International Journal of Future Generation Communication and Networking
Vol.14, No. 1, (2021), pp. 4173 – 4181

# Behavioral Tracking Analysis using AI

**Shubham Dhanawade[1], Pramod Dhanure[2], Shreeprasad Sonar[3], Prof. P. G. Chilveri[4], Prof. A. B. Chandgude[5]**

[1,2,3]*Student, Department of E&TC, SKNCOE, SPPU, Pune, India*

[4,5]*Assistant Professor, Department of E&TC, SKNCOE, SPPU, Pune, India*

[1]Shubhamdhanawade29@gmail.com

[2]Pramod.dhanure24@gmail.com

[3]sonarshreeprasad@gmail.com

***Abstract-***

*One of the critical parts of an educational program is the exam and educational programs on an online platform are no exception. During the exam, there might be a possibility of malpractice so its detection and prevention are important. To prevent this an online proctored system is at most important. This system monitors the test takers continuously similar to human supervision. The objective behind this is to avoid any malpractice happening by detecting their voice, active window, gaze. It reduces the workload required for identifying, verifying, and communicating with off-campus proctors.*

***Keywords:*** *Python, dlib, OpenCV, Web camera, AI*

## I. INTRODUCTION

As now the education system is shifting to a virtual platform, there might be possibilities of any misconduct happening. In an Educational program, exams are a critical component. The detection and malpractice prevention in exams is important to maintain academic integrity. Exams are supervised in a conventional and proctored classroom environment, where the students are continuously monitored by a person present in the room. AI-based auto proctored system for continuous online supervision is introduced, in case if there is an unavailability of halls and during the lockdown, if the exam needs to be conducted. This system takes access to the candidate's system web-camera and microphone. By tracking the eye movement and any audio disturbance the candidate is warned by the system. A candidate is monitored continuously, and based on the proctored report if a candidate is guilty or not is decided. The goal of this system is to maintain the academic integrity of the exams and prevent the students from misconduct.

## II. REVIEW OF LITERATURE

| Author Citation | Adopted Methodology | Feature | Limitation |
|---|---|---|---|
| Maryam Sohail [1] | Point of Gaze (PoG) Calculation | Simple mathematical equations can be used to get PoG, pupil detection accuracy 100% | The system should work at a frame rate of 10 to 15 frames per second, pointer size is large due to low resolution. |
| Zhou Zhang [2] | Stereo matching | Toovercome shortcomings of existing facial recognition methods, a stereo matching algorithm was proposed to | They impose more severe restrictions on the students than a human proctor would, The reliability of these |



ISSN: 2233-7853 IJFGCN
Copyright ⓒ 2020 SERSC

| | | | |
|---|---|---|---|
| | | improve the reliability offacial recognition, 'C++ accelerated massive parallelism was used in the implementation of the stereo matching algorithm, and therefore the speed of execution of the program was increased significantly. | software systems highly depends on the initial conditions under which the educational activity is started. |
| Sebastian Höfffffner[3] | GazeCapture | Gaze provides flexible data sources consisting of multiple pre-defined steps, Gaze is publicly available and the source code and software can be modified and redistributed without any limitations. | Webcams are limited to small frame rates and it is extremely difficult to find the real specifications like sensor sizes and focal lengths for many webcams that need iTracker to detect gaze points. |
| Yousef Atoum [4] | Active window detection, gaze estimation | Low-level features are extracted from six basic components: user verification, text detection, speech detection, active window detection, gaze estimation, and phone detection, features are then processed in a temporal window to acquire high-level features. | It is possible that the automatic OEP system might not achieve perfect performance. |
| Ahmad Aljaafreh [5] | Pygame, Dlib, and OpenCV feature detection | Raw data of gaze location and dynamic stimuli is obtained using a low-resolution webcam, an algorithmic approach to process and extract data. | The results demonstrate a significant difference between normal and Multiple Sclerosis (MS) participants, latency time and amplitude gain are essential to perform a satisfactory diagnosis but 9 out of 10 MS participants, exhibit a significant difference for both parameters, either more or less. |
| Mortiz Kassner, William Patera [6] | Open-source software, a Pupil detection algorithm | Under idle conditions, 0.6 degrees of accuracy is obtained. Precision is obtained at 0.08 degree | Paralox error and tracking robustness in IR rich environment. |
| PrabinSharma, Shubham Joshi [7] | The haar-cascade algorithm, CNN, | To determine the eye movement and facial emotion of the students with the help of an in-built webcam. It is observed that the students with the best scores have a | The system was unable to correctly find the concentration index as the students wore the glasses. |



