

Deep Learning and Computer Vision-Based Attentiveness Tracking for Autistic Students in Online Learning Environments

Nanaki Singh^[1]

Jumeirah English Speaking School, Dubai, UAE

Mohan Kshirsagar^[2]

On My Own Technology Private Limited, Mumbai, India

Reetu Jain^[3]

On My Own Technology Private Limited, Mumbai, India

Shekhar Jain^[4]

On My Own Technology Private Limited, Mumbai, India

Abstract: Individuals with Autism Spectrum Disorder (ASD) struggle with attention deficiency. Concentration loss in autistic patients has been heightened by the onset of virtual learning during the COVID-19 pandemic as facilitators now have little control over a student's immediate, potentially distracting, physical learning environment. Developing a methodology to track the attentiveness of an autistic student for the duration of an online lesson is crucial to support teachers and to discern the quantity of information the student is retaining. 'Panacea' is a deep learning software that tracks the facial expressions, facial orientation, and pupil orientation of an autistic individual during a virtual lesson and determines whether the student is attentive or not. It provides real time data about a student's level of concentration to an instructor. Live input data is taken from a web-camera, resized, and passed through a hard-coded program that detects blinking, pupil orientation and facial orientation. Additionally, a machine learning model is used to determine an individual's facial emotion and improve the accuracy of the results. The outputs from the four functions assigned manual probabilities of 25%, 18%, 17% and 40%, respectively, and passed through a probability density function that outputs either 'Attentive' or 'Non-Attentive'. These results have an accuracy of 97%. Data from every six frames is aggregated, averaged, and displayed on the screen alongside the real-time video footage and on a continuously updating graph. The results, date and time are then recorded in an excel document, which can be used for analysis at a later date.

Keywords: Autism Spectrum, Machine Learning, Convolutional Neural Networks, Pupil Orientation, Facial Orientation, Facial Expressions, Facial Emotions, People of Determination.

I. Introduction:

Autistic Spectrum Disorder (ASD) is a neurodevelopmental disorder that is characterised by social, communication and behavioural challenges. The Centre of Disease Control estimates that autism affects over 1 in 54 children^[1] in the United States. Although no treatment for

ASD exists, several multidimensional and multidisciplinary interventions (behavioural approaches, biomedical agents, and alternative medicine) have been used to reduce symptoms, improve cognitive ability, and maximise the child's function^[2].

Leitner ^[3] claims that over a third of autistic children have comorbid attention-deficit/hyperactivity disorder (ADHD) symptoms. When present together, the general psychopathology is heightened, making it even more difficult for a student to concentrate.

Autistic individuals are easily distracted by bright lights, smells and sounds in their surroundings, and without a supervisor to reduce sensory stimuli, they are likely to lose focus. In a traditional classroom setting, a teacher can interact with ASD students and create an environment that caters to their special needs.

However, virtual learning limits an instructor's control over an ASD student's immediate physical environment, increasing chances of sensory overload and concentration loss. It also makes it difficult for lecturers to predict and recognise a decline in attention whilst teaching.

Autistic individuals with extreme behaviour issues that cannot be addressed over the internet struggle to retain vital information in virtual learning sessions. Similarly, non-verbal autistic, who usually receive visual in-person cues, have trouble interacting with teachers and contributing to the lesson. The struggles faced by educators are intensified by a lack of technological know-how, which hinders their ability to support their students.

'Panacea' is a deep learning software that measures the concentration levels of an autistic student in a virtual lesson by evaluating their facial orientation, facial expression and pupil orientation. It produces a binary output of 'Attentive' or 'Non-Attentive' every second, displays the data in graphical format, and stores it onto an excel document.

Literature Review:

Patricia Goldberg developed a machine learning approach to assess the visible engagement of non-autistic, mainstream students in a physical classroom setting. The program combines the results of a self-reported manual rating system with a machine vision-based model based on pupil gaze, head pose, facial expressions, body pose and gestures ^[5].

