COMPUTER PERFORMANCE EVALUATION FOR VIRTUAL CLASSROOM WITH ARTIFICIAL INTELLIGENCE FEATURES

Kah Yee Lim[1], **Hau Joan**[1] **and Yiqi Tew**[1]

[1] Faculty of Computing and Information Technology, Tunku Abdul Rahman University College, Kampus Utama, Jalan Genting Kelang, 53300, Wilayah Persekutuan Kuala Lumpur, Malaysia
*Corresponding author: yiqi@tarc.edu.my

ABSTRACT

The advancement of computer technology allows students to interact with Artificial Intelligence (AI) through smart classrooms. Smart classroom is one of the latest technology-enhanced learning (TEL) which allows the classroom and students to interact during the learning process. Currently, smart classrooms are believed to change current dull teaching methods and enhance the students' learning experience. Therefore, the proposed paper is a comprehensive study of applying artificial intelligence features to an intelligent classroom system (a.k.a virtual classroom system) that provides face detection and hand gestures through e-learning classrooms. Artificial intelligence features will be implemented and compared on three machines with varying hardware specifications. According to the results of this study, Tensorflow Handpose provides more accuracy than MediaPipe Hands, although it requires higher hardware specifications. Face-api.js also outperforms TensorFlow and MediaPipe when it comes to executing face detection functions. According to this study, the present face and hand APIs can be adopted in smart classroom systems.

Keywords: Virtual Classroom, Google Meet, Face Detection, Hand Gesture Detection, Object Recognition

1.0 INTRODUCTION

Education plays a crucial role in the modern technological world, as it is an important tool for a better future. Through the rapid development of Internet technology and artificial intelligence (AI), virtual classrooms have been appointed for modern education in order to provide better teaching and learning services (Yu Fei et al., 2020). A virtual classroom is a virtual teaching environment where instructors and students can deliver course content, engage and interact with each other, and collaborate in groups online (Racheva, 2018). A virtual classroom differs from a traditional classroom in that it takes place in a real-time, synchronous environment. There are several software used to conduct virtual classrooms, such as zoom, google meet, microsoft teams, etc. While online education may typically involve viewing pre-recorded, asynchronous material, a virtual classroom setting involves live interaction between the lecturer and students (Rapanta et al., 2020). Through the research of other researchers (Olszewska, J. I., 2021), virtual classrooms that realize artificial intelligence have become a reality and can assist interactive education. Virtual classrooms have the benefit of collective intelligence; students can share what they find relevant and interesting to the particular concepts taught in the classroom. Again, participation in the classroom ultimately depends on the students, and what AI can ensure is to improve the chances of that happening. Artificial intelligence opens many creative doors for students and teachers alike. Students' work can be unconventional; demonstrating their abilities and knowledge beyond the prescribed books, which in turn makes them more confident in their work. Then, teachers can figure out each student's tendencies from a fairly young age (AIT Staff Writer, 2021). In addition, facial biometrics contribute to competitive authentication methods and advances while ensuring the reliability and validity of e-learning systems. To ensure the authenticity of users, the use of facial biometrics is recommended. This will provide an effective authentication method for learners and reduce the probability of cheating and other user authentication anomalies (J. Valera, 2015). In this paper we are going to study the efficiency of machine learning libraries for face detection and hand gesture detection in order to have proper guidance in future development of virtual classrooms.

2.0 BACKGROUND STUDIES

To facilitate the intelligent feature of virtual classrooms, several domains are studied and examined based on the features and feasibility of deployment. Face detection and gesture detection were considered in our study.

2.1 Face Detection

The distinctions between face detection and face recognition are frequently misunderstood.
Facial detection identifies face segments or areas from a picture, whereas face recognition identifies an individual's face based on personal information. Face detection and identification are advanced in today's culture, but they will encounter certain challenges throughout the way, (Howard, 2018). Table 1 is a list of the issues.

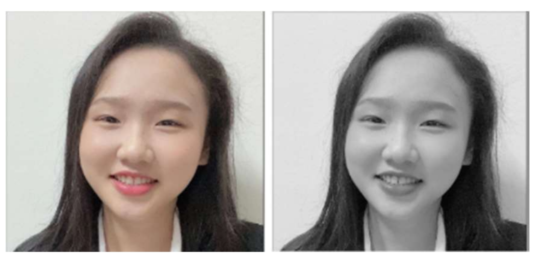
Table 1. Difficulties of Face Detection

Difficulties	Explanation
Background	Changes in the background and surrounding of the person in the image will influence the face detection accuracy.
Light Level	Various lighting environments reduce the ability to detect facial features.
Pose	The different angles of the captured facial images distort the face recognition process.
Expression	Changes in expressions cause changes in spatial relationships and changes in the shape of facial features
Occlusion	If there is a part of the face that is not observable, it will affect the performance and face recognition due to the not enough information provided.
Rotation, scaling and translation	Transformation of the image may distort the original information of the image.