ISSN: 2233-7853 IJFGCN
Copyright ⓒ 2020 SERSC

| | | | |
|---|---|---|---|
| | | higher concentration index. | |
| Dhaval Pimplaskar, Dr.M.S.Nag mode, Atul Borka [8] | Drowsiness detection algorithm, C++ using OpenCV | For detecting drowsiness in real-time.<br>Memory consumption for detecting both the eyes | It shows promising results under good lighting conditions. |
| Aniket M. Taksande, Prof. R. P. Patil [9] | Viola-Jones algorithm, OpenCV libraries, Xilinx | By using the OpenCV library function gives more promising and real-time results in image processing.<br>When used along with Vivado HLS it shows the alternative method for implementing given face and detection.<br>It helps the disabled person to control the computer effectively just by his eye movement. | It will not gives accurate results when the light intensity in a room is not sufficient to capture images.<br>If the person is wearing any spectacles will resist eye detection and tracking. |
| Liu Li, Mai Xu, Xiaofei Wang, Lai Jiang, Hanruo Liu [10] | AG-CNN | Glaucoma could be detected using lighted by the visualized maps of pathological areas, based on the predicted attention maps.<br>AG-CNN approach significantly advances state-of-the-art glaucoma detection. | AG-CNN method over the RIM-ONE database is above 0.83, despite slightly smaller than the results over our LAG database. |
| Nora Hollenstein, Maria Barrett, Marius Troendle, Francesco Bigiolli, Nicolas Langer, Ce Zhang [11] | NLP, gaze and EEG | It shows great potential in improving NLP tasks and facilitates insights into language processing that can be applied to NLP. | While integrating gaze or EEG signals separately significantly outperforms the baselines, the combination of both does not further improve the results. |
| Hemalatha Vadlamudi [12] | YOLO based algorithm, GMM model, CNN, and Deep learning | To identify the targeted object in a moving sequence.<br>To analyze the object motion in the video. | It is insufficient to detect the possibly present error.<br>Sometimes it is difficult to track the object. |



ISSN: 2233-7853 IJFGCN
Copyright ⓒ 2020 SERSC

| | | | |
|---|---|---|---|
| Ildar Rakhmatulin, Andrew T. Duchowsk [13] | Deep learning, YOLOv3 | A dual coordinate system is given for controlling the computer with the help of gaze.<br>A method of labeling the eyes is given, in which 3 objects are used to track gaze. | Different lighting, reflections on the eye, camera resolution, and camera position can affect the accuracy of the developed device. |
| Shima Rashid, Krista Ehinger, Andrew Turpin, Lars Kulik [14] | CNN, Logistic Regression | It detects objects target in the natural background that closely mimics human performance.<br>It detects the known target in the natural background based on biological aspects of the human visual system. | This model requires training on a large set of backgrounds so that it will not select an unrepresentative background. |

## III. METHODOLOGY

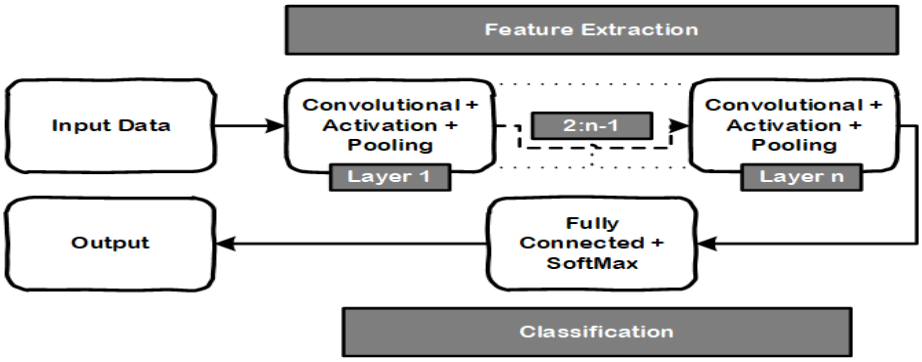



**Fig 1. Block Diagram of CNN**

Input data to the CNN network is the image dataset, which is then passed through a convolutional layer which in combination has an activation and pooling layer. Depending upon the required functionality the image dataset is passed through this feature extraction phase many times. Then after the required data is being obtained from the feature extraction phase it moves forward through the classification phase which consists of a Fully connected layer and softmax, where data is being classified into different classes with the help of fully connected layers. The softmax function is used for multiclass classification. Then the required output is obtained at the end.A feature map is created in CNN by applying a filter to an input that summarizes most relevant features in the input. The convolution layer gives CNN its name. The operation called a "*convolution*" is done by this layer. In convolution layer multiplication of array of input data and a two dimensional array of weights takes place with the help of filter or kernel.Each value of the feature map is passed through a*n activation function*, right after convolution has taken place. The activation function is used to activate the neuron by adding weighted sum and bias to it and also introduces non-linearity at the output of the neuron which in turn brings the relevant features into focus. Most commonly used activation function in CNN is ReLU. Output of the activation function is a feature map which is fed to the input of pooling layer