JanezZaleteji and Andrej Kosirutsilise data obtained through Kinect One sensors to survey the facial and body properties of a student (i.e. gaze point and body posture). This 3D data collection and evaluation approach utilises machine learning algorithms to train classifiers that estimate time-varying attention levels of individual mainstream students ^[6].

Working Principle

Live image data is passed to the model via a web camera and the 68-facial point detector model ^[6] is used to detect an individual's face. The input is resized to capture the face and the dimensions are manipulated to a 48-by-48-pixel size with a bit depth of 8. This improves the clarity of visual features and reduces the processing power per picture. Additionally, the input is converted to greyscale to assist with machine learning.

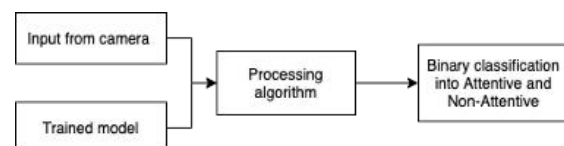


Figure1 – Overview block diagram

Figure1 presents a brief overview of the working model, in which data is taken from a camera and fed into a processing algorithm alongside a trained deep learning model. The convolutional neural

network program is used to enhance the accuracy of results.

The processing program conducts real-time analysis and contains four functions to evaluate pupil and facial orientation. Using the 68-facial points detector pre-trained model, we measure the distance between points on the top and bottom eyelid to determine whether the eyes are open or not. If the eyes are not open, the individual is categorised as ‘Non-Attentive’. Conversely, when the eyes are open, the program evaluates the individual’s facial direction.

The real-time analysis model determines an individual’s facial orientation by standardising the available data and using thresholds and height-to-width ratios. The output of the facial direction function can be categorised into 9 subclasses and classified into ‘Attentive’ and ‘Non-Attentive’, as indicated in *Figure2*.

Model Classification	Model Categorisation into Subclasses				
‘Attentive’	‘Up’	‘Up-left’	‘Up-right’	‘Straight’	
‘Non-Attentive’	‘Down’	‘Down-left’	‘Down-right’	‘Left’	‘Right’

Figure2 – Facial orientation classification

The processing program also studies the region of the eyes the pupil is located in. In order to do this, we evaluate each eye individually and aggregate the results. For example, if the centre of the left pupil is located in the centre of the image and is

fluctuating between the four regions, as shown in *Figure3*, the left eye is looking straight.



Figure3 – Pupil location

Similar analysis is conducted for the second eye. If the person is looking left, right or down, the input is classified as ‘Non-Attentive’, and if the person is looking straight or up and right/left, the image is classified as ‘Attentive’.

When these three features are utilised in conjunction with one another, we achieve an accuracy of 85%. To improve this value, we rely on a deep learning program. This model has been trained on 39,791 images, 30,182 of which belong to an open source dataset and 2,116 of which have been extracted from authentic videos. *Figure4* states how an individual’s facial expressions are classified into 9 emotions and categorised into ‘Attentive’ and ‘Non-attentive’.

Model Classification	Model Categorisation into 9 emotions				
‘Attentive’	‘Happy’	‘Neutral’	‘Surprised’	‘Attentive’	
‘Non-Attentive’	‘Angry’	‘Sad’	‘Fearful’	‘Disgusted’	‘Non-Attentive’

Figure4 – Facial emotion classification

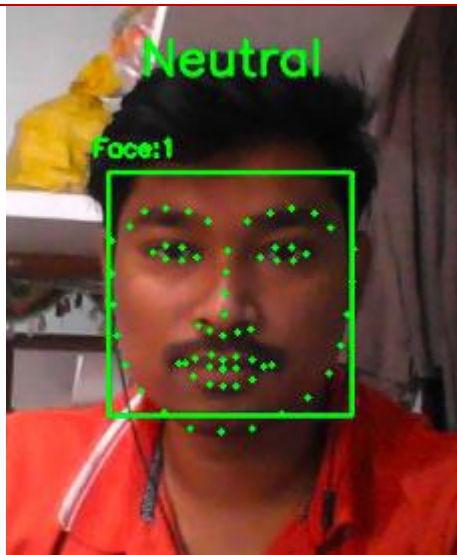


Figure5 – 68 facial point evaluation

Figure5 illustrates how the 68 facial points are evaluated to determine a facial emotion of class ‘Neutral’, which translates to an output of ‘Attentive’.

