Multimedia Tools and Applications (2023) 82:30375–30404
https://doi.org/10.1007/s11042-023-14604-w

An automated online proctoring system using attentive-net to assess student mischievous behavior

Tejaswi Potluri[1] • Venkatramaphanikumar S[1] • Venkata Krishna Kishore K[1]

Received: 4 October 2021 / Revised: 4 May 2022 / Accepted: 4 February 2023 /
Published online: 22 February 2023

Abstract
In recent years, the pandemic situation has forced the education system to shift from traditional teaching to online teaching or blended learning. The ability to monitor remote online examinations efficiently is a limiting factor to the scalability of this stage of online evaluation in the education system. Human Proctoring is the most used common approach by either asking learners to take a test in the examination centers or by monitoring visually asking learners to switch on their camera. However, these methods require huge labor, effort, infrastructure, and hardware. This paper presents an automated AI-based proctoring system- 'Attentive system' for online evaluation by capturing the live video of the examinee. Our Attentive system includes four components to estimate the malpractices such as face detection, multiple person detection, face spoofing, and head pose estimation. Attentive Net detects the faces and draws bounding boxes along with confidences. Attentive Net also checks the alignment of the face using the rotation matrix of Affine Transformation. The face net algorithm is combined with Attentive-Net to extract landmarks and facial features. The process for identifying spoofed faces is initiated only for aligned faces by using a shallow CNN Liveness net. The head pose of the examiner is estimated by using the SolvePnp equation, to check if he/she is seeking help from others. Crime Investigation and Prevention Lab (CIPL) datasets and customized datasets with various types of malpractices are used to evaluate our proposed system. Extensive Experimental results demonstrate that our method is more accurate, reliable and robust for proctoring system that can be practically implemented in real time environment as Automated proctoring System. An improved accuracy of 0.87 is reported by authors with the combination of Attentive Net, Liveness net and head pose estimation.

Keywords Online proctoring · Face spoofing · Face detection · Head pose estimation · Attentive -Net

✉ Venkatramaphanikumar S
svrphanikumar@yahoo.com

[1] Department of Computer Science & Engineering, Vignan's Foundation for Science, Technology & Research, Vadlamudi, Guntur, Andhra Pradesh, India

1 Introduction

The outbreak of the COVID-19 pandemic has severely affected different sectors of society, especially education. The education sector is recovering from the pandemic with the usage of digital devices to avoid a generational catastrophe in teaching, learning, and evaluation. Every component of education — including but not limited to classroom teaching and teaching methods, student learning and exposition, reciprocation and review, examination and testing, and assessment and evaluation — has undergone paradigm shifts to meet the expectations in the realms of academia, research, and innovation at large. The considerable shift in the expected skills, training, and acumen of graduates has initiated these changes, particularly in the domains of science, technology, and engineering. Information and Communication Technology supports knowledge transmission to remote places. Various technological platforms have been introduced into the education sector to decrease the transitional distance among the learners. The orientation towards blended learning is highly promoted as a combination of both conventional and online teaching. Online education offers the learners to attend the classes either in synchronous or asynchronous mode. The challenges of the virtual learning world demand an array of strategies to gauge the attentiveness of the learners in the e-classrooms. The pandemic made the teachers, institutions and systems to imbibe victorious strategies to usher in a learning system to "make your way by unexpected routes and tackle unguarded spots - problems that are novel in virtual learning. In addition, it facilitates the instructors to evaluate the learners also in both modes. Asynchronous mode of evaluation is not preferred by all the instructors for summative assessment. For synchronous mode of examinations, a human proctor manually examines the examinee and alerts if any mischievous behavior is observed. But, this physical proctoring does not identify all types of malpractice behavior of the examinee [31]. In line with the present scenario revisiting the existing/past practices to re-visualize, recast, and redesign the leading conventions and evolving as "competent" individuals, who can examine the students and online proctoring is the switch required. As per the current scenario, various Software industries have introduced proctoring software like ProctorU [40] to help the instructors in the conduction of examination in online mode, though there is still a need for trained proctors [10] or sensors [50] to do online monitoring of the examines to avoid malpractices. The authors in [2] state that the percentage of students indulging in mischievous behavior has been increased drastically during these online modes. Semi-automated proctoring software restricting the examinee to not to open any other browsers during the exam and also take the snaps of the students in random intervals to capture the behavior through various video Processing techniques like dominant Color [36], SURF, HARRIS, BRISK features [35], Color and shape features [37]. Automated Online proctoring systems uses Artificial Intelligence and Deep Learning techniques to identify the attentiveness of learner/examinee in virtual learning. Fully automated online proctoring facilitates an instructor to identify the misbehavior of the examinee with minimum requirements like internet connectivity, webcam, etc. using various deep learning models [15] and various convolutional neural networks [60]. The proposed automated proctoring identifies different physical activities of the examinee through the use of a camera. The proctoring system alerts the examiner if the examinee is not found or if multiple persons are detected or if face spoofing is identified. Furthermore, these proctoring systems are also used to assess the attentiveness of the examinee during the examination and as well as in the regular classrooms.

The proposed automated proctoring system – Attentive System helps to examine the attentiveness of the examinee with minimum false alarms. In this study, the authors proposed

Attentive-Net along with an Inception-Resnet to detect various activities of the examinees during an examination. The proposed system aims to detect the faces from the live video captured through the camera. To reduce the complexity of the model, the proposed system checks for the aligned faces in the frame and stops further processing and alerts the proctor if no aligned face is detected. For aligned faces, the proposed system extracts the facial features and detects the liveliness of the face. Our system also employs these features to estimate the head pose of the examinee.

In this work, a real-time online proctoring system – Attentive System is proposed with the following objectives:

- Face detection system using an Attentive Network to assess the attentiveness of the examinee.
- FaceNet approach to extract features using Inception-Resnetv1 blocks to carryout landmark localization and Affine Transformation to check the face alignment.
- Evaluation of the proposed method on customized and CIPL Datasets to estimate the accuracy in Head Pose and Face Spoofing respectively.

In Chapter 2, a thorough survey on literature published on various online proctoring systems and face detection systems is given. Chapter 3 illustrates various activities including facial detection, multiple face detection, no person identification, face alignment, liveness detection, head-pose estimation, and other mischievous actions performed by the examinee. The performance of the proposed work with discussions is presented in Chapter 4. Chapter 5 contains the conclusion and the future Scope.

2 Related work

Over the recent years, researchers have developed a wide spectrum of proctoring methodologies for online examinations. These methodologies are mainly divided into four categories: (i) No proctoring (ii) Manual Proctoring (iii)Semi-automated Proctoring and (iv) Fully Automated Proctoring. In the absence of online proctoring, mischievous behavior of the examinees is identified through eight online exam control procedures like offering the online exam only at one set time, exam to be accessible only for a brief period, to randomize questions, etc. [11]. In [51], network security issues were considered to prevent cheating of examinee. In manual proctoring, well-trained proctors are employed but this needs more manpower. Guo et al. [19] in their study took snapshots at regular intervals and observed the behavior of the examinee. In Semi-automated proctoring, 360^0 cameras are employed to get the view of the examinee from all angles. Some manual proctoring systems [25] uses group cryptography for controlling of ports and inputs. In desktop Robo [42] robots are also used to monitor the examinee's behavior. In [27], both front and rear camera are used to identify the examinee behavior. These methods are very costly and do not identify all types of malpractice behavior. In [24], AI based techniques are used to identify examinee behavior. In (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8) some fully automated online proctoring systems, eye-gaze tracking is used to monitor the examinees which requires high-end infrastructure. In [38], by using Dlib's facial keypoint detector and OpenCV, eye-gaze tracking is implemented. Object detection techniques (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8) are used to identify various objects used for cheating proctor. This system is limited to identifying only one type of misbehavior. Pratish et al. [38] used audio-visual traits to identify the mischievous behavior of the examinee. Atoum et al. [2] used two cameras to do behavior profiling to capture both what the examinee sees and what their behavior is. In this work, the authors proposed a semi-automated system in which they used four-vision-based capabilities like gaze tracking, mouth open or close, face accounting, and mobile phone detection.

Face Detection Facial detection plays a prominent role in the online proctoring application. Despande et al. [14] used Region-Based Convolutional Neural Networks (R-CNN) for feature extraction from the faces detected. Histogram Intersection Kernel with Support Vector Machines (SVM) classifier is used for the classification. Sun et al. [47] used a faster R-CNN framework including feature concatenation, hard negative mining, multi-scale training for face detection. This framework needs to be boosted for different patterns of the face and has a computational burden. Resnet50 based Caffe model (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8) identifies multiple persons before the camera and as well as the absence of a person before the camera. It is observed that the background affects the accuracy of the model. In [12], Haar Cascade algorithm combined with three weak classifiers based on skin hue-histogram matching, eyes detection and mouth detection. This is not suitable for the real time applications due to its high computational time. In [48], authors applied skin filter and entropy rate super pixels (ERSs) to obtain face candidates. Then, angle compensation, refinement and symmetry extension are applied to improve accuracy of face detection. This technique is robust to in-plane rotation even if out-plane rotation exists. Chong Li et al. [29] improved Yolov3 for the face detection by changing detection layer and choosing soft max loss function to maximize the difference of inter-class features.

Face alignment Face alignment verification is not performed in this methodology which plays key role in proctoring. In [26], authors proposed PCA (Principal Component Analysis) to reduce the large amount of data storage. Self-space projection is used in face detection by defining a number of facial images with the own vectors of the covariance matrix. Authors in [54] designed a hierarchical attention mechanism instead of convolution neural networks. Authors proposed Gaussian kernels to search for proper positions and scales of local regions to extract informative features of the face. Then LSTM is used to model relations between the local parts and adjust their contributions to the detection tasks. The LSTM introduces additional computational cost. In [7], a three-level Cascaded Neural Network learned via residual learning was used for face alignment identification. These subnetworks are based on residual part heatmap regression. Wenyan wu et al. [53], uses boundary information of the face to predict landmarks coordinates. Face geometric structure is drawn using landmark coordinates which is used for verifying face alignment. The boundary aware face alignment algorithm does not perform well for very small faces as landmarks are not accurately reconstructed from boundary. In [45], authors proposed multiple shape prediction layers for detecting certain cluster of semantically relevant landmarks. As training should be performed for each cluster of landmarks, training cost increases. Authors in [61] used both non-linear layers and linear layers for face alignment. The non-linear layers with cosine activations encodes relationships between representations of images and shapes of facial landmarks. The linear layers are used to encode landmark correlations by low-rank learning. This works better for heavy occluded images. Authors of [6] estimated the facial landmarks by using the image intensities. Rotation Vector is calculated using Rodrigues rotation formula for verifying the face alignment. Authors [8] proposed discriminative Feature Learning method for face alignment based on the fully convolutional Network. Affine Transformation is used on hourglass network for aligning the face. In [30], authors analyze the texture of the facial images using multi-scale local binary patterns. The high-level features are robust to real time tasks than the textural features of image, deep features are extracted. Gustavo et.al [13], integrates deep texture features with LBP descriptor and gives as input to convolution neural network. The Caffe- framework is considered as a base framework.

