INTELLIGENT ONLINE PROCTORING SYSTEM

Apar Garg, Gopan Ravikumar Girija

Institute of Systems Science, National University of Singapore, Singapore 119615

ABSTRACT

The Covid-19 pandemic has been one of the defining events in recent history. It has affected millions of lives and has had an impact on every sector of civilization. No matter the domain, the pandemic has forced it to implement radical and innovative reforms. Education and academia has been identified as one such sector that has been impacted most adversely due to the pandemic. Disrupting the age-old classroom setup, the pandemic has forced educational institutions like schools and universities to implement 'online classes'. However, the evaluation aspect of education remains to be desired. Many automatic online exam proctoring systems have been proposed for online examinations during this Covid-19 pandemic but they have certain limitations like fewer and inaccurate functionalities. In this paper, we build a smart exam monitoring system, which addresses many of the problems with past systems, to help institutions avoid malpractices during the exams.

***Index Terms*—** Online proctoring system, Education, Authentication, Abnormal behavior detection

1. INTRODUCTION

Exams are a critical component of any educational system. The effect of COVID-19 on education has caused many schools and universities to switch their mode of exams from in-person exams to the online mode to adhere to public safety regulations. Educational Testing Service (ETS), the non-profit educational organization which offers standardized tests including GRE and GMAT, is also allowing examinees to give exams from home where they will be monitored by a proctor for the whole duration of the exam. However, after multiple exams being conducted online, it is observed that examinees have scored remarkably high in tests due to a lack of the number of proctors.

Many automatic online exam proctoring (OEP) systems have been proposed to handle this problem. But they mainly focus on the authentication of examinees by performing tasks such as fingerprint authentication, face verification, and voice recognition.

However, the above methods ignore the abnormal behavior during the examinations. Normal examinees face the display screen to answer the questions. The abnormal examination behavior usually manifests in the abnormal changes in the head posture and eye movement and the continuous opening and closing of the mouth.

In this paper, we aim to build an automatic online exam proctoring system that provides improved authentication and abnormal behavior monitoring of examinees in the online examination based on image information. The input to our system is a real-time video stream. The video stream is analyzed frame-by-frame and alarms are generated whenever pre-defined triggers are encountered. We improve the authentication process by including a face spoofing feature. Through the use of head pose estimation, eye tracking, mouth movement analysis, and the combination of certain decision rules, the monitoring of abnormal behavior such as turning heads and eyes and speaking during the online examination is completed.

2. LITERATURE REVIEW

2.1. Existing Proctoring Systems

S.Prathish et al. [1] used the model-based head pose estimation method and the audio-based detection method to complete the test abnormal behavior detection. However, the accuracy rate of the head pose estimation of this method is not high enough, and the use of a microphone to collect sound can infringe the relevant privacy of examinees. Moreover, the abnormal behavior detection process does not consider eye tracking and mouth movement analysis.

In [2], the authors proposed a multimedia analytics system for online exam proctoring. With the captured videos and audio, they extract low-level features from six basic components: text detection, user verification, active window detection, speech detection, gaze estimation, and phone detection. These features are then processed in a temporal window to acquire high-level features and then used for cheat detection. However, the system is not feasible as it requires the examinee to have a wearcam. Moreover, the solution does not have a face spoofing feature for user authentication.

Hu et al. [3] proposed a system that uses an image-based head pose estimation model and mouth movement analysis to discriminate the abnormal behavior of the examinee during

the online examination. However, the system does not take eye-tracking functionality into consideration for analyzing the abnormal behavior of the examinee.

2.2. Head Pose Estimation

One of the common approaches to estimate head pose is using landmark points [4, 5, 6]. In this approach along with detected 2D landmarks, an average 3D mean mask and the intrinsic camera parameters are required to calculate the 3D head pose angle. Using this information, the extrinsic parameter of the camera is calculated which contains the information about 3D rotation and translation of face from the center of the camera. This approach has a few drawbacks. Firstly, the accuracy of this method heavily depends on the accuracy of the landmark model. The landmark model usually fails for large head pose angles since half of the features in the face are invisible. In such scenarios, the accuracy of head pose estimation will also drop. In addition to that, a mean 3D mask is used to perform the 3D to 2D alignment, and this will also introduce errors in head pose calculation. Since its accuracy drops when the examinee gives a large head pose, this approach is not suitable for our online proctoring system.