ISSN: 2233-7853 IJFGCN
Copyright ⓒ 2020 SERSC

where the pooling layer reduces the dimension of the feature map, by reducing the number of parameters.The high-level reasoning is performed in the fully connected layer. The neurons in fully connected layer are connected with the activations of previous layer. Softmax function is used to get the ultimate probabilities of the image of a particular class whose values ranges between 0 and 1.

## IV. EXPERIMENTATION

**Eye Detection:**

The first thing to do is to find eyes before image processing and to find the eyes first thing to find is face. To find the face DNN face detector in OpenCV is used.To find faces network is loaded using cv2.dnn.readNetFromCaffe and model's layers and weights are passed as its arguments.To achieve the best accuracymodel is run on on BGR images which are resized to 300x300 by applying mean subtraction of values (104, 177, 123) for blue, green and red channels respectively.A 4-D array is returned at end, which contains the confidence and coordinates which are scaled down to the range of 0 to 1 and by multiplying them by the original width and height, predictions for the original image can be obtained as opposed to of the 300x300 on which the model predicted. Following is the image for the points that are going to be detected on the face.

From 1 to 17: Jaw
From 18 to 22: First Eyebrow
From 23 to 27: Second Eyebrow
From 37 to 42: First Eye
From 43 to 48: Second Eye
From 28 to 36: Nose
From 48 to 60: External Part Of Lips
From 61 to 68: Internal Part Of Lips

**Fig 2. Facial Landmarks**

First, libraries are imported which are, OpenCV which is a library designed to solve real time computer vision problems, NumPy which is a Python library used for working with arrays and Dlib library which is the most popular library for detecting landmarks on the face. Then frames are obtained from the camera usingcap = cv2.VideoCapture(0), 0 is for the first webcam, if having a different webcam index can be changed to capture a frame from that camera. Then a loop is run to get the frame from the camera.Ret,img = cap.read() is used to get video frames. After that, image is converted to greyscale format.





ISSN: 2233-7853 IJFGCN
Copyright ⓒ 2020 SERSC

**Fig 3. Initial Image After cap.read()** **Fig 4. GreyScale Image**

Faces are detected in the detections after loading the model using. First, model is loaded using cv2.dnn.readNetFromCaffe(configFile, model file)the input to this model must be of 300 X 300 dimension so cv2.resize() is used to resize the image. With function cv2.dnn.blobFromImage() mean subtraction, normalizing, and channel swapping is done, on an image obtained through a webcam. Passing blob through the network to detect and predict.Here co-ordinates of face are obtained with one point at the top left of the screen where the face is detected and other point at the right bottom and these specific points are needed to get a rectangle where our face is.The same can be done using a grayscale image.