The outputs from the blink-detector, pupil orientation, facial orientation, and deep learning functions are evaluated using a probability density function (P.D.F.) which are assigned manual weightages of 25%, 18%, 17% and 40%, respectively. These values can be manipulated as per the user’s requirement.

The single result provided by the P.D.F. is either ‘Attentive’ or ‘Non-Attentive’ and is displayed on the screen alongside the date and time. Furthermore, the results are recorded on an excel document and presented in graphical format, with time on the x-axis and the result on the y-axis.

The aggregated machine learning, and real time analysis program produces an accuracy between 95-99% on real-environment data.

Technology:

The high-level block diagram in Figure6 depicts how data processing and evaluation is conducted using real-time analysis and a machine learning algorithm.

The 68 face landmarks model, pre-trained deep learning program, and the necessary packages, libraries and models, are loaded onto the program. Input images are extracted from a web camera at 30 frames per second, resized to 48-by-48 pixels, and converted to greyscale. The data is passed to the Face_utils open source wrapper library, which searches for faces on the screen. If it successfully detects a face, the 68 facial landmarks model identifies the region of the face and produces a cropped image, which is passed to a real-time image processing program.



Figure6 – High-level block diagram

Property	Value
Origin	
Date taken	
Image	
Dimensions	40 x 40
Width	40 pixels
Height	40 pixels
Bit depth	8

Figure7 – Image data properties

Categorisation	Dataset (grey scale)
Happy	
Sad	
Neutral	
Angry	
Disgust	
Fearful	
Surprised	
Attentive	
Non-Attentive	

Figure8 – Input data categorisation

Image processing can be split into two distinct components: facial orientation and pupil orientation. To conduct both processes, the face and its features need to be standardised to account for the individual’s distance from the screen. A threshold is generated by calculating the diagonal pixel length of the image, dividing that by the distance of the individual from the screen, and square rooting it. To find the distance from the screen, we use the following formula [11]: $\frac{2*3.14*180}{w+h(360)} + 3$. Height, h , and width, w , are extracted using the Haas carcass [8] face detection algorithm, which plots a rectangle over the face. An overview of the real-time data analysis is shown in Figure9.

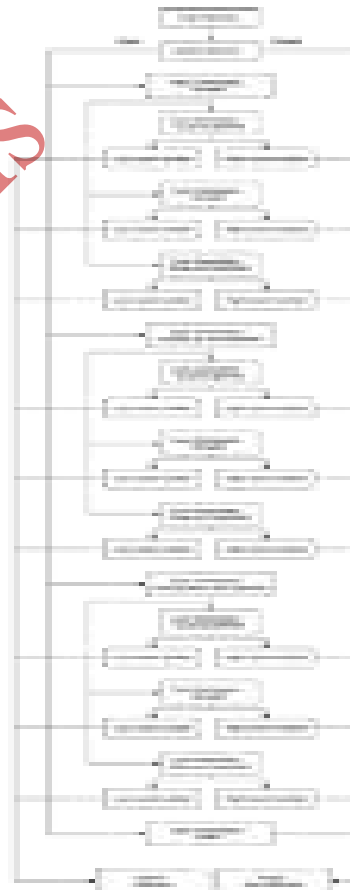


Figure9 – Real-time analysis block diagram

Firstly, the program uses the Blink_Detector function to calculate the distance between the

eyelids of both eyes to determine whether they are open or not. When they are not open, the face is automatically classified as ‘Non-Attentive’ because we assume the person is either bored, tired, or falling asleep. When the eyes are open, the program evaluates facial orientation.

To determine the direction the face is looking in, we calculate the distance of the nose to the left and right eye, both of which are passed into a Right_Left and an Up_Down function alongside the threshold.