Another common approach is using deep learning-based classification methods. Some of the state-of-the-art models in this approach are Hopenet [7] and WHENet [8]. Hopenet uses Resnet50 as its backbone feature extractor, followed by a classifier that classifies each head pose angle. The network is trained using a combination of both classification and regression loss functions. WHENet also follows a similar approach but uses EfficientNet as a backbone feature extractor. Even though the two models mentioned here give a very accurate performance, they are not usable in our online proctoring system because of their computational complexity. We require a model that can achieve real-time performance with sufficient accuracy to identify when an examinee is looking away from the screen.

Table 1. Subset of functionalities in our system

Category	Functionality
Authentication	Face Verification Face Spoofing
Abnormal Behavior Detection	Image-based Head Pose Estimation Eye Tracking Mouth Movement Analysis

As per our knowledge, there is no previous work that has integrated all Authentication and Abnormal Behavior Detection features provided in Table 1.

3. DATASET

For training and validation of the head pose module, we used the Pandora dataset [9]. The dataset contains 100 annotated sequences collected from 10 male and 10 female subjects. Each subject has been recorded 5 times. For each subject, two sequences are performed with constrained movements, changing the yaw, pitch, and roll angles separately. Three additional sequences are completely unconstrained. The overall size of the dataset was around 130k. Example images from the Pandora dataset are shown in Fig. 1.

Fig. 1. Example images from Pandora dataset.

For testing the head pose module, we used the BIWI [10] benchmark dataset. It contains 15k frames, with RGB (640 × 480) and depth maps (640 × 480). 20 subjects have been involved in the recordings: 4 of them were recorded twice, for a total of 24 sequences. The ground truth of yaw, pitch, and roll angles is reported together with the head center and the calibration matrix. Example images from the BIWI dataset are shown in Fig. 2.

Fig. 2. Example images from BIWI dataset.

To improve the diversity of the dataset and prevent the model from overfitting we performed a series of color and geometrical augmentation techniques like horizontal flip, random zooming, adding random noise (Gaussian, salt, and pepper), color jitter (random brightness variations, random contrast variations, random saturation variations), and random image translations.

4. PROPOSED SYSTEM

4.1. System Architecture

This work proposes a system that uses a webcam to monitor examinees during the online examination. The system architecture is shown in Fig. 3. After the camera captures the frame, banned item detection and person detection are performed. If one person is detected in the frame, then face detection is performed. The detected face is input to face spoofing, face verification, face landmark detection, and head pose estimation models. The detected landmarks from the landmark detection model are input to mouth movement analysis and eye-tracking. We analyze the output from all models to conclude whether the examinee is cheating or not. Table 2 contains all the functionalities present in our proposed system.

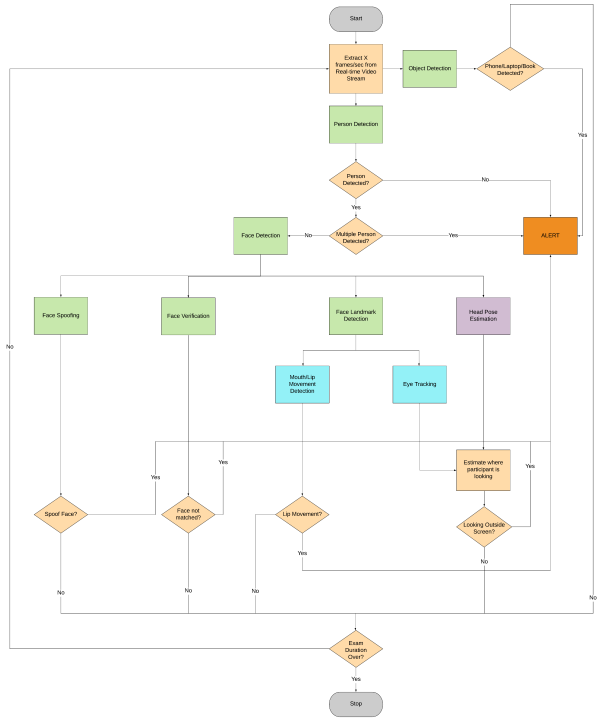

Fig. 3. Architecture of the proposed Online Proctoring System.