Face Spoofing In [28], Local Binary Patterns (LBP) layers were combined with convolution layers for face spoofing detection. The VGG-face network is used for face detection. For face spoofing, the VGG-face network needs to be fine-tuned. The accuracy of face spoofing detection depends upon the features extracted from the face. Authors in [46], designed a descriptor called spatial pyramid coding micro-texture feature to characterize the local appearance information and then uses SSD to conduct end-to-end face presentation detection. Then, fake faces are detected by designing template-face matched binocular depth feature. This method requires a special hardware device like binocular camera. In [56], authors used convolution neural networks named G-Net to rough estimate the head pose and localize seven primary landmarks. One more CNN LNet is used to learn local CNN features and predict pose. Overfitting is adjusted to only some extent.

Head Pose Estimation In [5], the Regression algorithm applied to a web-shaped model was used for head-pose estimation. This method also predicts the value of the dependent variable for all three angular values. Pitch angular error is a limitation to this method. In [43], the authors used face identification, multiple face detection, and head-pose estimation for the identification of malpractices during examinations. In this work, the authors used yaw angle for head pose estimation and achieved 80% accuracy. Yaw angle was used to obtain the three-dimensional rotation angles of the face. To identify the malpractices, three-dimensional angles are not required which increases the complexity of the model. But this method identifies only some types of malpractices.

Mobile Detection Authors in (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8) used pre-trained Yolov3 for mobile phone detection. Yolo model is trained on Microsoft Common Objects in Context (COCO) dataset in which mobile phones are identified as class 67. Yolov3 is based on Darknet consisting of 53 layers. In [17], authors combined top-down and bottom-up connections to fuse features across scales. Authors also combined with various backbone models in RetinaNet Framework. Authors in [32] used weighted bi-directional feature pyramid network and Efficientnet backbones for object detections.

Vaishali and Singh (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8) used the ResNet50 based Caffe model for online proctoring and implemented various modules such as face detection, head pose estimation, eye ball-tracking, mouth opening, and phone detection. In the performance evaluation, the authors mentioned that they have not received the expected accuracy due to issues with face alignment, and pre-processing. Table 1 presents a thorough review of the literature of online proctoring systems including modules such as face detection, object detection, and head-pose estimation.

Table 1 Summary on existing models

Ref	Methodology	Dataset	Metrics
[20]	Multi-layer feed-forward deep neural network	Yale Faces	Accuracy-95.4%
[38]	Tree-Structured Model and Yaw angle.	Own Dataset of 39 videos each of 2 minutes duration on average	Accuracy-80%
[2]	Minimum Average Correlation Energy with SVM	OEP dataset (Prepared by own)	True Detection Rate-87% False Alarm Rate-2%
[14]	Viola-Jones Algorithm, PCA, ANN	BioID Face Database	Accuracy-94%
[49]	Caffe Model	CoCo, Kitti, Open Image	Accuracy-81%
[1]	6DoF pose in a Faster R-CNN–based framework	AFLW2000-3D, BIWI Wider Face	Easy-level−90%; Medium-level- 89.1%; Hard-level- 83.9%
[21]	MLP for regression of pose angles.	BIWI and AFLW	Mean Absolute Error- 4.06
[43]	ResNet 50 with Cross-Entropy	AFLW2000, BIWI	Mean Average Error- 6.155
[58]	SSR-Net Based FSA-Net	300 W-LP, AFLW 2000, BIWI Dataset	Mean Absolute Error- 4.00
[16]	R-CNN is used for Feature Extraction and Histogram intersection kernel with SVM	VOC 2007, VOC 2010–12	Mean Average Precision- 62.4
[9]	Two Stream Convolution Neural Network	CASIA-FASD, REPLAY-ATTACK and OULU	Attack Presentation Classification Error Rate (APCER) and Bona Fide Presentation −3.6% Classification Error Rate−2.2%
[34]	CNN and adaptive gradient methods	Prima Head pose dataset, AFLW and AFW datasets	Accuracy (Pitch)-73.9% Accuracy (Yaw)-66.6%
[52]	Google Net and Coarse to fine strategy	Biwi Kinect Head Pose Dataset, Pointing'04 dataset and Boston dataset	Accuracy (synthetic images)- 99.92% Accuracy (Real images) – 92.35%
[23]	GoogleNet with Multiple Regression Loss	ALFW2000 dataset	Mean absolute error of the Euler angles (EMAE) – 4.556 Mean absolute error of the quaternions (QMAE) -0.0248
[3]	POSEidon model along with Deterministic conditional GAN model	Biwi Kinect Head Pose and ICT-3DHP, and Pandora	Accuracy (Head)(Pandora Dataset) - 73.6%
[57]	Multi-Layer Perceptron with scatter max layer.	Biwi head pose dataset	MAE-1.26
[22]	Object Relation Module with Region Proposal Network	COCO dataset	mAP accuracy – 31.9
[33]	Local Binary Pattern Histogram Algorithm	Customised Dataset	Accuracy – 62%
[39]	Haar Cascade Method	Instagram Data	Accuracy −71.48%
[59]	Multi-Class Classification Network	Wider face and FDDB datasets	Face Detection Accuracy – 84.2%
[4]	CNN based cascade structure face detector	Masked Face dataset	Accuracy – 86.6%
[41]	Faster RCNN	JC2463 and AEC912 datasets	Precision −70.9% Recall −48.4% F-Measure −57.5%
[18]	Local Binary Patterns	UCCS datasets	Face detection rate- 91.2%

By considering the limitations of existing systems, authors proposed a real time system which requires to meet the objectives. Authors proposed Attentive system with less complexity and less computational time. The Attentive system identifies the face from the frame using three staged Attentive-Net. Features are extracted only from the aligned faces using the Face net algorithm which is based on inception-resnetv1 blocks. Further, the proposed system records if the examinee is seeing towards left, right, by estimating the head pose using SolvePnp Equation. The sequential layered Liveness Net Model is employed in this system to detect face spoofing.

3 Methodology

The proposed work is used to detect the face and used to check head alignment according to pose. Further, this system also detects the presence of the examinee and verifies the multiple faces and other accessories notification. The architecture of the proposed work is presented in Fig. 1. Figure 1 shows that the examinee's video is continuously recorded through a webcam and then is divided into frames for further processing. As a part of the first module, all the faces within the frame are detected using the top, bottom, left, and right coordinates, and a customized border is drawn around the face for visualization. The three-stage cascaded neural network-based model Attentive-Net is used for face detection. Attentive-Net is formed by taking Multi-Task Cascaded Convolution Networks(MTCNN) [55] as base model along with naïve-inception module and face alignment verification. The frame with a minimum size of 80 $\times$ 80 is given as an input to the face detection algorithm. In our online proctoring system, face detection alone will not serve the purpose. We also need to detect the liveness and estimate the head pose of the face. To detect the liveness, first, the faces are checked for the alignment using Affine transformation matrix. For the aligned faces Inception-ResNet v1 architecture-based model, FaceNet is used to extract the features of the face. This feature extraction algorithm yields a feature vector of length 128 comprises landmark points such as nose tip, chin, left eye left corner, left mouth corner, right mouth corner, etc. From these landmarks, the spoofed face is identified and an alert is given to the examiner.

The same landmarks extracted by Attentive Net is used to find out rotation vector and translation vector of aligned face to estimate head pose either as left or right. The head-pose estimation is done for the live face image to avoid malpractice issues. In this process, if the examinee is taking help from any other devices or the neighborly people, the examiner is notified. Cell phones, laptops, and books in the frame are detected, and an alert is generated for the examiner. The following section describes the process of video acquisition followed by proctoring.

3.1 Video acquisition and frames extraction

In the proposed model, a webcam is used to continuously stream and capture the video of the examinee and convert them into frames. Furthermore, keyframes are extracted based on the motion detection or according to the changes in the visualization. Malpractices may be found even without major changes in motion and visualization. So, in our work, instead of extracting only keyframes, all the frames are processed. For all the frames, faces are detected by using the following process.

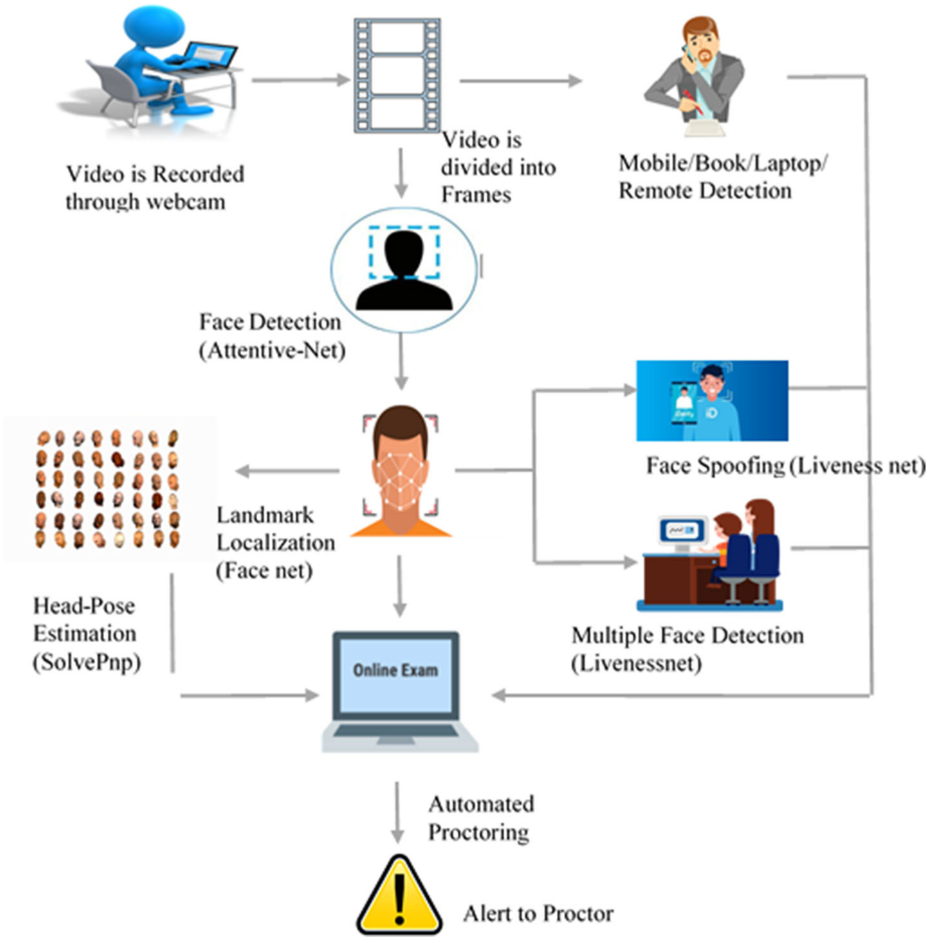
Video is Recorded through webcam
Video is divided into Frames
Mobile/Book/Laptop/ Remote Detection
Face Detection (Attentive-Net)
Face Spoofing (Liveness net)
Landmark Localization (Face net)
Head-Pose Estimation (SolvePnp)
Multiple Face Detection (Livenessnet)
Online Exam
Automated Proctoring
Alert to Proctor

