recognition systems. Experiments are performed on a newly developed, dedicated multi pose face dataset for this purpose.

In-Ho Choi et al. [20] proposed a method for detecting a driver's drowsiness that utilizes gaze path monitoring and head pose estimation. The head posture is estimated using the optical flow of facial features acquired using a corner detection algorithm. Three distinct components of the driver's head action are discernible: turning, shifting, and rotating. To determine the driver's gaze position, they map the pupil's center point using CDF measurements and measure the intensity of eye movement.

For the finding distance between objects Convolutional neural network (CNN) methods have been shown to outperform others on a range of object identification benchmarks [25]. Faster R-CNN [26] is a method for identifying objects in images that is based on deep learning. The faster R-CNN is divided into two stages. A Region Proposal Network (RPN) is the initial step, which suggests prospective object bounding boxes. Step 2, essentially Fast R-CNN [27], harvests characteristics from each prospective box using RoIPool and then does classification and boundary analysis. It might be possible to combine the characteristics used by both phases for better inference. Mask R-CNN uses a certain two-stage technique, with the first phase being identical (which is RPN). Mask R-CNN [28] produces a segmentation mask for each RoI in the second phase, in addition to analyzing the classification and box offset. It is contrary to other contemporary systems, which rely on mask predictions for categorization.

The extended Mask RCNN is indeed a deep learning-based technique that, in multilayered neural networks, integrates convolution, quantization, and fully connected layers. Following that, each item classification dataset is used to train the network's whole layer weights. The extended Mask RCNN is

used to identify visual objects Convolutional layers were supervised feature learning algorithms that are

effective for defining artifacts in a variety of scenarios.

3 Procedure

In this work, we aim to use visual clues to identify several different types of cheating behavior throughout an online examination session. The test taker will be under continuous monitoring during the duration of the exam. The image will be captured from the webcam at regular intervals and sent to the server for analysis. The image will be analyzed for four main features: number of faces in the image, head pose, mouth open or closed, and eye gaze direction. The image will be analyzed in the order shown in the flowchart (Figure 1). First, the number of faces in the image will be counted, if it's one, then the head pose of the test taker will be determined. If the person is facing the screen, then the image will be further processed to check if the mouth of the test taker is close. If it's close, the last analysis will be done to determine if the person is looking at the screen. In the event that any of these conditions fail, an appropriate warning message will be sent by the inference system based on the inference rules (Figure 2) to the test taker.

Algorithm

A. Face Detection using Caffe Framework

After trying multiple face detection algorithms such as Haar Cascade, Histogram of Oriented Gradients (HOG) Classifier, we concluded that the DNN module of OpenCV has the most accuracy. It supports a variety of frameworks for deep learning, which include Caffe, TensorFlow, and Torch/PyTorch. It is based on the Single-Shot Multibox detector (SSD) [22] and is built on the ResNet-10 architecture. This is the most accurate one, and runs in real-time on the CPU. It works well under substantial occlusion, extreme non-frontal images, and for different scales. However, if the picture is very large, this may trigger issues. Generally, we do not deal with photos larger than 3000 x 3000, so this should not be an issue.

S.no	Rule	Message
1	Face count is zero	No face found
2	Face count more than one	Multiple faces found
3	Head Pose more than threshold	Don’t Turn
4	Mouth Aspect Ratio greater than threshold	Don’t Speak
5	Eye Gaze greater than threshold	Look into screen