The face detected from an image is suggested to crop out only the face for further processing in Singh's work (Singh et al., 2015). Any colored image will convert to grayscale for image pre-processing. Also, the face detected will then be aligned based on the eye's position and the scale of the image. Several publications by Akshara J. et al., Arun K. et al. and Chintalapati, S. et al. advocated using histogram equalisation to facial images and preprocessing the images by scaling (Akshara J. et al, 2017, Arun K. et al., 2017, Chintalapati, S. et al., 2013).

Pre-processing can improve the performance of the system (Howard, 2018). It is important for enhancing the accuracy of facial recognition. One of the required preparatory stages for processing the image's size is scaling. Due to the reduced number of pixels, scaling of images can increase the processing speed by reducing the system computation. The image’s size and pixels contain its unique spatial information. The spatial information is important since it provides a measurement of the image's least identifiable detail. As a result, spatial data must be treated with care to avoid picture distortion and tessellation effects. For normalization and standardization purposes, the dimensions of all images should be the same. The length and width of the image are preferred to be the same size based on the proposed Principal Component Analysis (PCA).

For pre-processing, colour photographs are commonly converted to greyscale images as shown in Figure 1. A grayscale image is commonly referred to as a black and white image, but the name emphasizes that such an image will also include many shades of gray. Grayscale images are considered to be less sensitive to lighting conditions and to calculate faster. A colour image is a 24-bit image with pixels ranging from 0 to 16777216, whereas a grayscale image is an image with 8-bit and pixels ranging from 0 to 255 (Howard, 2018). As a result, colour photographs demand more storage space and processing power than grayscale ones (Kanan and Cottrell, 2012). If the colour picture is not required for the computation, it is referred to as noise. Furthermore, preprocessing is required to improve the image's contrast. Histogram equalisation is one way of pre-processing to increase the image's contrast (Pratiksha M. Patel, 2016). It may decrease the effect of uneven lighting while providing a consistent intensity distribution on the horizontal axis of intensity.



(A) Coloured Image (B) Grayscale Image

Figure 1. Image convert from (A) to (B)

2.2 Hand Gesture Detection

Hand gesture can be parsed as one of the most natural and intuitive ways of communication between humans and machines, especially in the Human Computer Interaction (HCI) field, because it closely mimics the way of interaction between humans (Ren et al., 2011). In order to detect hand gestures, these processes must be passed through, that is, input images or frames through the sensor, execute the Application Programming Interface (API) for image processing, and finally display the returned results (Zhang et al., 2020). In these processes, efficient API has played a very important role in Hand Gesture Detection. Until now, a lot of Hand Gesture Detection APIs have been released by others, such as Tensorflow Handpose and MediaPipe Hands. These APIs have different architectures to process the input, which result in different accuracy and efficiency of Hand Gesture Detection.

3.0 PROPOSED METHOD

In our proposed work, the following proposed method will use Google Meet as the main platform for the virtual classroom. In addition, the following proposed method will be developed as a plugin for Google Meet through Google Chrome extension. Figure 2 shows the results of using hand detection and face detection in Google Meet Platform where the hand and face landmarks will be drawn while the feature is detected.

Face detection is used to take participants' attendance in a virtual classroom. There are few processes that will need to be follow as mentioned below:

a. Open the Google Meet and browse the "Face Recognition" function.
b. A HTML video will start playing, and the screen captured from the camera will be drawn using the canvas.
c. A face detection library will be executed immediately, and a facial landmark with face emotion will be displayed on the screen.
d. The data URL of the images shown in the video will be generated when the user captures the face.
e. The firebase storage will be used to store the face captured from the user site.
f. A total of five faces will be stored in the firebasedatabase and storage for further training.