Table 2. Functionalities in our system

Category	Functionality	Technique
Base	Person Detection and Counting	Pre-trained
	Object Detection	Pre-trained
	Face Detection	Pre-trained
Authentication	Face Verification	Pre-trained
	Face Spoofing	Pre-trained
Abnormal Behavior Detection	Image-based Head Pose Estimation	Trained from Scratch
	Eye Tracking	Image-Processing
	Mouth Movement Analysis	Image-Processing

4.2. Person Detection and Counting

We used OpenCV's [11] YOLOv3 object detector for detecting and counting the number of people in the frame. If no one or more than one person is detected for more than 10 consecutive frames, then the examinee is said to be cheating. Outputs from the Person Detection and Counting module are visible in Fig. 4. and Fig. 5.

Fig. 4. Output from Person Detection and Counting module for frame with single person.

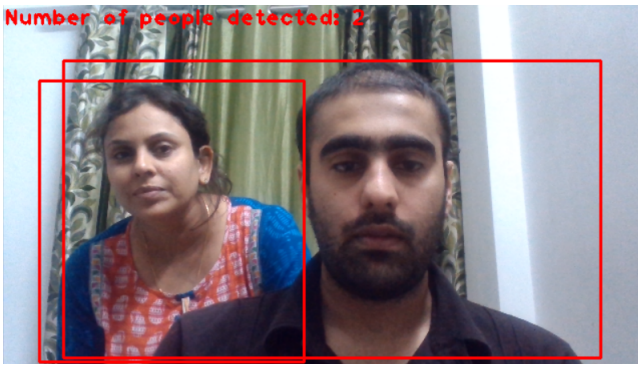

Fig. 5. Output from Person Detection and Counting module for frame with multiple person.

4.3. Object Detection

OpenCV's YOLOv3 object detector was also used for finding any instances of banned items including mobile phones, laptops, TV, and books. If one or more than one instance of any banned item is detected for more than 10 consecutive frames, then the examinee is said to be cheating. Outputs from the Object Detection module are shown in Fig. 6. and Fig. 7.

Fig. 6. Output from Object Detection module for frame with no banned objects.

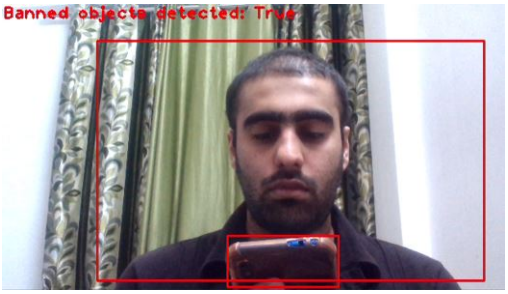

Fig. 7. Output from Object Detection module for frame with banned object.

4.4. Face Detection

We used OpenCV's DNN (Deep Neural Network) module to find the examinee's face in the frame. The face detector is based on the Single Shot Detector (SSD) framework with a ResNet base network. Output from the Face Detection module is shown in Fig. 8.

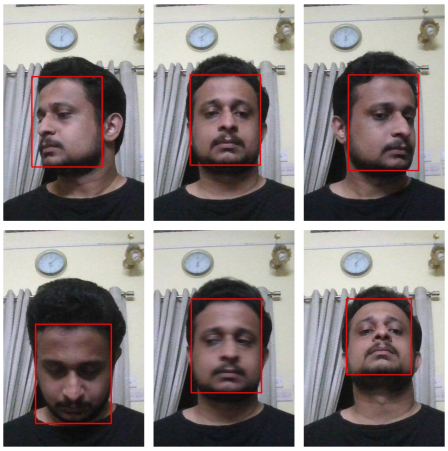

Fig. 8. Output from Face Detection module.