The Right_Left function studies the difference in the distance from the nose to both eyes. If the left-eye-distance is greater than the threshold added to the right-eye-distance, the face is looking to the left. As depicted in *Figure10*, the left-eye distance increases as the individual turns their head to the left. Similarly, If the right-eye-distance is greater than the threshold added to the left-eye-distance, the face is looking to the right. However, if the values do not satisfy either requirement, the face is classified as looking straight.






Facial Orientation	Straight	Up	Down	Left	Right
Image					

Figure10 – Facial orientation data

The Up_Down function calculates the mean of the left and right eye distance and uses similar analysis to conduct its evaluation. When an individual is looking up, the camera detects a proportionally

smaller component of the upper face region and measures a smaller value from the left eye and right eye distance. If the value of the mean is less than ten times the threshold, the face is classified as looking up. Conversely, when the individual is looking down, the camera detects a proportionately greater component of the upper face region, so the recorded distance of the left eye and the right eye to the nose increases. If the mean is greater than twelve times the threshold, the face is classified as looking down. However, when the mean value falls between these values, it is classified as looking straight.

In our preliminary examination of autistic students, we observed looking up and to the right/left is an indication of the individual processing information. When he/she is looking down, left or right, they are typically distracted and are focusing their concentration onto another object. When we aggregate the results of both functions, we classify the face into one of nine subclasses and categorise it into ‘Attentive’ or ‘Non-attentive’ as demonstrated in *Figure2*.

After conducting facial evaluation, the model uses the Eye_Crop functions to determine the pupil direction. This is done by examining the region of the eye the pupil is located in. A snapshot of one eye is magnified to produce a rectangular of size 300-by-150 pixels, which is split into four equal quadrants labelled ‘top left’, ‘top right’, ‘bottom left’, and ‘bottom right’. We use a mask to detect the pupil, which is the darkest point of the eye, draw a contour around it, and determine the coordinates of the centre point. Using the classification technique outlined in *Figure11*, we analyse the position of the centre of the pupil for each eye, aggregate the results and provide an

output. Since the specifications for male and female eyes are the same, the program is not gender specific.

Pupil Location	Top-Left/Top-Right	Top-Left/Bottom-Left	Top-Right/Bottom-Right	Situated in the middle of the page, but fluctuating between regions
Meaning	Looking Up	Looking Left	Looking Right	Looking Straight
Classification	'Attentive'	'Non-Attentive'	'Non-Attentive'	'Attentive'

Figure 11 – Pupil location classification

As discussed, we achieve an 85% accuracy by utilising these three functions. However, the convolutional neural network model, outlined in Figure 12, is used to augment the reliability of the results.



Figure 12 – Deep learning model block diagram

Grey-scale facial emotion data taken from an open-source library and extracted from authentic video sources are standardised into nine subclasses and categorised, as directed in Figure 9. The 31,840 and 7,951 training and testing images are split using a ratio of four-to-one. A model with five hidden layers of nodes 32, 64, 128, 128, 1024 was built. It outputs 9 dense functions in categorical form.

The model was trained by passing it through a compile in-built function, that supplies pre-processing data, and a model fit generator in-built function, that converts data into hex format, performs back propagation and updates the weights in the model. The program produced an accuracy of 96.7% as shown in the model accuracy-loss plot in Figure 13. The model was saved as a '.h5' file and loaded onto the processing computer program.

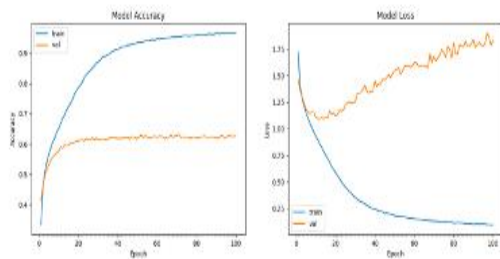


Figure13 – Deep learning accuracy-loss graph

The results from the four functions, Blink_Detector, Eye_Crop, facial orientation and machine learning model, are averaged using a probability density function, which provides a binary classification output of ‘Attentive’ or ‘Non-Attentive’. The assigned probabilities were manipulated and tested for multiple values. We found that the machine learning output was correct 70% of the time, so it was given a 40% weightage. Although the blink detector was successful in detecting eye orientation 85% of the time, it cannot evaluate any other facial feature or provide a balanced result. Therefore, it was given a weightage of 25%. The remaining 35% was split equally amongst the facial orientation and pupil orientation functions.