Fig. 1 Architecture of the proposed attentive system

3.2 Face detection using Attentive-Net

All the frames detected are inputted to the Face Detection module. Face detection is used to detect the frontal faces from the frames captured through the webcam. Attentive-Net is used to detect the faces from the frame and resized into different scales and a scale pyramid is formed. Further, the image is smoothed with a sequence of smoothing filters, which has a radius twice the radius of before. The smoothing is carried with the deferrable smoothing filter f is mentioned in Eq. (1).

$$\widehat{s} = \sum_{m=-2}^{2} \sum_{n=-2}^{2} f(m,n).s_0(i-m)(j-n) \tag{1}$$

Further, down sampling is applied to reduce the frame size by half of its size with smoothing.

$$s_{l+1}(i,j) = \sum_{m=-2}^{2}\sum_{n=-2}^{2} f(m,n).s_l(2i-m)(2j-n) \tag{2}$$

Considering each image has only one face and multiple landmarks in real world, each image is represented as (x_n, y_n, z_n), $n = 1,2,\ldots,N$. Here x_n represents the n-th image sample, $y_n = c$, $c = 0,\ldots,C\text{-}1$ is Face label and, $z_n = [\,z_n^1,\ldots,z_n^M]^T$ is the facial landmarks for the n-th image sample. If the n-th image sample has the m-th facial landmark is set as $z_n^m = 1$, otherwise $z_n^m = 0$. Therefore a given dataset is denoted as $(X,Y,Z) = \{(x_n, y_n, z_n), n \in \{1,2,\ldots,N\}\}$. Three stages (not directly connected) are implemented on a scale pyramid, but the outputs of the earlier stage are fed as input to the next stage.

In stage 1, a full CNN called Naive-Proposal network (NP-net) is applied to extract facial windows and bounding box regression vectors. For each scaled frame, a 12 × 12 stage 1 kernel scans the image from the top left corner (0,0) to (12,12). In this connection, any face we find will be tracked. To achieve the real time performance, the same process is repeated with stride 2 reducing the number of operations to a quarter of the original. Within each of these kernels, 3 convolutions are run through with 3 × 3 kernels along with Naive Inception module. The naive inception module consists of NP-net contains 1 × 1 convolution, 3 × 3 convolution, 5 × 5 convolution along with max-pooling. For each convolution and max-pooling 16 × 16 filter is used. The final concatenated output results in 64 channels. The output of all these is concatenated and is given as input to the third convolution layer. After every convolution layer, the Parametric Rectified Linear Unit (PReLU) layer is added to multiply every negative pixel with alpha. To remove every other pixel and leave only the largest one, the max-pooling is added after the first PReLU layer. The network is split into two layers after the third convolution layer. The activations are passed on to both layers (convolution 4–1, convolution 4–2) but, the SoftMax function is applied to only one of the above convolution layers (convolution 4–1). The SoftMax adds up to the probability that there is a face or there is not a face. The convolution 4–2 outputs the coordinates of the bounding boxes. For each candidate box, we predict the offset between it and the nearest ground truth (i.e., bounding boxes left, top, height and width). As the learning objective is formulated as regression problem, Euclidean loss for each sample is implemented as below:

$$\text{Loss} = \left\|\widehat{p}(y) - p(y)\right\|_2^2 \tag{3}$$

Where, $\widehat{p}(y)$ is the regression target obtained from network and $p(y)$ is the ground truth coordinate. This results in overlapped bounding boxes with different confidences. Non-maxima suppression is implemented on all the bounding boxes of all kernels to remove bounding boxes with low confidence. For all the bounding boxes (b_{oi}) with different confidences (C_i), remove the bounding boxes with overlap (IoU ≥ NMS_t), where IoU: Intersection over Union, M: argmax of C, NMS_t: NMS threshold,

$$IoU(M, b_{oi}) \geq NMS_t, \text{where } B_o \leftarrow B_o - b_{oi}; C \leftarrow C - C_i \tag{4}$$

NMS threshold is considered as 50%. All bounding boxes are reshaped into a square in stage 1. The complete structure of the process can be explained with the following algorithm.

Algorithm 1. NP-NET

```
1: $I(x,y) \leftarrow frame$
2: $NL \leftarrow no.of.layers$
3: for i = 1 to NL do
4:   for $x = 0$ to width do
5:     for $y = 0$ to height do
6:       $g(x,y) \leftarrow \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} w(dx,dy) * I(x+dx, y+dy)$
7:       if $g(x,y) < 0$ then
8:         $g(x,y) \leftarrow \alpha * g(x,y)$ //Convolution Layer in P-NET
9:       end if
10:      if i = 1 then
11:        perform 2 x 2 max pooling //Add max pooling only after first layer
12:      end if
13:      if i = 1 then
14:        Insert Naive-Inception Module with filter size 16 x 16 //Results in 64 channels
15:      end if
16:    end for
17:  end for
18: end for
19: $\sigma z \leftarrow e^{C_{1i}} / \sum_{j=1}^{K} e^{C_{1j}}$ //Soft max operation on input vector $C_{1i}$
20: $NMS_t \leftarrow 50$ //Non Maxima Supression threshold is set to 50
21: for All bounding boxes do
22:   if $IoU(M, C_i) \geq NMS_t$ then
23:     Remove the bounding box $C_i$ //If Intersection of Union is less than threshold, Boundary box will be removed.
24:   end if
25: end for
```

NR-Net (Refine Network) is similar to NP-Net with more layers. In stage 2, NR-Net padding is applied to reshape out-of-bound boxes and also filter them. For the bounding boxes related to occluded or partial faces, padding is carried with zeros. NR-Net has 3 convolutions run through with 3 × 3 kernels along with Naive Inception module. The naive inception module consists of NP-net contains 1 × 1 convolution, 3 × 3 convolution, 5 × 5 convolution along with max-pooling. For each convolution and max-pooling 48 × 48 filter is used. The final concatenated output results in 192 channels. The output of all these is concatenated and is given as input to the third convolution layer. All the bounding boxes obtained from stage 1 are resized to 24 × 24 and their values are normalized between −1 and + 1. Similar to NP-Net, NR-Net also splits into two layers which returns classification and bounding boxes with confidence as shown in the following algorithm. Bounding boxes with low confidence are further deleted using the Non-maxima suppression (NMS) logarithm.

Algorithm 2. NR-NET

```
1: $Input \leftarrow I1(x,y)$ //Get Input from P-NET
2: $NL \leftarrow no.of.layers$
3: for i = 1 to NL do
4:    for $x = 0$ to width do
5:       for $y = 0$ to height do
6:          $g1(x,y) \leftarrow \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} w(dx,dy) * I1(x+dx, y+dy)$ //Convolution operation in R-NET
7:          if $g1(x,y) < 0$ then
8:             $g1(x,y) \leftarrow \alpha * g1(x,y)$ //Multiply every negative pixel with alpha
9:          end if
10:         if i = NL-2 then
11:            perform 2 x 2 max pooling
12:         end if
13:         if i = NL-1 then
14:            perform 3 x 3 max pooling
15:         end if
16:         if i = NL-1 then
17:            Insert Naive Inception module with filter size 48x48 //Results in 192 channels
18:         end if
19:      end for
20:   end for
21: end for
22: $\sigma z \leftarrow e^{C_{2i}} / \sum_{j=1}^{K} e^{C_{2j}}$ //Soft max operation on input vector $C_{2i}$
23: $NMS_t \leftarrow 50$ //Non Maxima Supression threshold is set to 50
24: for All bounding boxes do
25:    if $C_i$ is occluded or partial then
26:       Padd with zeros //Padding is done to all the occluded and partial bounding boxes
27:    end if
28:    if $IoU(M, C_i) \geq NMS_t$ then
29:       Remove the bounding box $C_i$ //If Intersection of Union is less than Threshold, Bounding box will be removed
30:    end if
31: end for
```

In stage 3, Output Network (O-Net), is implemented for the out-of-bound boxes with padding as shown below.

Algorithm 3. O-NET

```
1: $Input \leftarrow I2(x,y)$ //Get Input from R-NET
2: $NL \leftarrow no.of.layers$
3: for i = 1 to NL do
4:   for $x = 0$ to width do
5:     for $y = 0$ to height do
6:       $g2(x,y) \leftarrow \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} w(dx,dy) * I2(x+dx, y+dy)$ //Convolution operation in O-Net
7:       if $g2(x,y) < 0$ then
8:         $g2(x,y) \leftarrow \alpha * g2(x,y)$ //Multiply every negative pixel with alpha
9:       end if
10:      if i = NL-3 or NL-2 then
11:        perform 3 x 3 max pooling
12:      end if
13:      if i = NL-1 then
14:        perform 2 x 2 max pooling
15:      end if
16:    end for
17:  end for
18: end for
19: $\sigma z \leftarrow e^{C_{3i}} / \sum_{j=1}^{K} e^{C_{3j}}$ //Soft max operation on input vector $C_{3i}$
20: $NMS_t \leftarrow 50$ //Non Maxima Suppression threshold is set to 50
21: for All bounding boxes do
22:   if $C_i$ is outofbound then
23:     Padd with zeros //Padding is done to all the out-of-bound boxes
24:   end if
25:   if $IoU(M, C_i) \geq NMS_t$ then
26:     Remove the bounding box $C_i$ //If Intersection of Union is less than Threshold, Bounding box will be removed
27:   end if
28:   $L_b \leftarrow -(p(x) * log t(x) + (1-p(x)) * log(1-t(x)))$ //Binary Cross Entropy Loss function
29: end for
```

The O-Net also splits into three layers which output the probability of face being in the box, coordinates of the bounding boxes, and the coordinates of the facial landmarks. In this stage, all the boxes are resized to 48 × 48 and then five facial landmarks like eye tips, nose tips, and lips tips are extracted. Finally, classification is applied to categorize the features as either face or no face. The binary cross entropy loss function is used for binary classification as per Eq. (5) given below, where *t(x)* is the probability produced by the network that indicates sample to be a face. The notation p(x) ∈{0,1} denotes ground-truth label.

$$L_b = -(\mathrm{p(x)}.\log t(x) + (1-p(x)).\log(1-t(x))) \tag{5}$$

The face landmarks are given as an output from the cascaded model. Different types of training images in learning process like face, nonface and partially aligned face are considered. In this case, Euclidean loss or Cross entropy loss are not alone used. The cross-entropy loss function is implemented for the sample of background region. This can be implemented directly with a sample type indicator as below:

$$\mathrm{Min} \sum_{i=1}^{N} \sum_{j\in\{\mathrm{det,\ \ box,landmark}\}} \alpha_{j,} \beta_i^j, L_i^j \tag{6}$$

Where, N is the number of training samples and $\alpha_{j,}$ denotes on the task importance, L_i is the Euclidean Loss. We use $\alpha_{det} = 1$, $\alpha_{box} = 0.5$, $\alpha_{landmark} = 0.5$ in NP-Net and NR-Net, while $\alpha_{det} = 1$, $\alpha_{box} = 0.5$, $\alpha_{landmark} = 1$ in output network. $\beta_i^j \in \{0, 1\}$ is the sample type indicator. In these cases, stochastic gradient descent is used to train the CNNs. Figure 2 shows the network architecture of the proposed cascaded model. The following section describes the extraction of landmarks of the faces detected from three staged networks of Attentive-Net.