Moreover, we include a hand gesture detection feature in our proposed virtual classroom that uses Tensorflow Handpose and Mediapipe Hands for capturing participants' activeness in the classroom. The overall process for Hand Gesture Detection are described as follow:

a. Once access to the Google Meet, the model for the Hand Gesture Detection will be loaded.
b. A HTML video element will be created to retrieve the user's local webcam stream by using captureStream() instead of using Google Meet's video source.
c. A HTML Canvas element will be created and assigned to the video stream captured in step 2 to display the results.
d. The captured video stream is passed to the API frame by frame and the result is generated.
e. The results obtained from the API will be drawn in the HTML Canvas element created in step 3.

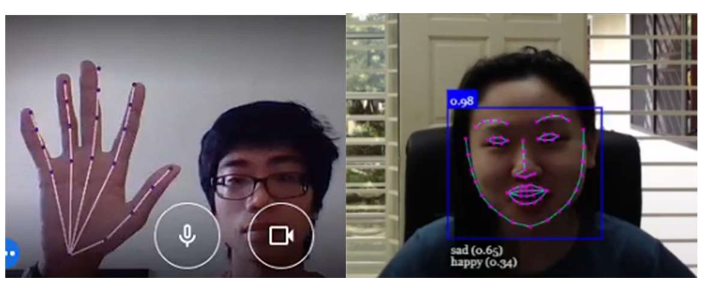


Figure 2. Snapshot of gesture handpose and face detection using proposed Google Meet API.

Our proposed work is working with the following hardware and software as shown in Table 2 for the development. In addition, we also used higher hardware specifications (i.e., machine C) as shown in Table 2 to evaluate the performance and the software used in the tested proposed solution. Besides, a 0.922 megapixel 1080P high-definition webcam is used in this research paper.

Table 2. Hardware and software specification for proposed work

Machine	Hardware Specification	Software Specification
A	Intel Core i5-8300H, 2.30GHz 16GB RAM GTX 1050	Python 3.6 with OpenCV2, Tensorflow, MediaPipe, Visual Studio Code with (HTML, CSS, JSON, JS) Google Firebase Google Meet
B	Intel Core i7-7700HQ, 2.8GHz 24GB RAM GTX 1050	
C	Intel Core i9-9900KF, 3.60GHz 128GB RAM GTX 2080	

4.0 RESULT AND DISCUSSION

Results on libraries' efficiency used for face detection and hand gesture detection are collected. For hand gesture detection, results of each Tensorflow and MediaPipe models in detecting hand landmarks are collected. In order to ensure the consistency of the generated results from the same API, we use a series of recorded videos with the same hand gesture movement as a baseline video. We have implemented a frame per second (FPS) counter in the code itself instead of using Google Chrome's default FPS meter to achieve a more reliable FPS. In addition, we use the confidence provided by API and counting to calculate the accuracy of the model's recognition of Hand Landmark in the recorded video.

The results of time taken for the face detection in three different libraries which include face-api.js, Tensorflow and MediaPipe are collected and discussed at the section below. For the comparison of face detection between the three different libraries, an image video is used for gathering the results of the time taken of face detection for each library in both machine B and C.

In machine A, the model load time of Tensorflow Handpose and MediaPipe Hands are collected, as shown in Figure 3. The model is loaded for 10 times and its average value is calculated. It is analyzed that Tensorflow Handpose (TFJS) (i.e., in blue line) requires more time to load the model in the Google Meet compared to MediaPipe Hands. MediaPipe Hands shows faster performance with 333.5 times faster than TFJS. In Table 3, the backend library, FPS, confidence level and accuracy of the model used in detecting the hand of the user are shown. Under the lower hardware specification requirements (i.e., machine A), the performance of MediaPipe Hands has a slightly higher FPS than Tensorflow Handpose. Nevertheless, the FPS of Tensorflow Handpose has higher average accuracy than MediaPipe Hands.

Based on Figure 4, it is analyzed that Tensorflow Handpose requires more time to load the model in the Google Meet compared to MediaPipe Hands, but higher hardware specifications (i.e., machine C) can reduce the time required to load Tensorflow Handpose. Based on Table 3, it shows the backend, FPS and accuracy of the model in detecting the hand of the user. It can be seen from the figure that under the higher hardware specification requirements (i.e., machine C), MediaPipe Hands has a higher FPS than Tensorflow Handpose. Although the FPS of Tensorflow Handpose is about five FPS lower, it is more accurate than MediaPipe Hands.