4.5. Authentication

4.5.1. Face Verification

We used Dlib's [12] face verification model to get the examinee's name. The model uses a pre-trained Resnet50 CNN model to extract a 128D feature vector from all facial images in the database. Then the model uses the same steps on the detected examinee's face to extract a 128D feature vector. After that, Euclidean distance is calculated between the two feature vectors. If the distance is below a certain threshold, then both faces are assumed to be the same. Dlib's default threshold of 0.6 was used for face verification. Outputs from the Face Verification module are illustrated in Fig. 9. and Fig. 10.

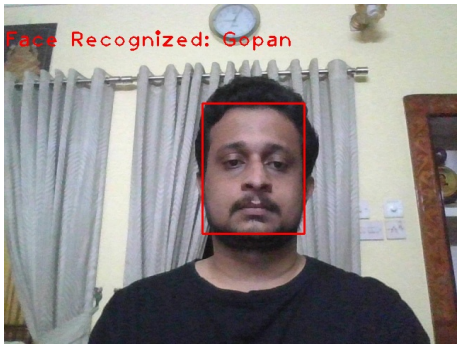

Fig. 9. Output from Face Verification module for frame with valid examinee.

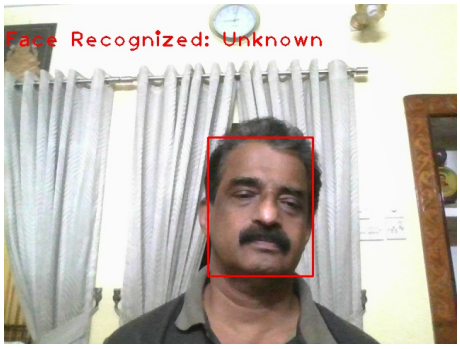

Fig. 10. Output from Face Verification module for frame with invalid examinee.

4.5.2. Face Spoofing

To identify whether the examinee is real or a photograph, we implemented face spoofing functionality. After capturing the examinee's face image using the Face Detection module, it is further converted into YCrCb and CIE L*u*v* color spaces using OpenCV. Later, histograms are calculated from both the color spaces and concatenated together. The concatenated histogram is sent to Scikit-learn's [3] ExtraTreesClassifier model for classifying face into real/spoof. If the face

is classified as a spoof for more than 10 consecutive frames, then the examinee is said to be cheating. Outputs from the Face Spoofing module are shown in Fig. 11. and Fig. 12.

Fig. 11. Output from Face Spoofing module for frame with real face.

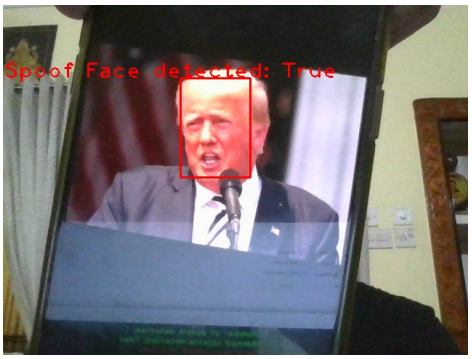

Fig. 12. Output from Face Spoofing module for frame with spoof face.

4.6. Abnormal Behavior Detection

4.6.1. Head Pose Estimation

We trained a lightweight head pose estimation model that can achieve real-time performance in a system with low computational power. Our method can predict accurate pitch, yaw, roll angles of a person directly from the face crop without the requirement of landmark or depth maps. We decided to train a pose estimation network from scratch instead of using landmark because we believe that deep networks have large advantages compared to landmark-to-pose methods due to the following reasons:

- Deep networks are not dependent on the head model chosen, the landmark detection method, the subset of points used for alignment of the head model, or the optimization method used for aligning 2D to 3D points [13].
- They always output a pose prediction which is not the case for the latter method when the landmark detection method fails especially for extreme poses [13].

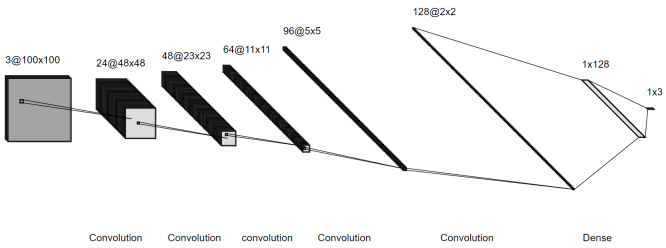

Fig. 13. Convolutional neural network architecture of proposed head pose estimation model.