Function	Machine Learning	Blink Detector	Pupil Orientation	Facial Orientation
Assigned Probability	40%	25%	18%	17%

Figure14 – Assigned probabilities in the aggregated model

The P.D.F outputs either ‘Attentive’ or ‘Non-Attentive’.

Repetitive behaviours in autistic students can affect the real-time analysis of the program, causing the results to fluctuate between ‘Attentive’ and ‘Non-Attentive’ over a matter of seconds. For this reason, we aggregate the data of every six frames, calculate the mean, and display the result on a live video feed output alongside the date and time. It takes approximately forty milliseconds to process each frame. An excel sheet is used to store the per-second result, date and time.

Additionally, this data is displayed on a real-time update Matplotlib bar graph (time count on the x-axis), in which ‘1’ and ‘-1’ on the y-axis denote ‘Attentive’ and ‘Non-Attentive’, respectively.

Results and Conclusion:

In our novel solution, the three visible features of student engagement, facial orientation, facial expression, and pupil orientation, operate in tandem to generate an accurate, verifiable result. The graph in Figure15 is generated by an aggregate program running over 400 seconds of video footage. Figure16 and Figure17 are the results of the real-time analysis and machine learning program running on the same recorded data individually.

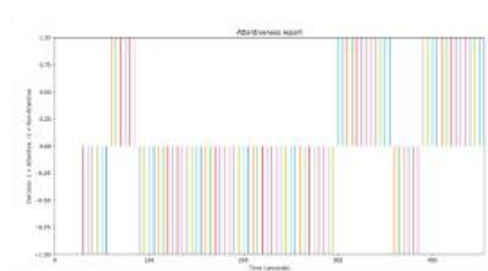


Figure15 – Graph for aggregated model over 400 seconds

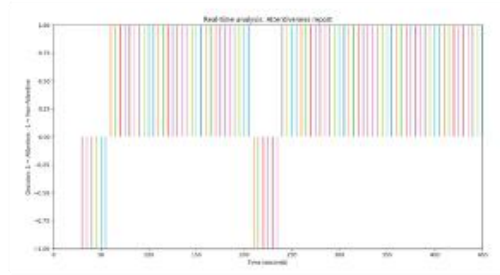


Figure16 – Graph for real-time analysis model over 400 seconds

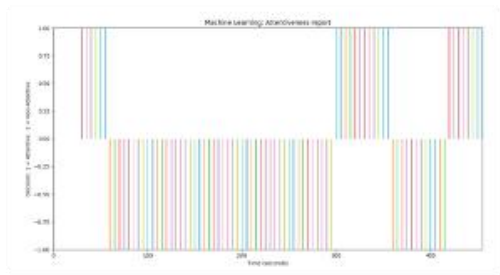


Figure17 – Graph for machine learning program over 400 seconds

Figure16’s data is based on a P.D.F., in which facial orientation, pupil orientation, and blink detection are assigned probabilities of 30%, 30% and 40%, respectively.

The video data from the 380th second is shown in Figure18. The student’s facial orientation and pupil gaze is concentrated down and to the left, away from the screen. In the moments before this image was taken, the student was adjusting her seating position, making faces and looking in different directions. From this, I can conclude that the student is not attentive.

Results from the different models:

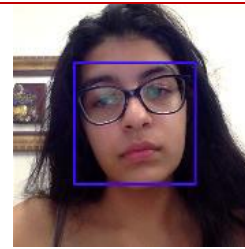


Figure18 – Image of ‘Non-Attentive’ student

	Model		
	Aggregated	Machine Learning	Real-time analysis
Output	‘Non-Attentive’	‘Non-Attentive’	‘Attentive’

Figure19 – Results from different models

In this instance, the machine learning model provides the correct output, clearly indicating that the real-time analysis is not sufficient to produce reliable results. However, the evaluation of the different facial features in the hard-coded algorithm is necessary as facial expressions alone do not always correlate to concentration levels.