Table 3. Comparison of Tensorflow Handpose and MediaPipe Hands

Model	**TensorFlow - Handpose**						**MediaPipe - Hands**					
Backend	Web Graphics Library (WebGL)						WebAssembly (Wasm)					
Machine	A			C			A			C		
FPS	~ 23 FPS			~ 55 FPS			~50 FPS			~60 FPS		
Total Confidence from API	590	512	364	1733	1711	1719	989	1156	998	1904	1883	1891
Total Count Allocated	595	516	370	1744	1782	1732	1041	1209	1046	1995	1973	1985
Accuracy	99.22	99.23	98.43	99.39	99.41	99.30	94.99	95.64	95.41	95.43	95.44	95.27
Average Accuracy	98.96			99.37			95.35			95.38		

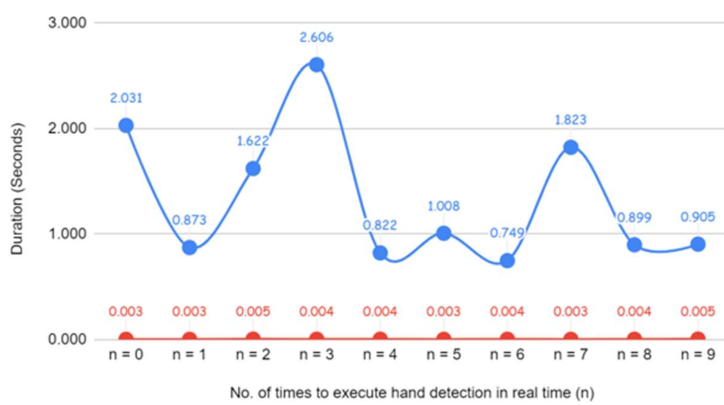


Figure 3. Results of TFJS Model Handpose (Blue) vs MediaPipe Handpose (Red) detection in Machine A.

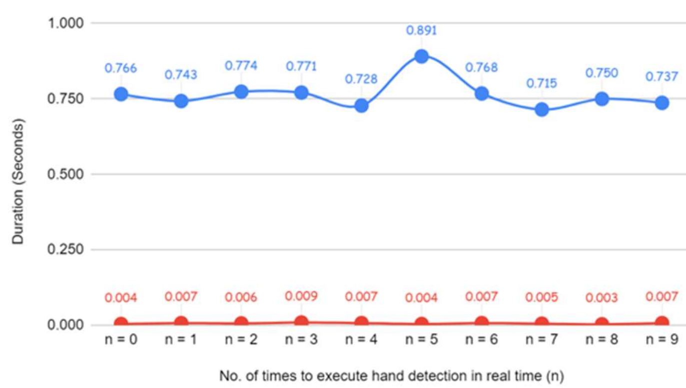


Figure 4. Results of TFJS Model Handpose (Blue) vs MediaPipe Handpose (Red) detection in Machine C.

Based on Figure 5, it is analysed that MediaPipe requires more time to execute the face detection compared to others. The performance for the libraries to execute the face detection can be improved by using a machine with higher specification. The same step is carried out for gathering the execution time of face detection in machine B where each of the libraries will run for 10 times and an average time taken for the face detection is calculated. By using machine B, the performance of the face-api.js is 1.78 times and 2.14 times better than TensorFlow and MediaPipe respectively. Due to the lower hardware specification of machine B, the time taken among each of the libraries provides a bigger gap compared to the same libraries running in machine C. A performance analysis of executing different libraries in machine C is illustrated in Figure 6.

Figure 6 illustrates the time taken for face detection among different types of libraries used in machine C. In order to calculate the average time taken for each of the libraries used, a face detection program with each of the libraries is executed 10 times. Based on the figure, MediaPipe provides the highest time taken for face detection execution which represents that MediaPipe has the worst performance among these libraries. In addition, the sequence of performance from high to low of the libraries for face detection is face-api.js, TensorFlow and MediaPipe. Face-api.js shows the least execution time for face detection where it is 1.89 times and 1.76 times faster than the MediaPipe and TensorFlow respectively. Besides, the performance for different libraries is affected by the hardware limitations where the performance for the libraries in machine C is better than in machine B.