Figure 1: Inference Rules

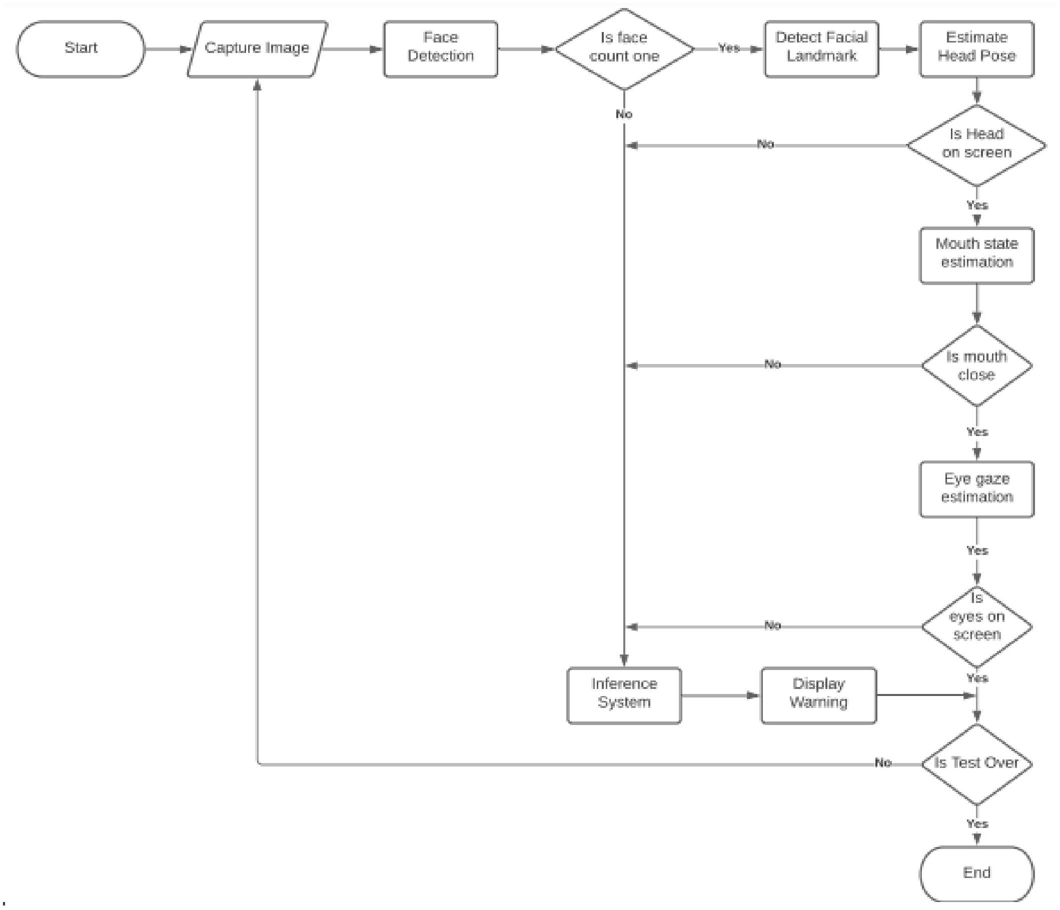


Figure 2: Flow Chart

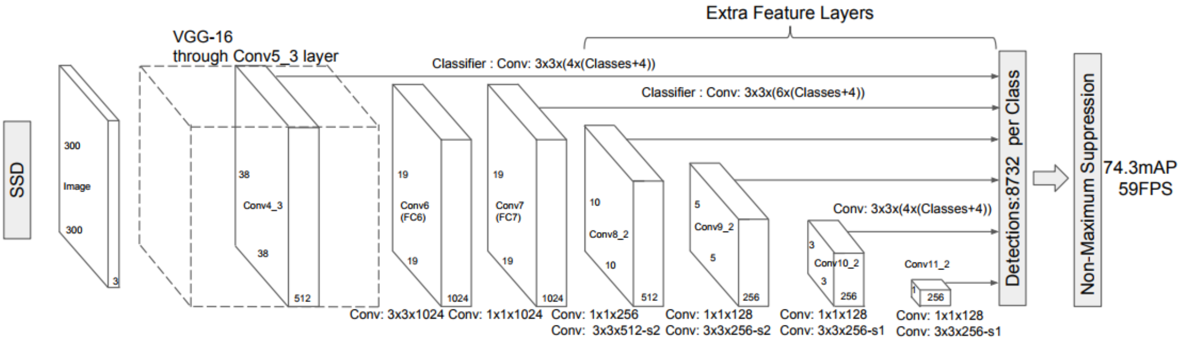


Figure 3: SSD Architecture

Two groups of files are needed when using OpenCV's neural network-based module with Caffe model:

- The file(s) with the extension. prototxt that describe that network model (i.e., the layers themselves)
- The caffemodel register, which includes the actual layer weights

Algorithm 1 Face detection using caffe model

```
Input: Image captured from a webcam
Output: Coordinates of the bounding box on the face
Using the -prototxt and -caffemodel files, load the model; save the model as a net;
while the webcam is on do
    capture image;
    create image blob;
    traverse the blob with the net;
    count number of faces;
    if face_count is one then
        further, process the image for visual cues;
    else
        raise alert;
    end
end
```

The problem of identifying face images is a subclass of the problem of shape prediction. A shape predictor uses an input picture to attempt to localize specific focal points along the shape. It is capable of locating and representing prominent facial regions such as the eyes, brows, nose, mouth, and jawline. They are detected by using pretrained DNN based model.

B. Extraction of Facial Landmarks

```
Algorithm 2 Facial landmark detection
Input: Captured Image
Output: Coordinate facial landmark
Load the pre-trained model file;
Pass the image to the loaded model;
Convert predictions to landmarks;
```

C. Estimation of Eye Gaze

The used architecture (Figure 4) is built around a visual representation for the 3D gaze direction (i.e., a gazemap), in which the network suits the original image data and from which further convolutional layers approximate the final gaze direction. Our neural network is composed of two components:

- Regression between the eye image and the gazemap.
- Regression between the gazemap and gaze direction.

Although any CNN architecture [23] can be used to implement the gazemaps, regression needs a completely convolutional architecture similar to that seen in human pose estimation. We adapt Newell et al [9].'s stacked hourglass architecture for this mission

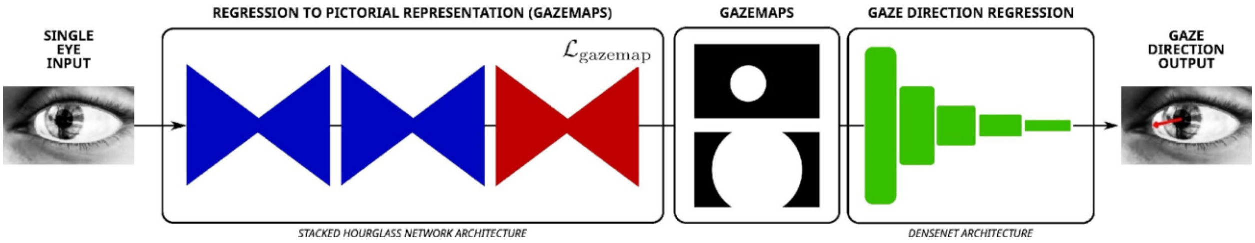
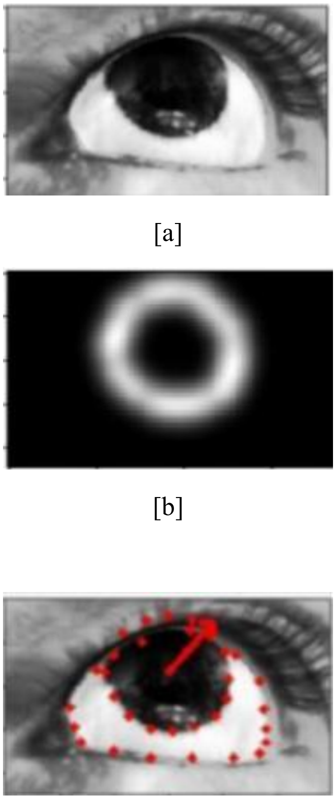

SINGLE EYE INPUT
REGRESSION TO PICTORIAL REPRESENTATION (GAZEMAPS)
$\mathcal{L}_{\text{gazemap}}$
GAZEMAPS
GAZE DIRECTION REGRESSION
GAZE DIRECTION OUTPUT
STACKED HOURGLASS NETWORK ARCHITECTURE
DENSENET ARCHITECTURE

Figure 4: CNN architecture

Pre-processing

A rectangular region around the eye is extracted from each captured image using eye points from facial landmarks. These are then normalized to have a width equal to the eye width (1.5 times the distance between eye corners). EyeNet, the pre-trained model, is then used to make predictions of the gaze direction (pitch, yaw) using this cropped image. The model architecture of EyeNet is based on the stacked hourglass model. The main modification was to add a separate pre-hourglass layer for predicting gaze direction. The output of the additional layer is concatenated with the predicted eye-region landmarks before being passed to two fully connected layers. This way, the model can make use of the high-level landmark features to predict the gaze direction.