Our lightweight regression model architecture is illustrated in Fig. 13. We used a 5x5 convolution layer with stride 2 for the first layer followed by 4 3x3 convolution layers with stride 2 for feature extraction. The feature extraction is followed by a dense layer with 128 nodes and the output regression layer with 3 nodes. For all layers except the last convolution layer and regression network, we used the ReLU activation function. For the last convolution layer before Flatten layer and the following dense layers, we used the tanh activation function. Finally, for the output layer, we used the linear activation function. Fig. 14. illustrates the output from the Head Pose Estimation module.

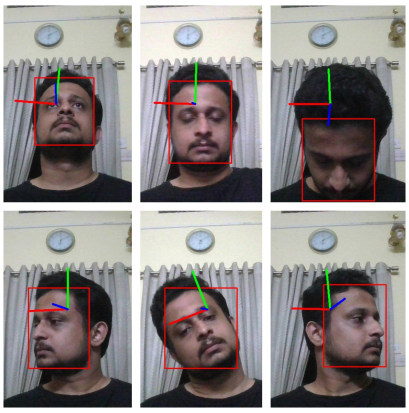

Fig. 14. Output from Head Pose Estimation module. 3D (red, green and blue) vectors are used to illustrate the predicted head pose angles (pitch, yaw and roll)

4.6.2. Eye Tracking

We used Dlib's pre-trained network for detecting and predicting 68 facial landmarks on the examinee's face. Left eye is defined by the following landmarks - 36,37,38,39,40,41. Right

eye is defined by the following landmarks - 42,43,44,45,46,47. First, we segmented the eye regions by using a mask. Then we applied binary thresholding on the eye regions to separate the eyeballs from the rest of the eye regions. Eyeballs become black and the rest regions stay white. Then a vertical separator was created at the middle of each eye. Finally, to determine if the examinee is looking left or right, we defined an eye-tracking ratio as:

$$AvgETR = \frac{RightEyeETR + LeftEyeETR}{2}$$

where,

$$RightEyeETR = \frac{No.\ of\ white\ pixels\ on\ left\ side}{No.\ of\ white\ pixels\ on\ right\ side}$$

$$LeftEyeETR = \frac{No.\ of\ white\ pixels\ on\ left\ side}{No.\ of\ white\ pixels\ on\ right\ side}$$

After extensive trial and testing, we fixed the following thresholds for the AvgETR: $\leq$0.35 (looking outside the screen), 0.36 to 3.9 for the center (looking at the screen), $\geq$4 for left (looking outside screen). If the examinee is looking outside the screen for more than 10 consecutive frames, then the examinee is said to be cheating. Output from the Eye-tracking module is shown in Fig. 15.

Fig. 15. Output from Eye tracking module.

4.6.3. Mouth Movement Analysis

Lip region is defined by the following landmarks - 60, 61, 62, 63, 64, 65, 66, 67. To determine if the mouth is open or closed, we defined a lip aspect ratio as:

$$L.A.R = \frac{|P(62) – P(66)|}{|P(60) – P(64)|}$$

After extensive trial and testing, we fixed the threshold to be 0.1. If L.A.R>0.1, this means the mouth is open, else it is closed.

Furthermore, to check if the person is speaking or not, we defined a buffer. If the examinee keeps his mouth open for more than 10 consecutive frames, then the activity is classified as speaking and hence cheating. Outputs from the Mouth Movement Analysis module are shown in Fig. 16. and Fig. 17.

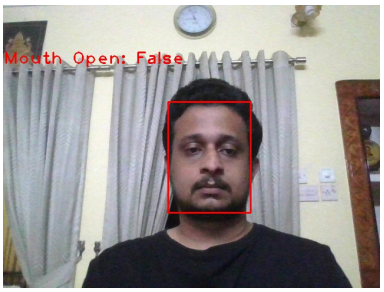

Fig. 16. Output from Mouth Movement Analysis module for frame containing examinee with closed mouth.