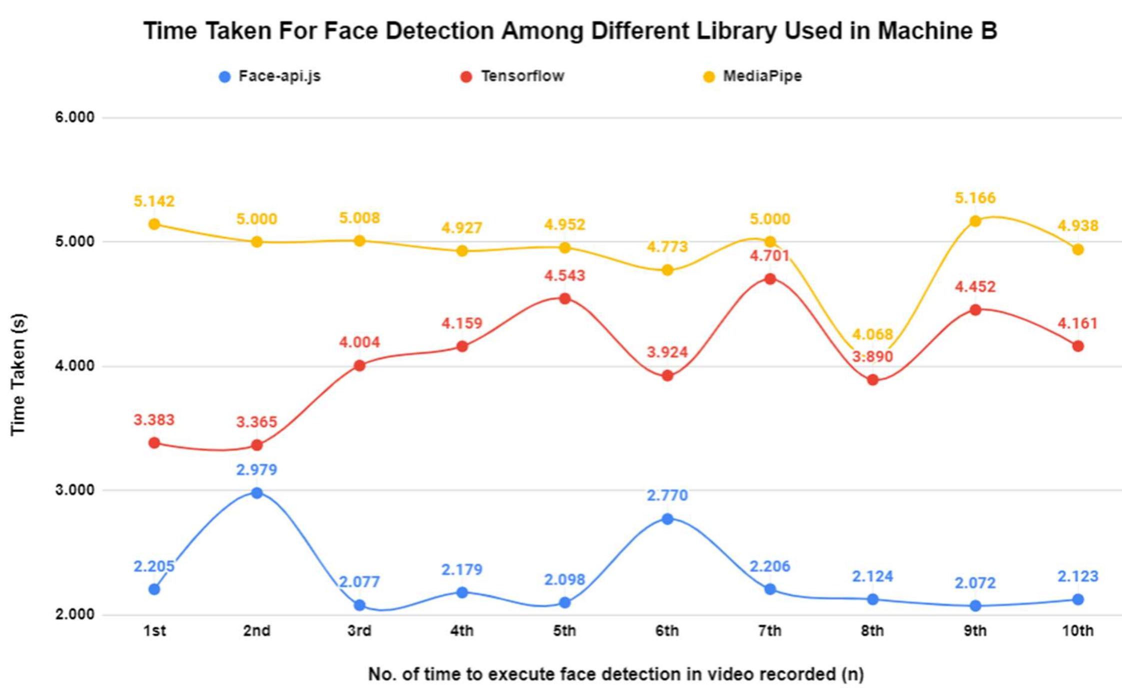


Figure 5. Results of Face-api.js, Tensorflow and MediaPipe Face Detection in Machine B

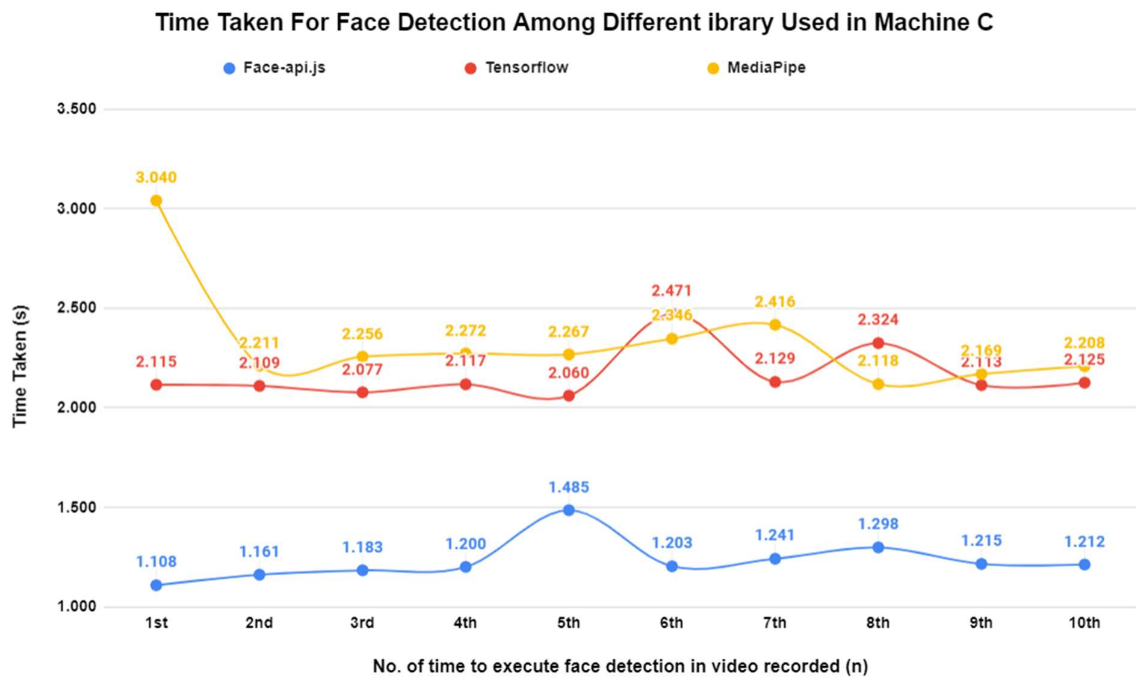


Figure 6. Results of Face-api.js, Tensorflow and MediaPipe Face Detection in Machine C