**Fig. 5 Coordinates after Shape_to_np converts the dlib shape objects to a NumpyArray**

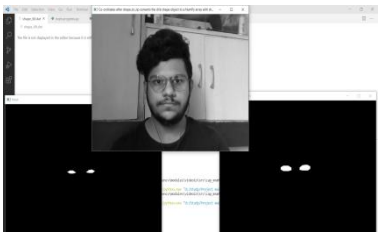


**Fig. 6 Coordinates after Shape_to_np converts the dlib shape objects to a NumpyArray**

**Fig. 7 Eyes Grey**

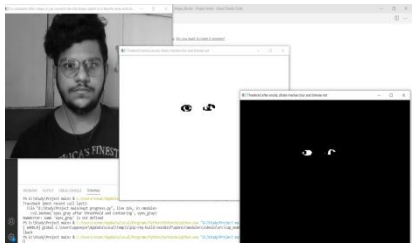


**Fig.8 Threshold before and after erode, dilate median blur and bitwise not**

**Eye DetectionResults** :

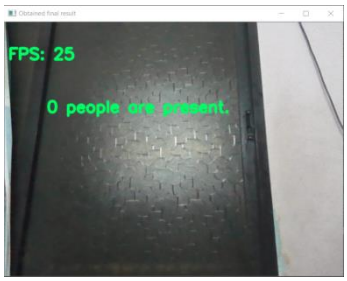



**Fig.9 Results while no one is present**

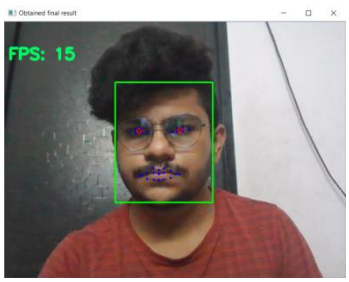



**Fig.10 Results while looking normal**

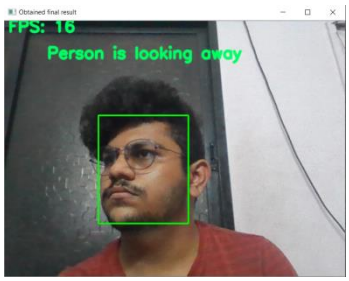
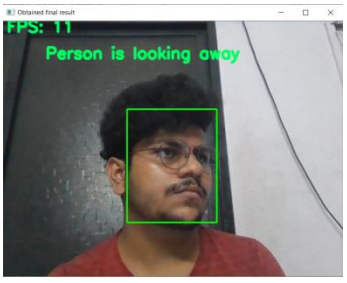
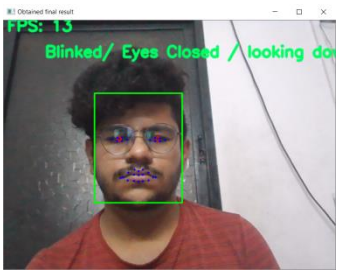
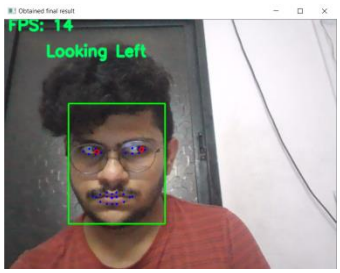



**Fig.11 Results while looking away**






ISSN: 2233-7853 IJFGCN
Copyright ⓒ 2020 SERSC

**Fig.12 Results while looking away**

**Fig.13 Results while blinked**

**Fig. 14 Results while looking left**

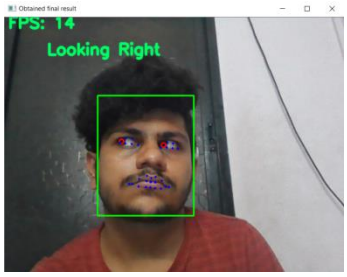



**Fig.15 Results while looking right**

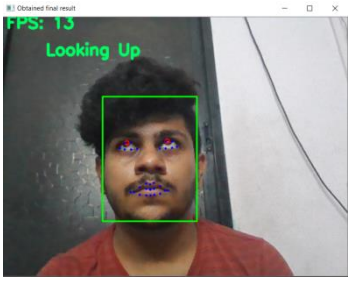



**Fig.16 Results while looking up**

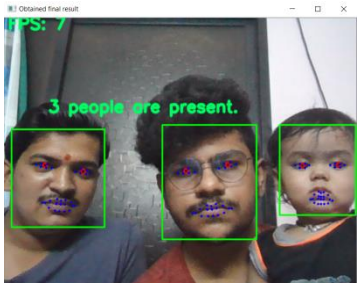



**Fig.17 Results while multiple people are present**

**Object Detection:**

For detecting objects a video is captured through a webcam and YOLOV3 Pre-trained model is used. First the OpenCV library for computer vision and NumPy library for working with arrays is imported. Webcam is enabled to capture real-time video by cv2.videocapture(0), then after running while loop frames are being captured by function cap.read().44. Then height, width, and channels are determined by img.shape, where height is at index 0, width is at index 1, and channels at index 2. With function cv2.dnn.blobFromImage() mean subtraction, normalizing, and channel swapping on an image is performed which was obtained through a webcam. Blob is passed through the network to detect and predict.