The first block of the architecture is the Naïve-proposal network which results in the classified faces and bounding boxes. The second block of the architecture is Naïve-Refine Network which performs further Non-Maxima Suppression and results in face classification and bounding boxes. The third block is Output Network which results in Face Classification, Bounding boxes and Landmarks Localization.

3.3 Landmarks localization

The extracted landmarks are localized to verify the alignment of the face image which is a submodule of Attentive-Net. The process continues only if the face detected is aligned otherwise alert is generated for the examiner by aborting the process. Generally, the accuracy of the facial alignment depends on the feature's extraction. To check the face alignment, the landmarks obtained from the three-staged networks of Attentive-Net are considered as original points. The average of the face positions according to the face points is considered as the base point. Further, similarity transform is used to compute the difference between original and base points using the Frobenius norm as in Eq. (7).

$$\|B\|_F = \left[\Sigma_{i,j}\ \text{abs}\ \left(b_{i,j}\right)^2\right]^{1/2} \tag{7}$$

The norm values of original and base matrices are used to calculate the covariance using Eq. (8):

$$Cov\left(B_o, B_b\right) = \Sigma((B_{oi} - B_{omean})*(B_{bi} - B_{bmean}))/ND \tag{8}$$

where B_o is Frobenius norm of the original matrix; $B_{o\ mean}$ is the mean of Frobenius norm of Original matrix; $B_{b\ mean}$ is mean of Frobenius norm of Base matrix, and ND is the number of data variables. Then, the singular value decomposition (SVD) of covariance such that UDV^T = SVD (Covariance) is computed, where U and V are the orthonormal left and right matrices of singular vectors, respectively. D is a diagonal matrix of decreasing positive singular values. Thereafter, the affine matrix is computed using u, d, and v values. Here, only the rotation about the base points is carried by using the affine transformation matrix using Eq. (9).

$$\begin{bmatrix} cos\theta & \sin\theta & 0 \\ -\sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

The rotation matrix is calculated with pixel mapping is given in Eq. (10).

$$R' = \begin{bmatrix} x\cos\theta - y\ sin\theta \\ x\sin\theta + y\ cos\theta \end{bmatrix} \tag{10}$$

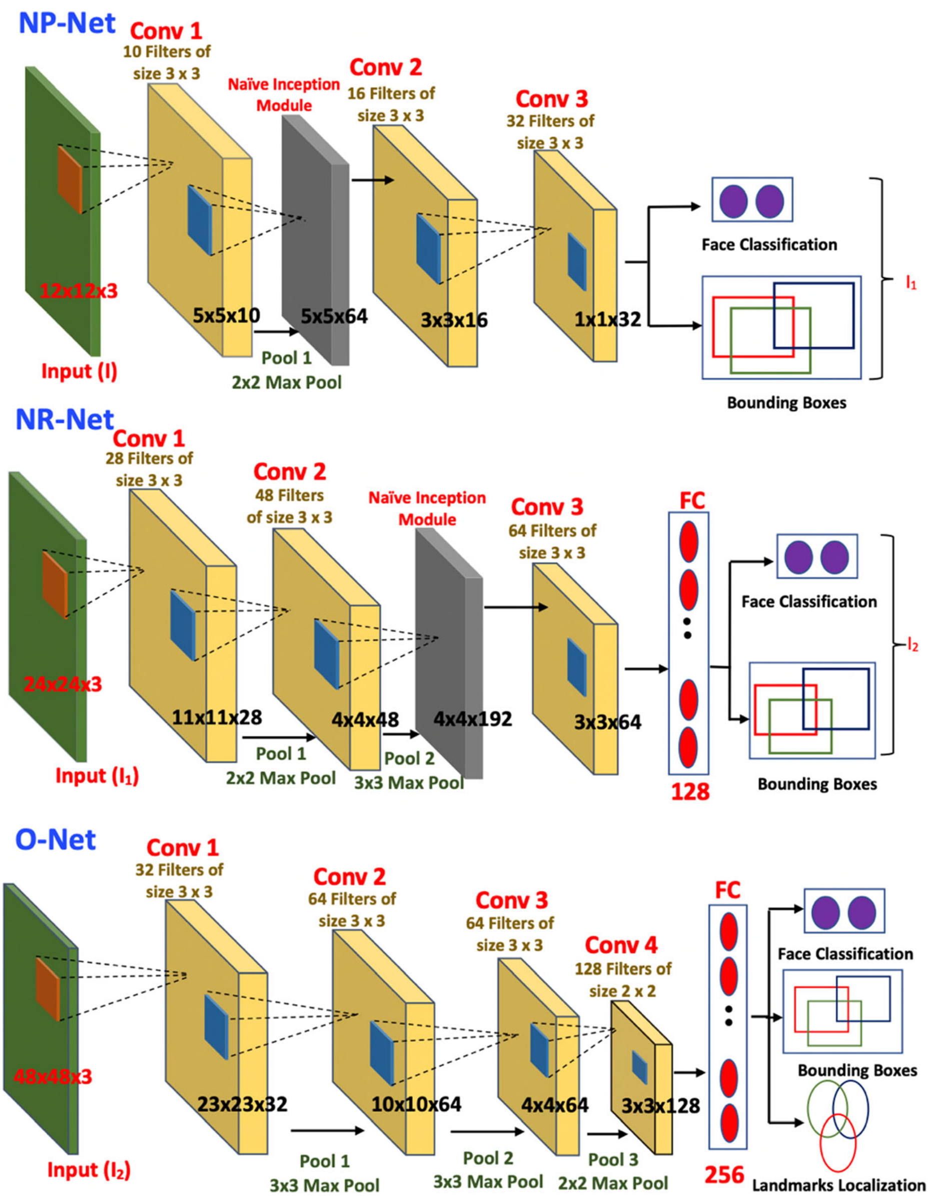

Fig. 2 Architecture of proposed Attentive-Net.

The affine transformation is applied on the original points and the length of the aligned face is 160. All the aligned faces of a frame are marked, and feature extraction is done only if a single aligned face is detected in a frame. The algorithm for the above-described process is as shown below. The input given to the Face alignment algorithm is the landmarks of the faces extracted by Attentive-Net. L1 are the landmark positions. A1 are the Average Face Positions.

Algorithm 4. Face Alignment

```
1: $\lambda 1 \leftarrow reshape(L1)$
2: $\beta 1 \leftarrow reshape(A1)$
3: $\lambda_{mean} \leftarrow mean(\lambda 1)$
4: $\beta_{mean} \leftarrow mean(\beta 1)$ //Compute the mean of original points and base points
5: for all $\lambda 1$ do
6:     $Cov \leftarrow covariance(\lambda 1, \beta 1)$
7: end for
8: $u, d, vt \leftarrow SVD(Cov)$//Singular value decomposition of covariance
9: if $det(Cov) < 0$ then //Compute the affine matrix
10:     if $d[1] < d[0]$ then
11:         $s[1,1] = -1$
12:     else
13:         $s[0,0] = -1$
14:     end if
15: end if
16: $ro \leftarrow u * s * vt$
17: $T_b \leftarrow \beta_{mean}.transpose() - c * ro * \lambda_{mean}.transpose()$
18: $T_m \leftarrow c * ro$ //Compute the similar transform $T_b, T_m$
19: $R' = RotationMatrix(T_b, T_m)$ //Calculate the Rotation Matrix
20: $positions(Le, Ri, Ce) = Affinetransformation(R')$ //Get the positions
    using Affine Transformation on Rotation Matrix
```

In (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8), the process is implemented for all the faces detected in the frame which increases the complexity. The examinee's face will be aligned during the examination and if not, an alert can be given to the proctor and further processing for non-aligned face can be terminated. FaceNet is a kind of deep neural network [44] applied to extract the features from a frame and it complies with inception_resnetv1 architecture as shown in Fig. 3. It is a 22-layered deep neural network. Face Net uses a 1 × 1 filter for dimensionality reduction. Along with these filters, 3 × 3, 5 × 5, and 7 × 7 filters are implemented parallelly to down sample the image by using max pooling. There is a chance of vanishing gradient problems during backpropagation. Therefore, two auxiliary outputs are tapped at middle layers and added to total loss by taking a weighted average. The facial points or fiducial points identified using FaceNet vary between 5 and 78. The minimum number of points to be identified is 5. In (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8), 68 facial points or fiducial points were identified. FaceNet compresses a face into a vector of 128 Dimensions by calculating the distance between these facial points. The 128-dimensional embedded vector contains the numbers which represent the important features of the face. The specific feature of FaceNet is its Triplet Loss Function as well as SoftMax activation with cross-entropy loss function, as represented in Eq.(11).

$$Cost\ Function = \sum_{i}^{N} Triplet\ Loss\ Function + L2\ Regularization \tag{11}$$

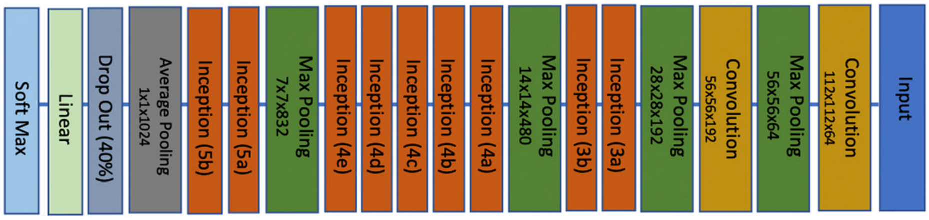

Fig. 3 Inception-Resnet V1 architecture for FaceNet

The embedded vector acts as a hash code of the face. This vector of 128 dimensions which is the distances between 68 specific points (landmarks) is used for identifying the spoofed face as explained below.