Acknowledgments

We would like to thank Dr. Michelle Kelly, Assistant Professor at ECAE, who helped us analyse the data and design the framework for real-time analysis.

We would like to give a special mention to Ms. Khawla Barley, Head of Initiatives at Special Olympics UAE, who helped garner support for this project and organise meetings students and professors to collect data and conduct detailed analysis.

Bibliography

[1] Centres for Disease Control and Prevention. Autism and Developmental Disabilities Monitoring (ADDM) Network. [Online]. Available from <https://www.cdc.gov/ncbddd/autism/addm.html> [Accessed 2nd August 2021]

[2] Centres for Disease Control and Prevention. Treatment and Intervention Services for Autism Spectrum Disorder. [Online]. Available from <https://www.cdc.gov/ncbddd/autism/treatment.html> [Accessed 2nd August 2021]

[3] Leitner, Y. (2021). The Co-Occurrence of Autism and Attention Deficit Hyperactivity Disorder in Children – What Do We Know?. *Frontiers In Human Neuroscience*. <https://doi.org/10.3389/fnhum.2014.00268>

[4] Goldberg, P., Sümer, Ö., Stürmer, K., Wagner, W., Göllner, R., & Gerjets, P. et al. (2019). Attentive or Not? Toward a Machine Learning Approach to Assessing Students' Visible Engagement in Classroom Instruction. *Educational Psychology Review*, 33(1), 27-49. <https://doi.org/10.1007/s10648-019-09514-z> (Goldberg et al., 2019)

[5] Zaletelj, J., & Košir, A. (2017). Predicting students' attention in the classroom from Kinect facial and body features. *EURASIP Journal On Image And Video Processing*, 2017(1). <https://doi.org/10.1186/s13640-017-0228-8> (Zaletelj&Košir, 2017)

[6] Shekhar Pandey. Dlib 68 points Face landmark Detection with OpenCV and Python. Study tonight. Weblog. [Online]. Available

from <https://www.studytonight.com/post/dlib-68-points-face-landmark-detection-with-opencv-and-python>. [Accessed 10th July, 2021].

[7] Haarcascades. Opencv. GitHub. Weblog. [Online]. Available from <https://github.com/opencv/opencv/tree/master/data/haarcascades>. [Accessed 10th July, 2021].

[8] Sergio Canu. Eye Faze detection 1 – Gaze controlled keyboard with Python and Opencv p.3. pysource. Weblog. [Online]. Available from <https://pysource.com/2019/01/14/eye-gaze-detection-1-gaze-controlled-keyboard-with-python-and-opencv-p-3/>. [Accessed 10th July, 2021]

[9] ManasSambare. FER-2013 Learn facial expressions from an image. Kaggle. Weblog. [Online]. Available from <https://www.kaggle.com/msambare/fer2013>. [Accessed 10th July, 2021].

[10] Asadullah Dal. Distance measurement using single camera. GitHub. Weblog. [Online]. Available from https://github.com/Asadullah-Dal17/Distance_measurement_using_single_camera/. [Accessed 15th July, 2021].

Author Information

First Author:

Nanaki Singh

Student at Jumeirah English Speaking School, Dubai, UAE. Email: nanakising111@gmail.com

Second Author:

Mohan Kshirsagar

Senior Research and Innovation Engineer at On My Own Technology Private Limited Mumbai.

Third Author:

Reetu Jain

Chief-mentor and Founder of On My Own

Technology Private Limited Mumbai.

Email: reetu.jain@onmyowntechnology.com

Fourth Author:

Shekhar Jain

Chief Executive Officer and Co-Founder of On My

Own Technology Private Limited Mumbai.

Email: shekhar.jain@onmyowntechnology.com

*i*Journals