**Fig.18 Chair detected as an object**


**Fig.19 Person detected as Object**


**Fig.20 Person and Cell phone detected as Object**


**Fig.21 Car detected**

## V.RESULTS & DISCUSSION



ISSN: 2233-7853 IJFGCN
Copyright ⓒ 2020 SERSC

As this system uses AI-based Proctor the results turn out to be very helpful for the organizations which monitor through this system. As this system makes use of a web camera for gaze estimation, object detection, or any other misbehavior in the frame is suddenly caught by the system. Based on the images captured by the web camera the candidate is being monitored and the warning is being displayed on the user's screen. A microphone captures the surroundings as well as the user's nose and tries to identify them. The use of python with dlib and OpenCV makes the system easy to build. This system turns out to be user-friendly by taking the survey of many test takers and also the overall efficiency of this system makes it better than human proctor. This work presents a comprehensive framework for online exam proctoring. By the reviews of the test-takers, it is observed that the system turns out to be user-friendly, But by making the slight changes the system can be improved for further uses. With the help of more advance developing algorithms like deep learning more accurate results can be obtained. Additional components such as pen detection can also be used. By displaying a simple icon on the computer screen to check that the web camera can see it or can play a quick sound clip to check that microphone can hear it these random commands and interventions make the system more robust against cheating behavior. This system can also allow humans to manually inspect the candidate. This manual inspection by using this system will help to verify the true detections as well as it will avoid false detection.After observing the output it detected the object properly, but in some cases, it shows wrong object detection. To improve this model needs to be trained with more image datasets with different classes. The accuracy of object detection also can be improved by using different pre-trained models like Resnet50, Retinanet.

## VI. CONCLUSION

Eye-tracking and proctoring using a webcam and there can be many ways to solve this problem. In this work, a computer vision algorithm-based solution is implemented. Computer vision algorithms can be successfully used for feature detection. Accuracy for feature extraction depends on image quality and lighting conditions. With the development of Computer Vision being backed by tech giant Intel, Future improvements in the system are promising.

## VII. SCOPE FOR FUTURE ENHANCEMENTS

With continuous development in CNN and OpenCV being backed by Intel corporation as results are expected to be more accurate in the future. With the use of a good webcam with a high frame rate future work is needed on integrations to get more accuracy in the future.

## REFERENCES

[1] I. E. Allen and J. Seaman. Grade change: Tracking online education in the united states, 2013. Babson Survey Research Group and Quahog Research Group, LLC. Retrieved on, 3(5), 2014. □

[2] Y. Atoum, S. Srivastava, and X. Liu. Automatic feeding control for dense aquaculture fish tanks. IEEE Signal Processing Letters, 22(8):1089–1093, 2015.



ISSN: 2233-7853 IJFGCN
Copyright ⓒ 2020 SERSC

[3] D. L. Baggio. Enhanced human-computer interface through webcam image processing library. Natural User Interface Group Summer of Code Application, pages 1–10, 2008.

[4] S. V. Bailey and S. V. Rice. A web search engine for sound effects. In Audio Eng. Society Convention 119. Audio Eng. Society, 2005.

[5] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. ACM T- TIST, 2(3):27, 2011.

[6] J. Chen and X. Liu. Transfer learning with one-class data. Pattern Recognition Letters, 37:32–40, 2014.

[7] C.D. McuMurrough, V. Metsis, J. Rich, and F. Makedon. An eye-tracking dataset for point of gaze detection. In ETRA,2012.

[8] C.H. Morimoto and M.R.Mimica. Eye gaze tracking techniques for interactive applications.CVIU,2005.

[9] Z.Zhu, Q. Ji. Eye gaze tracking under natural head movements. In CPVR,2005.

[10] X.Zhang, Y.Sugano, M.Fritz, and A.Bulling. Appearance-based gaze estimation in the wild. In CPVR,2015.

[11] K. Rayner. Eye movements in reading and information processing:20 years of research. Psychological Bulletin,1998.

[12] B.A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar. Gaze locking: Passive eye contact detection for human-object interaction. In UIST, 2013.

[13] K. A. F. Mora, F. Monay, and J.-M. Odobez. Eyediap: A database for development and evaluation of gaze estimation algorithms from rgb and RGB-d cameras. ETRA 2014.

[14] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical report, 1998. SKNCOE. Pune-41, Dept. of E&TC Engineering

[15] J. Chen and Q. Ji 3d gaze estimation with a single camera without IR illumination. In ICPR, 2008.

[16] A. Duchowski. Eye Tracking Metholodogy: Theory and Practice. Springer Science & Business Mdedia,2007

[17] https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

[18] https://docs.opencv.org/master/dd/d49/tutorial_py_contour_features.html

[19] https://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html

[20] https://docs.opencv.org/master/d3/d05/tutorial_py_table_of_contents_contours.html

[21] https://ibug.doc.ic.ac.uk/media/uploads/documents/sagonas_cvpr_2013_amfg_w.pdf

[22] https://ibug.doc.ic.ac.uk/resources/300-W/

[23] https://opencv-tutorial.readthedocs.io/en/latest/yolo/yolo.html

[24] https://dev.to/afrozchakure/all-you-need-to-know-about-yolo-v3-you-only-look-once-e4m



ISSN: 2233-7853 IJFGCN
Copyright ⓒ 2020 SERSC