For each preprocessed image (Figure 5.a), a set of gazemaps (Figure 5.b) corresponding to 34 eye region landmarks (Figure 5.c) was created. The model was trained to regress directly on the landmark locations and gaze direction in (pitch, yaw) form.



[a]

[b]

[c]

Figure 5: (a) Preprocessed image (b) Predicted gazemaps (c) Predicted landmarks and gaze vector

Algorithm 3 Eye gaze detection

```
Input: cropped_eye ← (left_eye, right_eye)
Output: Gaze direction in pitch, yaw form
for eye in cropped_eye do
    eye_predictions ← eyenet(eye);
end
smooth_eye_landmarks(eye_predictions);
for eye_pred in eye_predictions do
    gaze. append (eye_pred. gaze)
end
```

D. Estimation of the Head Pose

To calculate the three-dimensional position of an object in an image, the following information is required:

- Two-dimensional localization of these few points
- Three-dimensional positions with the same points
- The camera's intrinsic parameters

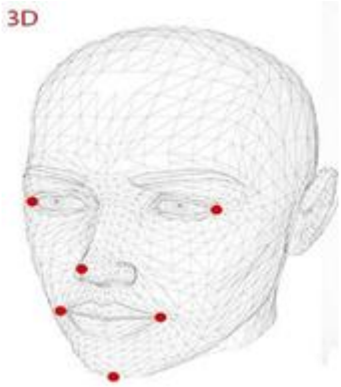


Figure 6: Facial Points for Head Pose Estimation

For head pose estimation, the main focus will be on the points: that top of the nose, the jaw, and the left and right corners of a left eye, the right and left corners of a right eye, the left and right corners the mouth. Figure 6 illustrates these points.

We do not need a three-dimensional model of the human in the photograph to obtain the three-dimensional positions of the facial landmarks. These points' three-dimensional positions in any arbitrary reference frame would suffice. We can use the World Coordinates as a reference (a.k.a Model Coordinates in OpenCV docs). To approximate the key parameters of the camera, we have used reality that the horizontal visual field of several webcams and mobile phones is between 50 and 70 degrees.

$$f = \frac{w}{2} cot\, cot\, \frac{\alpha}{2} \qquad (1)$$

Where, w = width of image, f = focal length, and α = field view of lens

After calculating the intrinsic parameter, the head pose is estimated using Open CV’s solve PnP function. Solve PnP implements several pose estimation algorithms that are adjustable via the parameter flag. The solvePnP function will take objectPoints, imagePoints, camera Matrix & distCoeffs as input where:

- ObjectPoints are three-dimensional points in the world coordinate system.
- ImagePoints are the two-dimensional points throughout the image that correspond to the image Points. f_x, f_y = f & c_x, c_y are the coordinates of the image center.
- $cameraMatrix = \quad [f_x\ 0\ c_x\ 0\ f_y\ c_y\ 0\ 0\ 1]$ (2)
- distCoeffs is the vector of distortion coefficients, which we will assume to be null in our case.

The solvePnP function will return in the calculated rotational & translation vectors. We know the coordinates (U, V, W) of a three-dimensional pixel Value in World Coordinates. Since we know the orientation R (a 33 matrix) and translating t (a 31 vector) of the coordinate system relative to the screen coordinates, we can use the following equation to determine the position (X, Y, Z) of a position P in the screen coordinate system:

$$[X\ Y\ Z\ ] = R[U\ V\ W\ ] + t \qquad (3)$$

Now By mapping the 3D point onto the 2D image, we can estimate the 2D positions of the 3d face points on the image. Given our knowledge of the points defining the 2D facial features, we can calculate the distance between predicted 3D points and the 2D facial features. When the predicted posture is perfect, the projected 3D points onto the picture plane will align nearly exactly with the 2D face shape. When the pose approximation is wrong, we can determine the re-projection error measure, which is the sum of the squared lengths between the estimated 3D points and the 2D facial feature points. By minimizing the re-projection error, an approximation of the pose is calculated from these.

```
Algorithm 4 Head pose estimation
Input: image_points, object_points, threshold
Output: Estimate head pose as rotational vector ($V_{rot}$)
Calculate focal_length ($f$) & optical_center ($c_x, c_y$);
Initialize camera_matrix & set dist_coeffs as null;
$P_{img}$ ← image_points
$P_{obj}$ ← object_points
$C_{mat}$ ← camera_matrix
$D_{coeff}$ ← dist_coeffs
$V_{rot}, V_{trans}$ ← solvePnP($P_{img}, P_{obj}, C_{mat}, D_{coeff}$)
if $V_{rot.z}$ > threshold then
    Report looking out of the screen;
    Raise alert;
else
    Report looking at the screen;
end
```

E. Mouth Tracking

Facial keypoints of lips can be used to calculate Mouth Aspect Ratio which is given by the formula:

$$MAR = \frac{|P_2 - P_8| + |P_4 - P_6|}{2 \times |P_1 - P_5|} \qquad (4)$$

By setting a threshold value, we can use Mouth Aspect Ratio to determine whether mouth is open or close.

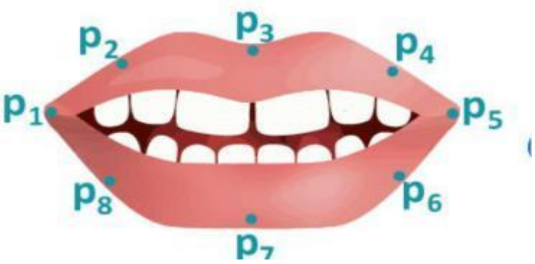


Figure 7: Facial Points for Mouth Tracking

```
Algorithm 5 Mouth tracking
```

```
Input: Key facial landmarks of face, threshold
Output: Detect whether the mouth is open or close
MAR ← mouth_aspect_ratio(facial_landmarks )
if MAR > threshold then
    Report mouth is open;
    Raise alert;
else
    Report mouth is close;
end
```

F. Distance identification between Student

```
Algorithm 6  Distance identification between student
Input: Image/video from one of the cameras
Output:  discovered the social distance between individuals

In dataset D s , for each picture I d :
From I d , create a feature vector.
Insert F p into the feature vector
Feed vector F p to a CNN architecture that has already been trained.
Return regions matrix.
Calculate interested region (R p )
Created shaded mask for interested region R p
Calculate the focal distance Fi with the interesting region
Calculate actual distance between individuals.
Finally, Yl represents the real distance between people; now compare it to the normal distance length.
Finally, the gap between the two people was discovered.
```

5 Dataset Description

For the purpose of evaluation, we have created our own dataset which includes 3 video each of 5 minutes duration of 3 different people. The frequency of each abnormal behavior is noted manually.

Implementation

1) Capture the image through webcam of the candidate appearing for the exam

2) Pass the extracted image through the face detection module, if the number of detected faces in the image is exactly one then proceed further otherwise raise an alert. Now from the face that has been extracted from the captured image, detect 68 key facial landmarks out of it. These landmarks will be used for further analysis. With the help of detected facial landmarks estimate the head pose and if the angle between the head and normal of the laptop screen is greater than a particular threshold then raise alert otherwise proceed further.

3) Estimate the state of mouth to check whether the candidate is speaking or not by calculating the mouth aspect ratio which is the ratio of the height of the mouth to the width of the mouth.

4) Estimate the eye gaze of the candidate to check if the candidate is looking into the screen or not and if candidate found to be looking outside the screen, then raise alert

5) utilizing Extensive Mask RCNN distance between student's heads is identified in such a way that anyone misleading by watching other's copies or asking another student can be identified.

6 Experimental Result

The experimental results were satisfactory and the parameter used for the analysis of system are precision, recall, and F1 measure. While experimenting, it has been observed that the model gives 95% precise results in moderate lighting conditions. The results are shown in the table below.