3.4 Face spoofing and multiple face detection

Face spoofing and multiple face detection is a crucial phase in an online proctoring system. In this phase, an examiner can understand whether the examinee is spoofing or any other person helping the examinee. In (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8), the spoofed face is not identified. All the faces identified in the frame are considered real and the remaining process is implemented. But there will be a great probability of spoofing the faces during an examination. Face spoofing is done by notifying the confidence level of the liveness of the face.

As shown in Fig. 4, the Face Detection module detects the face in the frame, if at least one face is aligned then it extracts the features from the aligned faces. The vector of 128 dimensions obtained from landmark localization which is unique for all faces is used for Liveness Model. The localized landmarks distance between the eye tips, distance between mouth tips, distance between eye tips and nose tip, distance between nose tip and mouth tip and the distance between mouth tips and chin from the Face Net are considered for the liveness detection algorithm. The procedure of extracting landmarks using Face net and detecting the liveness is described in the following algorithm.

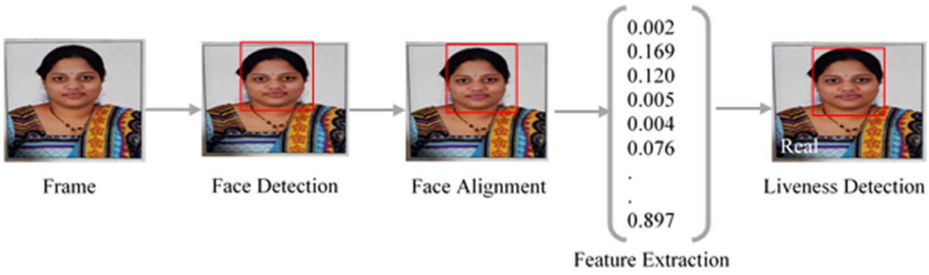

Fig. 4 Liveness detection. After detecting faces in the frame, face alignment is verified. Features are extracted for the aligned faces. The extracted features are used for liveness detection

Algorithm 5. Liveness Detection with Face Net

```
1: $A \leftarrow$ Aligned Face $Landmarks$
2: $f(A) \leftarrow Embedding(A)$ //Get Embedding vector using FaceNet
3: $NL \leftarrow No.of Layers$
4: $L \leftarrow \sum_i^n [\|f(A^a) - f(A^p\|_2^2 - \|f(A^a) - f(A^n)\|_2^2 + \alpha]$ //Apply Triplet Loss
   Function on Embedding vector
5: for $i = 0 to NL$ do
6:    for all $pixels\ in A$ do
7:       $g1(x, y) \leftarrow \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} w(dx, dy) * I1(x+dx, y+dy)$ //Convolution
   Layer in Liveness Model
8:       if then$g1(x, y) < 0$
9:          $g1(x, y) \leftarrow \alpha * g1(x, y)$ //Multiply every negative pixel with alpha
10:      end if
11:      perform 2 x 2 max pooling
12:   end for
13: end for
14: Add Flatten Layer
15: Add Dense Layer
16: $\sigma z \leftarrow e^{C_{1i}} / \sum_{j=1}^{K} e^{C_{1j}}$ //Soft max operation on input vector $C_{1i}$
17: if then$\sigma z < threshold$
18:    Fake Face
19: else
20:    Real Face
21: end if
```

As liveness detection is implemented to only aligned faces, false positives can be reduced. A sequential layered model is considered as base model on which liveness model is designed. The liveness model is a simple and shallow CNN with very few parameters to minimize the chances of overfitting small datasets. It exhibits VGGNet-esque qualities. Our CNN is very shallow with only a few learned filters. All the images loaded are resized into 32 × 32. All the pixel intensities are scaled to the range [0,1]. Two-layered sets of conv = > ReLU = > conv = > ReLU = > Pool are implemented along with batch normalization and dropout. Finally, fully connected and ReLU activated layers with a SoftMax are added. The complete summary of the model is shown in Fig. 5.

In this model, the ReLU activation function is implemented, and the pool size is taken as 2 × 2 all-time for Max Pooling 2D. Later, different activation functions are used for testing. The SoftMax classifier is considered which returns the probabilities.

$$\sigma\left(\vec{Z}\right)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{12}$$

If the probability returned by the liveness model is less than 0.5, the detected face is considered as fake, i.e., spoofed face else, the face is identified as a real face. An alert will be given to the proctor if a spoofed face is identified. If more than one real face is detected, our system gives an alert to the proctor that multiple faces are detected. To reduce the computational complexity, the head pose is estimated only for the faces identified as real as described below.

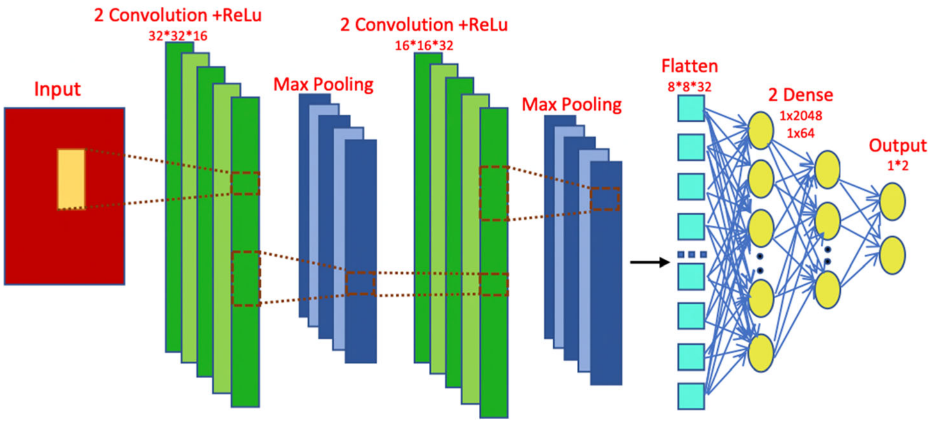

Fig. 5 Liveness net model summary

3.5 Head-pose estimation

Examinees may take help from others who are in the room or also they can also take help from other resources like laptops, systems placed in the room. This kind of misbehavior can be identified by estimating the head pose orientation. The head pose is estimated only for the aligned faces. If the examinee is seeing continuously other directions like left and right, with a certain angle, an alert is sent to the proctor. In [2] (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8), gaze tracking is implemented instead of head pose estimation. The eye-ball detection of the gaze-tracking does not give accurate results if the examinee is using spectacles. So, in our system, we are going with head-pose estimation using an affine rotation matrix. Step by step process involved in the head pose estimation is as follows:

1. Attentive net model is used to compute the landmarks such as Nose Tip, Chin, Left eye left corner, Right eye right corner, Left mouth corner, Right mouth corner.
2. Further, model points for 3D projection are also estimated. Camera Intrinsic parameters as below are also given as an input to the SolvePnp function. f_x and $f_{y,}$ are focal lengths that are expressed in pixel units and a principal unit ($c_{x,}$ c_y) is included in the matrix.

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

3. SolvePnp function is used to compute the Direct Linear Transform (DLT) solution followed by Levenberg-Marquardt optimization. DLT is used to solve a set of variables from a set of similarity relations like below:

$$x_i = Ay_i \ \ where, A = XY^T\left(YY^T\right)^{-1} \tag{14}$$

The Levenberg-Marquardt optimization algorithm starts with an initial guess $g_{0,}$ where g is adjusted by only δ exclusively for downward steps as follows:

$$\left(J^T J + \lambda I\right)\delta = J^T r \tag{15}$$

where J is Jacobian Matrix, λ and is Damping parameter and r is a Residual vector.

4. A 3 × 3 Rotation Matrix (R`) and 3 × 1 translation vector (t) are calculated is as follows

$$P_c = \begin{bmatrix} R^` & t \\ 0 & 1 \end{bmatrix} P_w \tag{16}$$

$$\begin{bmatrix} R^` & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

5. Plot the translation and rotation vector values in the view of the nose end tip.

From the above procedure, the head movement towards left and right can be identified. If the head is turned either to left or right, then an alert is given to the proctor. The above steps show that an alert will be given to the proctor either if face spoofing is identified for the aligned faces or if the head of real face is rotated towards left or right.

4 Experimental results and discussions

In this chapter, the authors discussed dataset statistics, evaluation procedures of the proposed method, discussion on experimental results, and comparative analysis with other methods. The proposed system is implemented on Keras with TensorFlow as the backend. In this work, each frame is captured video with a minimum size of the frame as 80*80 and then it will be given as an input to the system.

4.1 About dataset

The performance of the proposed model is evaluated on a dataset that comprises 200 videos with various combinations of malpractice behaviors. Attentive-Net is pre-trained on the CASIA-Web Face dataset which consists of 453,453 images to assess the liveliness of the examinees. Further, Liveliness is evaluated on a custom dataset with 200 images, where 100 images are fake and the remaining 100 images are real. Our proposed model is also evaluated on Computational Intelligence and Photography Lab (CIPL) dataset, which has 960 fake and

1081 real face samples. Figure 6 shows the sample fake and real images. The performance of the proposed method is evaluated by considering 75% samples for model building and the remaining 25% of samples are used to assess the model performance.

4.2 System evaluation

The input images extracted from frames are resized to 32*32 and are inputted to Attentive-Net. Firstly, acquired face image was duly aligned and then mean positions of face for five landmarks considered to determine the angle of rotation are x = [0.224152, 0.75610125, 0.490127, 0.254149, 0.726104] and y = [0.2119465, 0.2119465, 0.628106, 0.780233, 0.780233], respectively. Further, the affine matrix and the face position (left, right, center) details are computed. If at least one sample image is registered as aligned, then the FaceNet Algorithm is implemented to extract 128 dimensional features along with the computation of liveliness detection and estimating head pose. The performance of the proposed Attentive system is evaluated in both alignments: with and without face alignment. The experimentation shows that the feature extraction without face alignment has a high ratio of false positives. If the alignment process showing the examinee's face is not captured, immediately proctor is sent an alert instead of feature extraction. If the examinee is writing the examination, through the camera we can find at least one face image, and a further aligned face can be identified. The Attentive system yielded a 14.67% of false-positive ratio (FPR) without face alignment which was reduced to 11.65% with face alignment. Similarly, False Negative Ratio (FNR) without face alignment is 15.45% and with alignment, it was reduced to 11.29%.

During model building learning rates are varied from 1e-4 to 5e-4 in the proposed work and it is observed that the model performance is stable at 1e-4 learning rate, batch size as 10 and number of epochs as 60. But decay in performance is observed at the learning rates of 2e-4, 3e-4, 4e-4, and 5e-4. Table 2 shows the performance of the proposed model with Adam as an optimizer with varied learning rates and batch sizes.