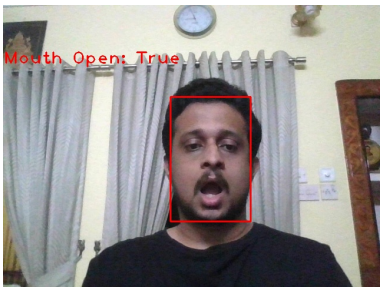

Fig. 17. Output from Mouth Movement Analysis module for frame containing examinee with open mouth.

5. EXPERIMENTAL RESULTS

5.1. Head pose Training

Our model was trained on images from the Pandora dataset. The model takes a face crop rescaled to 100 x 100 as input. The face detection bounding box output is enlarged by 100% and the resulting bounding box is used to crop out the head region and pass that as input to the network. We used the Adam optimizer with a learning rate of 0.001 to train the model. The model was either trained for 100 epochs or was used with early stopping which monitored the validation loss with the patience of 20 epochs. The corresponding training loss and validation loss achieved for the best model is shown in Fig. 18. The model was trained using Mean Squared Error (MSE) as a loss function and evaluated on the test dataset using Mean Average Error (MAE). To prevent the model from overfitting we used dropout regularization after each convolution and dense layer.

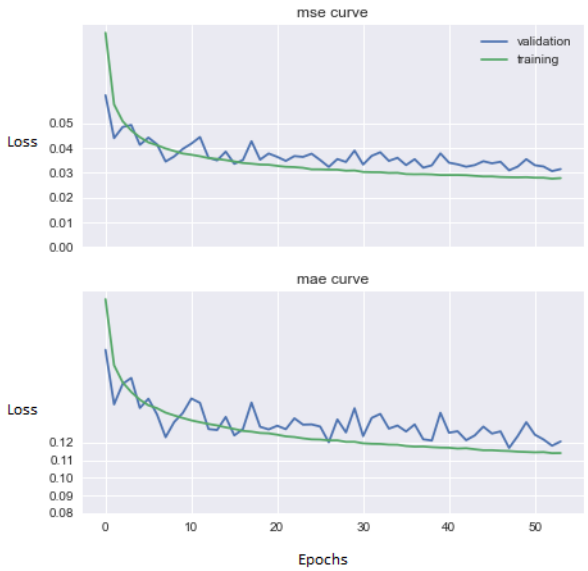

Fig. 18. Training and validation loss curves for Head Pose Estimation model.

5.2. Head pose Evaluation

We evaluated our lightweight head pose estimation model on the BIWI benchmark dataset. The dataset has around 15k RGB images with ground truth head pose angle values. The owners of the Pandora dataset have also provided the cropped face region for the BIWI dataset. So, we used this crop directly without manually cropping the face region from the original benchmark dataset. Our lightweight head poses estimation model was able to achieve a better accuracy compared to famous landmark-based approaches and 3D dense model-based methods. Table 3 shows the comparison of results obtained by our model with Dlib [12] and 3DFFA [6] models. We took performance evaluation results of Dlib and 3DFFA from Hopenet's (state-of-the-art model) paper. Hopenet is a deep learning-based classification method for fine-grained head pose estimation. Even though the accuracy obtained by Hopenet is much higher than our lightweight model, we haven't used it in our online proctoring examination system. The Hopenet uses Resnet as the backbone feature extractor because of which, the computational complexity of the model is much higher and will not provide real-time performance for low-cost systems. Hence its performance is not compared with our lightweight model in Table 3. MAE is used to evaluate the performance of all models.

Fig. 19. illustrates the predicted head pose vectors of our model on the BIWI benchmark dataset. This shows that our model was able to perform decently even for large head pose angles.

Table 3. Comparison of proposed head pose estimation model accuracy (MAE) with other available models

Model	Pitch	Yaw	Roll	MAE
Our model	**11.000**	**13.927**	7.471	**10.799**
Dlib	13.802	16.756	**6.190**	12.249
3DFFA	12.252	36.175	8.776	19.068

Fig. 19. Results by proposed head pose estimation model on BIWI benchmark dataset.