Table 1: Experimental Results

Data	Precision	Recall	F1
Video 1	0.94	0.92	0.93
Video 2	0.92	0.95	0.93
Video 3	0.93	0.91	0.92
Video 4	0.96	0.94	0.95
Video 5	0.98	0.95	0.96
Overall	0.95	0.93	0.93

7 Conclusion

After trying multiple face detection algorithms, we concluded that OpenCV's DNN based Caffe Model has the highest accuracy under non-ideal situations such as occlusion, extreme face angles, and different scales. For distance between students, the extended Mask RCNN does better in terms of identifying them. From the test taker's viewpoint, the device is affordable and easy to use. Although this method does not eliminate the need for a human proctor entirely. Certain methods of cheating are also possible with this device, such as a person who sits behind a laptop interacting with the test taker through writing. To eliminate cheating, we will need external hardware such as a spectacle camera that covers the test entire taker's field of view and applies computer vision to its feed, but this is not feasible because not every participant can afford it.

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

Author contribution: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Ethical and informed consent for data used: The research conducted research is not related to either human or animals use

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author.

Funding Statement*:* The author(s) received no specific funding for this study.

Acknowledgement: Thanks to the reviewing committee for their valuable points and notes.

References

[1] Atoum, Y., Chen, L., Liu, A.X., Hsu, S.D. and Liu, X., 2017. Automated online exam proctoring. IEEE Transactions on Multimedia, 19(7), pp.1609-1624.

[2] Prathish, S. and Bijlani, K., 2016, August. An intelligent system for online exam monitoring. In 2016 International Conference on Information Science (ICIS) (pp. 138-143). IEEE.

[3] Sakurai, K., Yan, M., Tamura, H. and Tanno, K., 2016, October. Comparison of two techniques for gaze estimation system using the direction of eyes and head. In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 002466-002471). IEEE.

[4] Wang, S., Wang, J., Peng, H., Gao, S. and He, D., 2016, December. A new calibration-free gaze tracking algorithm based on DE-SLFA. In 2016 8th International Conference on Information Technology in Medicine and Education (ITME) (pp. 380-384). IEEE.

[5] Mounica, M.S., Manvita, M., Jyotsna, C. and Amudha, J., 2019, March. Low Cost Eye Gaze Tracker Using Web Camera. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) (pp. 79-85). IEEE.

[6] Wu, Z., Rajendran, S., Van As, T., Badrinarayanan, V. and Rabinovich, A., 2019, October. Eyenet: A multi-task deep network for off-axis eye gaze estimation. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW) (pp. 3683-3687). IEEE.

[7] Liu, X., Liang, W., Wang, Y., Li, S. and Pei, M., 2016, September. 3D head pose estimation with convolutional neural network trained on synthetic images. In 2016 ieee international conference on image processing (icip) (pp. 1289-1293). IEEE.

[8] Ling, Z., Qing, X., Wei, H. and Yingcheng, L., 2017, October. Single view head pose estimation system based on SoC FPGA. In 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI) (pp. 369-374). IEEE.

[9] Liu, X., Lu, H. and Li, W., 2010, September. Multi-manifold modeling for head pose estimation. In 2010 IEEE international conference on image processing (pp. 3277-3280). IEEE

[10] Newell, A., Yang, K. and Deng, J., 2016, October. Stacked hourglass networks for human pose estimation. In European conference on computer vision (pp. 483-499). Springer, Cham.

[11] Ren, Z., Yang, S., Zou, F., Yang, F., Luan, C. and Li, K., 2017, November. A face tracking framework based on convolutional neural networks and Kalman filter. In 2017 8th IEEE international conference on software engineering and service science (ICSESS) (pp. 410-413). IEEE.

[12] Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G., Jiang, D., Tao, B., Xu, S. and Liu, H., 2019. Hand gesture recognition based on convolution neural network. Cluster Computing, 22(2), pp.2719-2729 .

[13] Anitha, C., Venkatesha, M.K. and Adiga, B.S., 2015, December. Real Time Detection and Tracking of Mouth Region of Single Human Face. In 2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS) (pp. 297-303). IEEE.