The performance of the proposed method is evaluated with various activation functions like Sigmoid, Hyperbolic Tangent, ReLu, and Leaky ReLu in the face detection module. Table 3 shows the performance of the proposed model in terms of Accuracy, Precision, Recall, and Loss; it also shows that the Sigmoid achieved lower accuracy when compared with SoftMax in the binary classification. The results show that the performance of ReLu and Leaky ReLu are almost the same. The experimentation shows that ReLu addresses well the gradient vanishing problem and is comparatively less computationally expensive. During the training process, it is observed that the training time corresponding to each face detection increases with increase in number of training samples. Conversely, the no. of training sample increases, the test time corresponding to each face detection decreases.

Fig. 6 Sample real and fake images of CIPL and custom datasets

Table 2 Accuracy analysis of various hyper parameters of CIPL dataset

Learning rate	Batch size	Accuracy	Precision	Recall
1e-10	8	0.63	0.65	0.74
1e-8	8	0.66	0.66	0.73
1e-8	10	0.71	0.74	0.82
1e-6	8	0.72	0.73	0.89
1e-6	10	0.75	0.78	0.9
1e-4	8	0.81	0.86	0.96
1e-4	10	0.86	0.9	0.97
1e-2	8	0.82	0.82	0.89
1e-2	10	0.87	0.88	0.89
1e-2	5	0.85	0.89	0.96
2e-4	8	0.73	0.76	0.86
3e-4	8	0.74	0.79	0.98
4e-4	8	0.76	0.81	0.83
5e-4	8	0.72	0.75	0.88

Table 3 Evaluation of model based on activation functions on CIPL dataset

Activation function	Accuracy	Precision	Recall	Test loss
Sigmoid	0.71	0.75	0.82	0.09
SoftMax	0.80	0.86	0.95	0.05
Hyperbolic Tangent	0.74	0.80	0.94	0.09
ReLu	0.81	0.86	0.96	0.05
Leaky ReLu	0.79	0.85	0.96	0.08

Further, Figs. 7 and 8 shows the performance of the Attentive system on how it identifies the malpractice behavior of the examinee with the level of confidence. For each captured face image, face alignment is carried, and then liveliness is computed continuously for 30 frames.

The SolvePnp equation is used to compute the rotation and translation vectors of each live face sample using the nose endpoint as illustrated in Fig. 9. To identify the left and right positions of the head image, the translation and rotation vectors were set as (40, 20) and (40, −20).

Our system also detected various objects like a mobile, book, laptop as shown in Fig. 10. An alert will be generated by our system if any of these objects are detected. Figure 11 shows the working of head-pose estimation. Our model checks for the aligned face and continues the further steps. Head pose is estimated only when at least one face is aligned. As the face is not aligned in the second part of Fig. 11, the Head pose is not estimated. In the remaining parts of

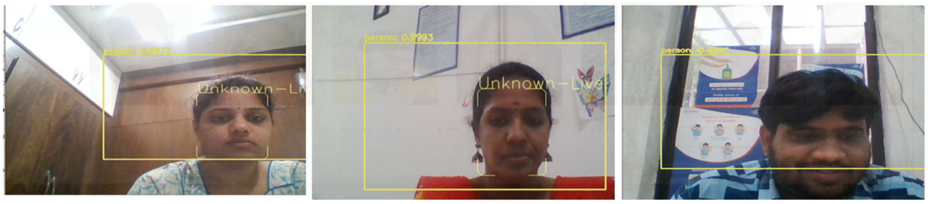

Fig. 7 Face detection with confidence

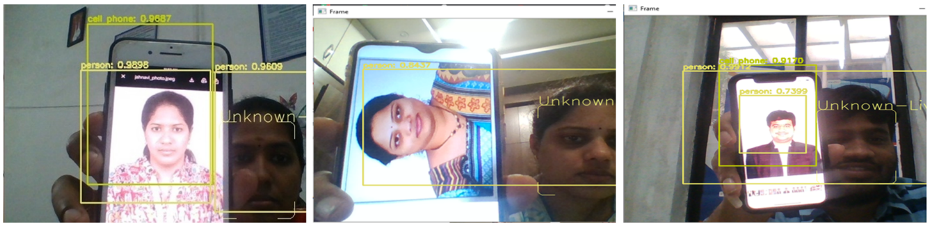

Fig. 8 Liveliness detection

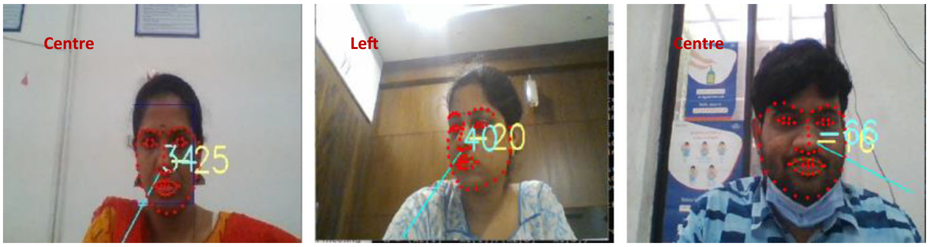

Fig. 9 Head pose estimation

the Figure, it is observed that the head pose towards the right, left and center can be detected by using the vector values displayed at the nose tip.

4.3 Comparison of different proctoring approaches

The performance of the proposed method is compared with the Vardhaman AI Proctoring system (https://towardsdatascience.com/automating-online-proctoring-using-ai-e429086743c8), Automated Online Proctoring system [2] as in Table 4. It is evident that the precision value of the proposed method is 0.81 and it is superior to the other two methods as shown in Fig. 12. The Vardhaman AI_Proctoring is limited in the recognition of live faces efficiently in the dim light when compared to our proposed approach. These existing systems do not verify the face alignment during the verification of face spoofing. Considering our Attentive system as a Computer Vision Problem, our system is compared with existing system [2] in different scenarios, Attentive system performs well in face detection over Viola-Jones. Also Attentive-Net estimates head-pose more accurately using Direct Linear Transform over

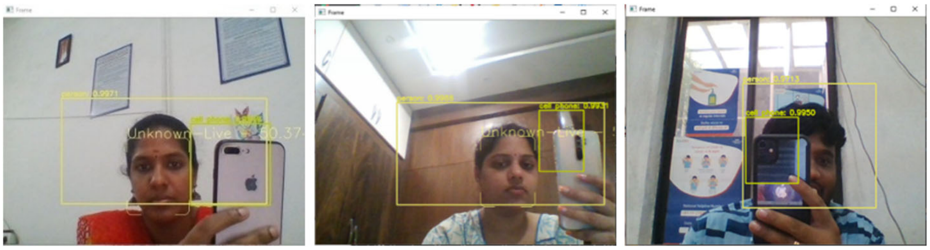

Fig. 10 Mobile/book/laptop detection

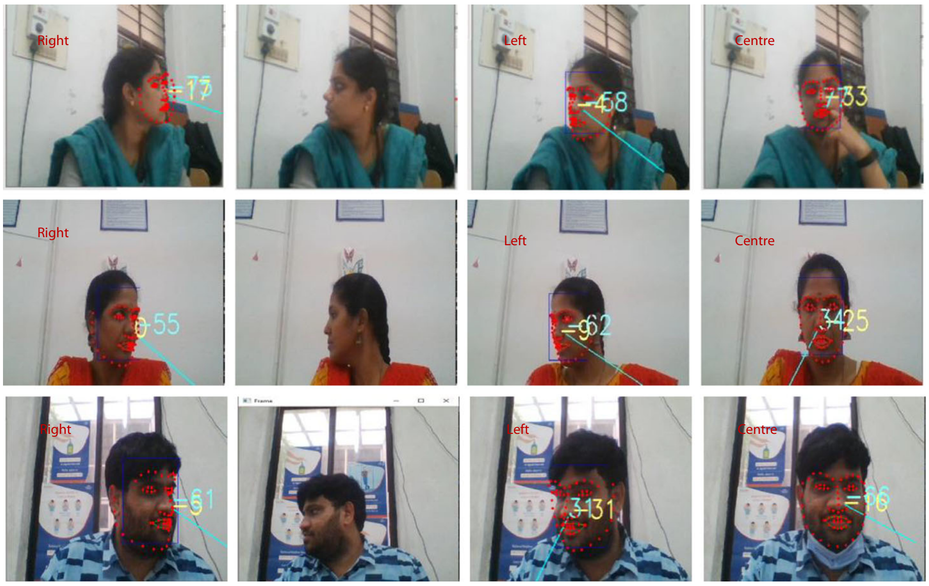

Fig. 11 (From Left to right) Head Right, not aligned face, Head Left, Head straight

Lucas-Kanade method used in [2]. Attentive system uses YOLOv3 model to detect various gadgets and books which can be used for malpractice.

The proposed research is also evaluated on a dataset with video samples having 15% of mischievous behavior when compared with cheating activities and the remaining section have regular activities. The mischievous actions include seeking help from others, usage of mobile phone, substituting the examinee, and moving away from the camera, etc. The performance of the proposed model is evaluated in two different scenarios like segment-based and instance-based metrics [2]. Both these metrics are determined using True Detection Rate (TDR) and False Alarm Rate (FAR). The mischievous behavior of the examinee is assessed using the following formulae:

$$\text{Instance}-\text{based TDR} = \frac{\Sigma i\#\text{detected cheating instances}}{\Sigma i\#\text{cheating instances}}$$

$$\text{Segment based TDR} = \frac{\Sigma i\#\text{detected cheating segments}}{\Sigma i\#\text{round truth cheating segments}}$$

Table 4 F1 score for existing systems and proposed system on the custom dataset (100 videos)

Method	Accuracy	Precision	Recall	F1 score
Vardhaman AI_Proctoring	0.66	0.65	0.86	0.65
Automated Online Proctoring	0.64	0.60	0.85	0.64
Attentive System (Proposed)	0.87	0.81	0.96	0.83

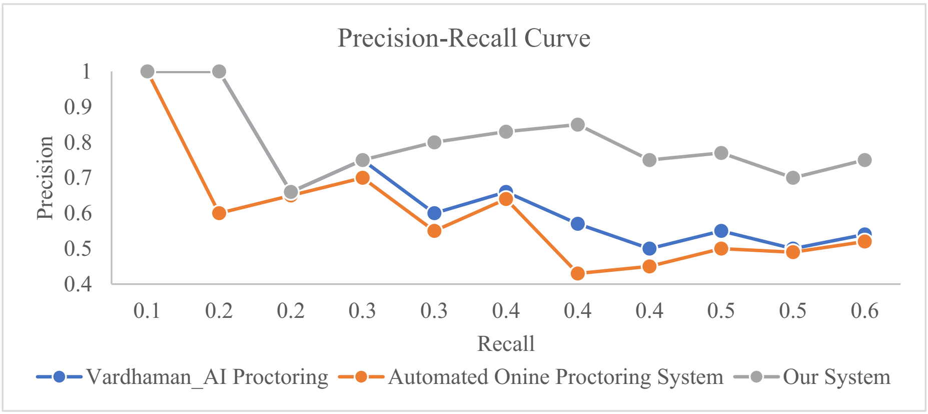

Fig. 12 Precision-recall curve over custom dataset (100 videos) for proposed attentive system and existing systems

$$\text{Instance-based FAR} = \frac{\Sigma i\#\text{false cheat instances}}{\Sigma i\#\text{of cheat-free minutes}}$$

$$\text{Segment based FAR} = \frac{\Sigma i\#\text{false cheat segments}}{\Sigma i\#\text{of cheat-free segments}}$$

By using the same process, two existing methodologies are compared with our system as shown in Fig. 13. Observably, when compared to the Automated Online Proctoring system [2] and Vardaman AI Proctoring, our system performs better both in the segment-based and instance-based metrics. As Vardaman AI Proctoring does not check for the face alignment, FAR are high. These observations show that our system gives FAR less compared to these systems and the True detection rate is high compared to other systems.