6. CONCLUSION AND FUTURE WORK

In this work, we proposed an Intelligent System that uses a webcam to monitor examinees during the online examination. The solution offers a comprehensive monitoring and analysis suite to prevent examinees from cheating in an online exam. Functionalities include user authentication and abnormal behavior monitoring.

Currently, our pipeline runs at less than 30 frames per second (fps) because of the complexity of the models that we have used. In the future, we are planning to improve the fps by using lightweight models and proper optimization techniques like quantization. Moreover, the performance of the face spoofing model is not satisfactory. In the future, we are planning to replace it with a more accurate model. Currently, we are using an image processing based eye-tracking method. Later, we'll replace it with a deep learning based gaze estimation model that accurately estimates where the examinee is looking.

7. CONTRIBUTIONS

7.1. Apar Garg

My contributions can be broadly classified into 2 heads:

- Implemented the following 7 vision-based functionalities in python backend: (1) Person Detection and Counting - Detect and count the number of people and report if no person or more than one person is detected

in the video frame. (2) Object Detection - Find and report any instances of banned items in the video frame including mobile phones, laptops, TVs, and books. (3) Face Detection – Detect the face of the examinee in the video frame. (4) Face Verification - Match the examinee's face in the video frame against a database of faces to authenticate the examinee. (5) Face Spoofing – Check whether the face of the examinee in the video frame is real or a photograph (spoof). (6) Eye Tracking - Track the eyeballs of the examinee during the exam and report if he/she is looking outside the screen. (7) Mouth Movement Analysis - Find if the examinee speaks during the exam by checking if he/she opens the mouth for a certain duration.

- Prepared the final project deliverable (Unethical Activity Detection System) by combining and organizing all the vision-based functionalities/modules.

7.2. Gopan Ravikumar Girija

My contributions can be broadly classified into 3 categories:

- Developing a lightweight head pose estimation model from scratch. I intensively experimented on different combinations of hyperparameters and activation functions until I landed on a model that performed best in my experiments. The model had approximately 300k parameters and achieved better performance than the landmark-based approach.

- Performing intensive testing and debugging on our final proctoring system pipeline.

- Editing and preparing the presentation video and Writing many sessions in the report.

8. REFERENCES

[1] Swathi Prathish, Kamal Bijlani, et al., "An intelligent system for online exam monitoring," in *2016 International Conference on Information Science (ICIS)*. IEEE, 2016, pp. 138–143.

[2] Yousef Atoum, Liping Chen, Alex X Liu, Stephen DH Hsu, and Xiaoming Liu, "Automated online exam proctoring," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609–1624, 2017.

[3] Senbo Hu, Xiao Jia, and Yingliang Fu, "Research on abnormal behavior detection of online examination based on image information," in *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. IEEE, 2018, vol. 2, pp. 88–91.

[4] Vahid Kazemi and Josephine Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1867–1874.

[5] Adrian Bulat and Georgios Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1021–1030.

[6] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li, "Face alignment in full pose range: A 3d total solution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 78–92, Jan 2019.

[7] Nataniel Ruiz, Eunji Chong, and James M. Rehg, "Fine-grained head pose estimation without keypoints," 2018.

[8] Yijun Zhou and James Gregson, "Whenet: Real-time fine-grained estimation for wide range head pose," 2020.

[9] Guido Borghi, Matteo Fabbri, Roberto Vezzani, Simone Calderara, and Rita Cucchiara, "Face-from-depth for head pose estimation on depth images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 3, pp. 596–609, 2018.

[10] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool, "Random forests for real time 3d face analysis," *International journal of computer vision*, vol. 101, no. 3, pp. 437–458, 2013.

[11] Gary Bradski and Adrian Kaehler, "Opencv," *Dr. Dobb's journal of software tools*, vol. 3, 2000.

[12] S Sharma, Karthikeyan Shanmugasundaram, and Sathees Kumar Ramasamy, "Farec—cnn based efficient face recognition technique using dlib," in *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*. IEEE, 2016, pp. 192–195.

[13] Nataniel Ruiz, Eunji Chong, and James M Rehg, "Fine-grained head pose estimation without keypoints," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 2074–2083.