[14] Shimizu, M. and Fukui, K., 2015, October. Eye-gaze estimation accuracy and key in human vision. In 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA) (pp. 48-53). IEEE.

[15] Dang, K. and Sharma, S., 2017, January. Review and comparison of face detection algorithms. In 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence (pp. 629-633). IEEE.

[16] Bacivarov, I., Ionita, M.C. and Corcoran, P., 2008, June. A combined approach to feature extraction for mouth characterization and tracking. In IET Irish Signals and Systems Conference (ISSC 2008) (pp. 156-161). IET.

[17] Islam, N., Amiri, A.M., Forlizzi, J. and Hiremath, S.V., 2018, December. Automatic Mouth Detection for Self-Feeding. In 2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB) (pp. 01-03). IEEE.

[18] Hui, L.E., Seng, K.P. and Tse, K.M., 2004, November. RBF neural network mouth tracking for audio-visual speech recognition system. In 2004 IEEE Region 10 Conference TENCON 2004. (pp. 84-87). IEEE.

[19] Gupta, P., Sharma, V. and Varma, S., 2021. People detection and counting using YOLOv3 and SSD models. Materials Today: Proceedings

[20] Wlodarczyk, M., Kacperski, D., Krotewicz, P. and Grabowski, K., 2016, June. Evaluation of head pose estimation methods for a non-cooperative biometric system. In 2016 MIXDES-23rd International Conference Mixed Design of Integrated Circuits and Systems (pp. 394-398). IEEE.

[21] Choi, I.H. and Kim, Y.G., 2014, January. Head pose and gaze direction tracking for detecting a drowsy driver. In 2014 international conference on big data and smart computing (BIGCOMP) (pp. 241-244). IEEE.

[22] Gupta, P., Sharma, V. and Varma, S., 2021. People detection and counting using YOLOv3 and SSD models. Materials Today: Proceedings.

[23] P Gupta, M Shukla, N Arya, U Singh, K Mishra - Machine Learning for Critical Internet of Medical Things, 2022

[24] Gupta, P., Sharma, V. and Varma, S., 2022. A novel algorithm for mask detection and recognizing actions of human. Expert Systems with Applications, 198, p.116823

[25] A. Krizhevsky, I. Sutskever, G.E. Hinton, “Imagenet classification with deep convolutional neural networks”, In Advances in neural information processing systems, pp. 1097-1105, 2012.

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks.In NIPS, 2015.

[27] K. He, X. Zhang, S. Ren, J. Sun, “Deep residual learning for image recognition”, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[28] He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[29] Gupta, P., Shukla, M., Arya, N., Singh, U. and Mishra, K., 2022. Let the Blind See: An AIIoT-Based Device for Real-Time Object Recognition with the Voice Conversion. In Machine Learning for Critical Internet of Medical Things (pp. 177-198). Springer, Cham

[30] Gupta, P., Arya, N., Singar, C.P., Chaudhari, A., Singh, U. and Gupta, S., 2025. Safety of Pedestrians in AI-Optimized VANETs for Autonomous Vehicles via Real-Time Vehicle-to-Vehicle Communication. In AI-Driven Transportation Systems: Real-Time Applications and Related Technologies (pp. 169-181).

Cham: Springer Nature

[31] Gupta, P. and Singh, U., 2025. Evaluation of several yolo architecture versions for person detection and counting. Multimedia Tools and Applications, pp.1-24.

[32] Gupta, P. and Kulkarni, N., 2013. An introduction of soft computing approach over hard computing. International Journal of Latest Trends in Engineering and Technology (IJLTET), 3(1), pp.254-258

[33] Kushwaha, U., Gupta, P., Airen, S. and Kuliha, M., 2022, December. Analysis of CNN Model with Traditional Approach and Cloud AI based Approach. In 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 835-842). IEEE.

[34] Singh, U, Gupta, P., Shukla, M., Sharma, V., Varma, S. and Sharma, S.K., 2023. Acknowledgment of patient in sense behaviors using bidirectional ConvLSTM. Concurrency and Computation: Practice and Experience, 35(28), p.e7819.