The performance of the proposed method is compared with Vardhaman AI Proctoring with the combination of various optimizers and activation functions. Table 5 shows further results.

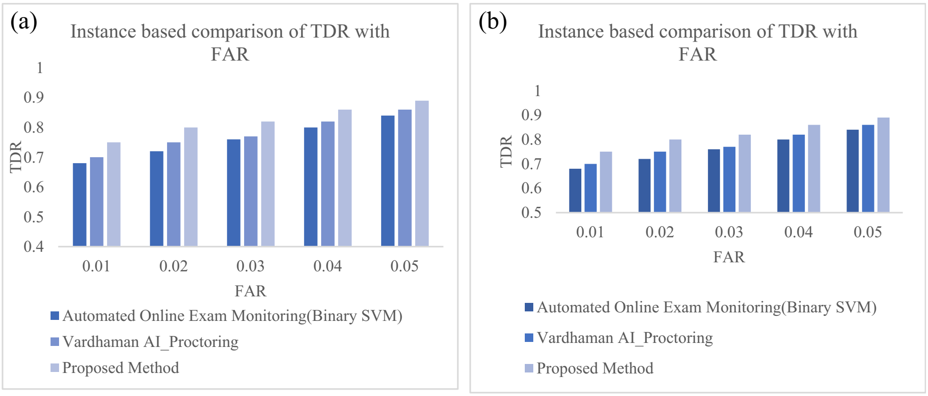

Fig. 13 **a** Instance-based comparison **b** Segmentation-based comparison of FAR with TDR.

Table 5 Performance comparison of the proposed method with Vardhaman AI in tune with various hyperparameters on the CIPL dataset

Model	Loss Function	Opt=Adam, LR= 0.0001			Opt=Adam delta, LR= 0.0001			Opt=Adamax, LR= 0.0001			Opt=SGD, LR= 0.0001			Opt=Adagrad, LR= 0.0001		
		A	P	R	A	P	R	A	P	R	A	P	R	A	P	R
Vardhaman_AI	Binary Cross Entropy	0.66	0.65	0.86	0.62	0.65	0.73	0.63	0.63	0.72	0.63	0.64	0.75	0.68	0.65	0.87
	Poisson	0.75	0.8	0.87	0.69	0.71	0.73	0.7	0.72	0.85	0.6	0.63	0.84	0.61	0.62	0.64.
	Cosine Proximity	0.66	0.68	0.69	0.62	0.63	0.62	0.63	0.64	0.77	0.71	0.73	0.82	0.6	0.62	0.62
	Kullback Leibler Divergence	0.66	0.66	0.7	0.62	0.63	0.69	0.61	0.62	0.72	0.69	0.6	0.7	0.6	0.62	0.77
	Hinge	0.63	0.68	0.86	0.63	0.66	0.72	0.61	0.62	0.61	0.58	0.59	0.68	0.61	0.63	0.74
	Mean squared logarithmic error	0.62	0.64	0.79	0.6	0.62	0.82	0.61	0.64	0.79	0.6	0.61	0.7	0.6	0.62	0.79
Proposed Method	Binary Cross Entropy	0.81	0.86	0.96	0.8	0.82	0.83	0.83	0.83	0.92	0.83	0.84	0.85	0.82	0.85	0.89
	Poisson	0.85	0.9	0.97	0.79	0.81	0.83	0.8	0.82	0.85	0.8	0.83	0.94	0.81	0.82	0.84.
	Cosine Proximity	0.86	0.88	0.89	0.82	0.83	0.82	0.83	0.84	0.87	0.81	0.83	0.84	0.8	0.82	0.82
	Kullback Leibler Divergence	0.86	0.86	0.9	0.82	0.83	0.89	0.81	0.82	0.82	0.79	0.8	0.9	0.8	0.82	0.87
	Hinge	0.83	0.84	0.86	0.83	0.86	0.92	0.81	0.82	0.81	0.78	0.79	0.98	0.81	0.83	0.84
	Mean squared logarithmic error	0.82	0.84	0.89	0.8	0.82	0.92	0.81	0.84	0.89	0.8	0.81	0.9	0.8	0.82	0.89

It is observed that the loss curve fits better with the Adam Optimizer when compared with other loss functions such as cosine, hinge, and mean squared logarithmic error. The proposed model maintains 0.86 accuracies for the cosine proximity and Kullback Leibler Divergence loss functions. Also, the loss function affects the learning rate. The characteristics of various optimizers are as follows; Adam needs less memory, Ada max is based on infinity norm, Adagrad depends on parameter specific learning rate and Adadelta is a stochastic gradient descent method. The existing Vardhaman AI proctoring system uses Adam delta optimizer, with the batch size 10 and learning rate 1e-4. The Vardhaman AI proctoring system is a ResNet50 based model which failed to extract embeddings for several faces from images with high Gaussian blur. It is also not able to vectorize poor quality images which can be done by the face net.

The average processing time of different modules like Face Detection, Head Pose Estimation, Phone Detection, Face Spoofing and Multiple person detection are compared and the results are presented in the Table 6. Apparently, introducing the face alignment module into our Attentive system saves the time of Head Pose Estimation significantly.

From the above discussions, it is observed that our Attentive system superior to existing methods in face detection, pose estimation and phone detection which can be embedded to any platform/software for online proctoring with additional services or features such as User Verification and Active Window Detection to avoid some more malpractice.

4.4 Discussion on challenges

The face detection module identifies an individual only if the complete face is bounded in the frame. In the face detection module, as the partial or occluded faces are not properly identified, an alert cannot be sent to the proctor. In some of the cases, those partial faces could be of people other than the examinee who may be extending their help to the examinee but are not identified. Besides, the examinee may spoof his/her face. The liveness detection model identifies the face spoofing and also alerts the proctor about the mischievous behavior of the examinee. We are training the model with images of different poses and expressions. If the faces with neutral expressions are categorized into spoofed images and faces with some expressions are categorized into live images, then the examinee who is with neutral expression while writing examination is also identified as spoofed image. The examinee's face is in the same position continuously for a long duration is identified as a spoofed image as there is no facial movement or expression. The accuracy in these terms can be improved by categorizing the datasets with more combinations. Along with liveliness detection, face verification also needs to be done. If the examinee's face is partially covered with a mask or hand then the pose

Table 6 Average processing time (ms)

Model	Face Detection module	Face Alignment	Head Pose Estimation	Phone Detection	Face Spoofing	Multiple person detection
Vardhaman AI_Proctoring	110	NA	125	238	NA	112
Automated Online Proctoring	120	NA	125	238	NA	122
Attentive System	98	110	109	236	120	100

is not estimated properly, as all facial features are not extracted accurately. The marking of the pose angles will also be disturbed due to missing facial features.

The liveness detection and estimation of the head pose may not be sufficient as the examinee may seek help from other persons or devices located in the room. The examinee may not rotate his/her head but moves eyes to seek help. These types of malpractice behaviors can be detected by using eye-gaze tracking. Without changes in the position of the head and eyes, the examinee can also speak to others to seek help that is not identified. Our Attentive system is limited to the recognition of some mischievous behaviors of the examinees. The proposed online proctoring system works efficiently in the place of a human proctor. Our proposed proctoring system detects the faces from live video, identifies the face spoofing, detects cell phones, laptops, books and estimates the head pose. Our system helps proctors to identify the malpractices carried by the examinees. The experimental results prove that the proposed system better performs when compared to the existing Proctoring system. The performance of the proposed system can be improvised in identifying if the examinee is speaking to someone else during an examination.

5 Conclusion

The main objective of this paper is to develop a well-rounded automation system that is capable of helping the proctor to monitor the students attending an online examination. Out of the several proposed features of the system, our paper has developed the ability to do multiple-person detection, face spoofing, and head pose estimation. While these form important milestones, the ultimate objective is to develop the automation system. Our system reduced the hardware and labor cost as the process is implemented only for aligned faces. Our model is trained in such a way to detect faces from frames with poor quality. The experimental results demonstrated that our automation system performs better with a precision value of 0.81 and recall of 0.91. Various cheating behaviors are also identified with 0.86 accuracy, where accuracy for existing systems is around 0.64. The model also performs better with various Hyperparameters. At the moment, the scope of the project is restricted to video Processing. It is hoped that the system can also be implemented using the examinee identity verifications, mouth opening detection mechanisms, and other audio capturing tools.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

1. Albiero V, Hassner T, Pang G, Yin X, Chen X (2020) img2pose: Face alignment and detection via 6DoF, face pose estimation. https://doi.org/10.48550/arXiv.2012.07791
2. Atoum Y, Chen L, Liu AX, Hsu SDH, Liu X (2017) Automated online exam proctoring. IEEE Trans Multimed 19(7). https://doi.org/10.1109/TMM.2017.2656064

3. Borghi G, Fabbri M, Vezzani R, Calderara S, Cucchiara R (2020) Face-from-depth for head pose estimation on depth images. IEEE Trans Pattern Anal Mach Intell 42(3). https://doi.org/10.1109/TPAMI.2018.2885472
4. Bu W, Xiao J, Zhou C, Yang M, Peng C (2017) A cascade framework for masked face detection. https://doi.org/10.1109/ICCIS.2017.8274819
5. Bulat A, Tzimiropoulos G (2016) Two-stage convolutional part Heatmap regression for the 1st 3D face alignment in the wild (3DFAW) challenge. https://doi.org/10.1007/978-3-319-48881-3_43
6. Chang F-J, Tran AT, Hassner T, Masi I, Nevatia R, Medioni G (2019) Deep, landmark-free FAME: face alignment, modeling, and expression estimation. Int J Comput Vision 127(6–7). https://doi.org/10.1007/s11263-019-01151-x
7. Chen D, Ren S, Wei Y, Cao X, Sun J (2014) Joint cascade face detection and alignment. https://doi.org/10.1007/978-3-319-10599-4_8
8. Chen W, Zhou Q, Hu H (2019) Face alignment by discriminative feature learning. https://doi.org/10.1109/ICIP.2019.8803112
9. Chen H, Hu G, Lei Z, Chen Y, Robertson NM, Li SZ (2020) Attention-based two-stream convolutional networks for face spoofing detection. IEEE Trans Inf Forensics Secur 15. https://doi.org/10.1109/TIFS.2019.2922241
10. Clarke NL, Dowland P, Furnell SM (2013) e-Invigilator: A biometric-based supervision system for e-Assessments. In: International conference on information society (i-Society 2013), pp 238–242.
11. Cluskey GR Jr, Ehlen CR, Raiborn MH (2011) Thwarting online exam cheating without proctor supervision. J Acad Bus Ethics 4(1):1–7
12. Cuimei L, Zhiliang Q, Nan J, Jianhua W (2017) Human face detection algorithm via Haar cascade classifier combined with three additional classifiers.. https://doi.org/10.1109/ICEMI.2017.8265863
13. de Souza GB, da Silva Santos DF, Pires RG, Marana AN, Papa JP (2017) Deep texture features for robust face spoofing detection. IEEE Trans Circ Syst II Exp Briefs 64(12). https://doi.org/10.1109/TCSII.2017.2764460
14. Deshpande NT, Ravishankar S (2017) Face detection and recognition using Viola-Jones algorithm and fusion of PCA and ANN. Adv Comput Sci Technol 10:1173–1189
15. Dong S, Wang P, Abbas K (2021) A survey on deep learning and its applications. Comput Sci Rev 40: 100379. https://doi.org/10.1016/j.cosrev.2021.100379
16. Girshick R, Donahue J, Darrell T, Malik J (2016) Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans Pattern Anal Mach Intell 38(1). https://doi.org/10.1109/TPAMI.2015.2437384
17. Golnaz G, Tsung-Yi L, Le Quoc V (2019) NAS-FPN: learning scalable feature pyramid architecture for object detection. In: CVPR, pp 7036–7045. https://doi.org/10.48550/arXiv.1904.07392
18. Gunther M et al (2017) Unconstrained face detection and open-set face recognition challenge. https://doi.org/10.1109/BTAS.2017.8272759
19. Guo P, Yu H-F, Yao Q (2008) The research and application of online examination and monitoring system. https://doi.org/10.1109/ITME.2008.4743914
20. Gupta P, Saxena N, Sharma M, Tripathi J (2018) Deep neural network for human face recognition. Int J Eng Manuf 8(1). https://doi.org/10.5815/ijem.2018.01.06
21. Gupta A, Thakkar K, Gandhi V, Narayanan PJ (2019) Nose, eyes and ears: head pose estimation by locating facial Keypoints. https://doi.org/10.1109/ICASSP.2019.8683503
22. Han H, Jiayuan G, Zheng Z, Jifeng D, Yichen W (2018) Relation networks for object detection. In: CVPR, pp 3588–3597.https://doi.org/10.48550/arXiv.1711.11575
23. Hsu H-W, Wu T-Y, Wan S, Wong WH, Lee C-Y (2019) QuatNet: quaternion-based head pose estimation with multiregression loss. IEEE Trans Multimed 21(4). https://doi.org/10.1109/TMM.2018.2866770
24. Jia J, He Y (2021) The design, implementation and pilot application of an intelligent online proctoring system for online exams. Interact Technol Smart Educ. vol. ahead-of-print, no. ahead-of-print. https://doi.org/10.1108/ITSE-12-2020-0246
25. Jung IY, Yeom HY (2009) Enhanced security for online exams using group cryptography. IEEE Trans Educ 52(3) https://doi.org/10.1109/TE.2008.928909
26. Khan M, Chakraborty S, Astya R, Khepra S (2019) Face detection and recognition using OpenCV. https://doi.org/10.1109/ICCCIS48478.2019.8974493
27. Li X, Chang K, Yuan Y, Hauptmann A (2015) Massive open online proctor. https://doi.org/10.1145/2675133.2675245
28. Li L, Feng X, Xia Z, Jiang X, Hadid A (2018) Face spoofing detection with local binary pattern network. J Vis Commun Image Represent 54. https://doi.org/10.1016/j.jvcir.2018.05.009
29. Li C, Wang R, Li J, Fei L (2020) Face detection based on YOLOv3. https://doi.org/10.1007/978-981-13-9406-5_34

30. Maatta J, Hadid A, Pietikainen M (2011) Face spoofing detection from single images using micro-texture analysis. https://doi.org/10.1109/IJCB.2011.6117510
31. Milone AS, Cortese AM, Balestrieri RL, Pittenger AL (2017) The impact of proctored online exams on the educational experience. Currents in Pharmacy Teaching and Learning 9(1). https://doi.org/10.1016/j.cptl.2016.08.037
32. Mingxing T, Ruoming P, Le Quoc V (2020) EfficientDet: Scalable and Efficient Object Detection. In: CVPR, pp 10781–10790. https://doi.org/10.48550/arXiv.1911.09070
33. Motwani S, Nagpal C, Motwani M, Nagdev N, Yeole A (2021) AI-based proctoring system for online tests. SSRN Electron J. https://doi.org/10.2139/ssrn.3866446
34. Patacchiola M, Cangelosi A (2017) Head pose estimation in the wild using Convolutional Neural Networks and adaptive gradient methods. Pattern Recognit 71. https://doi.org/10.1016/j.patcog.2017.06.009
35. Potluri T, Gnaneswara Rao N (2019) Content based video retrieval using SURF, BRISK and HARRIS features for query-by-image. 1035. https://doi.org/10.1007/978-981-13-9181-1_24
36. Potluri T, Nitta G (2016) Content based video retrieval using dominant color of the truncated blocks of frame. J Theor Appl Inf Technol 85(2)
37. Potluri T, Sravani T, Ramakrishna B, Nitta GR (2017) Content-based video retrieval using dominant color and shape feature. 507. https://doi.org/10.1007/978-981-10-2471-9_36
38. Prathish S, S AN, Bijlani K (2016) An intelligent system for online exam monitoring. https://doi.org/10.1109/INFOSCI.2016.7845315
39. Priadana A, Habibi M (2019) Face detection using Haar cascades to filter selfie face image on Instagra. https://doi.org/10.1109/ICAIIT.2019.8834526
40. ProctorU:Real People Real Proctor (n.d.) http://www.proctoru.com
41. Qin X, Zhou Y, He Z, Wang Y, Tang Z (2017) A faster R-CNN based method for comic characters face detection. https://doi.org/10.1109/ICDAR.2017.178
42. Rosen WA, Carr ME (2013) An autonomous articulating desktop robot for proctoring remote online examinations. https://doi.org/10.1109/FIE.2013.6685172
43. Ruiz N, Chong E, Rehg JM (2018) Fine-grained head pose estimation without Keypoints. https://doi.org/10.1109/CVPRW.2018.00281
44. Schroff F, Kalenichenko D, Philbin J (2015) FaceNet: a unified embedding for face recognition and clustering. https://doi.org/10.1109/CVPR.2015.7298682
45. Shao Z, Zhu H, Tan X, Hao Y, Ma L (2020) Deep multi-center learning for face alignment. Neurocomputing 396. https://doi.org/10.1016/j.neucom.2018.11.108
46. Song X, Zhao X, Fang L, Lin T (2019) Discriminative representation combinations for accurate face spoofing detection. Pattern Recognit 85. https://doi.org/10.1016/j.patcog.2018.08.019
47. Sun X, Wu P, Hoi SCH (2018) Face detection using deep learning: an improved faster RCNN approach. Neurocomputing 299. https://doi.org/10.1016/j.neucom.2018.03.030
48. Tsai Y-H, Lee Y-C, Ding J-J, Chang RY, Hsu M-C (2018) Robust in-plane and out-of-plane face detection algorithm using frontal face detector and symmetry extension. Image Vis Comput 78. https://doi.org/10.1016/j.imavis.2018.07.003
49. Vaishali, Singh S (2019) Real-time object detection system using Caffe model. Int Res J Eng Technol 06(05):5727–5732
50. Voss C, Haber NJ (2018) Systems and methods for detection of behavior correlated with outside distractions in examinations
51. Wahid A, Sengoku Y, Mambo M (2015) Toward constructing a secure online examination system. https://doi.org/10.1145/2701126.2701203
52. Wang Y, Liang W, Shen J, Jia Y, Yu L-F (2019) A deep Coarse-to-Fine network for head pose estimation from synthetic data. Pattern Recognit 94. https://doi.org/10.1016/j.patcog.2019.05.026
53. Wayne W, Chen Q, Shuo Y, Quan W, Yici C, Qiang Z (2018) Look at boundary: A boundary-aware face alignment algorithm. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2129–2138
54. Wu S, Kan M, Shan S, Chen X (2019) Hierarchical attention for part-aware face detection. Int J Comput Vis 127:6–7. https://doi.org/10.1007/s11263-019-01157-5
55. Xiang J, Zhu G (2017) Joint face detection and facial expression recognition with MTCNN. https://doi.org/10.1109/ICISCE.2017.95
56. Xu X, Kakadiaris IA (2017) Joint head pose estimation and face alignment framework using global and local CNN features. https://doi.org/10.1109/FG.2017.81
57. Xu Y, Jung C, Chang Y (2022) Head pose estimation using deep neural networks and 3D point clouds. Pattern Recognit 121. https://doi.org/10.1016/j.patcog.2021.108210
58. Yang T-Y, Chen Y-T, Lin Y-Y, Chuang Y-Y (2019) FSA-net: learning fine-grained structure aggregation for head pose estimation from a single image. https://doi.org/10.1109/CVPR.2019.00118

59. Zhang H, Wang X, Zhu J, Kuo C-CJ (2019) Fast face detection on mobile devices by leveraging global and local facial characteristics. Signal Process Image Commun 78 https://doi.org/10.1016/j.image.2019.05.016
60. Zhang Y, Tino P, Leonardis A, Tang K (2021) A survey on neural network interpretability. IEEE Trans Emerg Top Comput Intell 5(5):726–742. https://doi.org/10.1109/TETCI.2021.3100641
61. Zhen X, Yu M, Xiao Z, Zhang L, Shao L (2020) Heterogenous output regression network for direct face alignment. Pattern Recognition 105. https://doi.org/10.1016/j.patcog.2020.107311