5.0 CONCLUSION

In conclusion, through the analysis of Tensorflow Handpose and MediaPipe Hands, in terms of accuracy, Tensorflow Handpose is higher than MediaPipe Hands. The accuracy of Tensorflow Handpose is as high as 99.4%, while the accuracy of MediaPipe Hands is around 95.4%. However, in terms of FPS, MediaPipe Hands is more stable in terms of lower hardware specification requirements (i.e., machine A) or higher hardware specification requirements (i.e., machine C) than Tensorflow Handpose. Through the research on hand gesture detection, in order to have high efficiency on low-specification hardware, it is recommended to use MediaPipe Hands, on the contrary, Tensorflow Handpose is more suitable for high-specification hardware.

In addition, the face-api.js provides the best performance in executing the face detection function compared to TensorFlow and MediaPipe, therefore the library is used for implementing the face detection in a virtual classroom system. Although there is a difference of average time execution for face detection among each of the libraries in different machines, each of the libraries in different machines shows similar trends where the face-api.js provides the highest performance compared to others. According to the research, face-api.js is recommended for high efficiency on low specification hardware, whereas MediaPipe is more suitable for high specification hardware.

The performance of the libraries may be affected based on the programming languages, therefore a comparison for the performance of the libraries in different programming languages such as python versus javascript can be carried out in the future. Furthermore, in the future, in addition to detecting hand landmarks, another study can be proposed to enhance hand detection in virtual classroom systems by implementing hand gesture detection and recognition in virtual classroom systems. Besides, there are many face detection libraries which can be used for implementing the project. Hence, the better library can be used for analysis and implemented into the virtual classroom system in order to improve the performance of the system in the future.

6.0 ACKNOWLEDGMENTS

With our sincerest greetings, we would like to express our gratitude to all those who provided great help, moral assistance and cooperation to our research team. We are grateful to the Faculty of Computing and Information Technology (FOCS) of Tunku Abdul Rahman University College (TARUC) for providing the facilities at the Center of Computational Intelligence (Center of Computational Intelligence, 2020) to make the project successfully implemented with preliminary results, as shown in this paper.

REFERENCES

AIT Staff Writer 2021, *Role of Artificial Intelligence in Online Education*, AiThority, viewed 20 September 2021, <https://aithority.com/technology/education-and-research/role-of-artificial-intelligence-in-online-education/>.

Akshara J., Akshay J. T. L., Krishna Y. (2017), Automated attendance system using face recognition, in ‘International Research Journal of Engineering and Technology (IRJET)’, Vol. 4, pp. 1467–1471.

Arun K., Sudesh V. K., Amar. P. Z., Nikhil. D. B. & Chetan. J. B. (2017), Attendance system using face recognition and class monitoring system, in ‘International Journal on Recent and Innovation Trends in Computing and Communication’, Vol. 5, pp. 273–276.

Center of Computational Intelligence (2020), Tunku Abdul Rahman Universiti College, viewed 10 Sept 2021,<https://www.tarc.edu.my/focs/research/cci/>.

Chintalapati, S., Raghunadh, M. V. (2013), Attendance system using face recognition and class monitoring system, in ‘International Journal on Recent and Innovation Trends in Computing and Communication’.

Howard, C. (2018), ‘Face recognition based automated student system’. Bachelor of Engineering (Hons) Electronic Engineering.

J. Valera, J. Valera and Y. Gelogo, 2015, "A Review on Facial Recognition for Online Learning Authentication," 2015 8th International Conference on Bio-Science and Bio-Technology (BSBT), pp. 16-19, doi: 10.1109/BSBT.2015.15.

Kanan, C. and Cottrell, G., 2012. Color-to-Grayscale: Does the Method Matter in Image Recognition?. PLoS ONE, vol. 7, no. 1, pp. 29740.

Olszewska, J. I. (2021), The virtual classroom: A new cyber physical system, in ‘2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)’, pp. 000187-000192.

Pratiksha M. Patel, 2016, Contrast Enhancement of Images and videos using Histogram Equalization. International Journal on Recent and Innovation Trends in Computing and Communication, vol. 4, no. 11.

Racheva, V 2018, *What is a virtual classroom*, VEDAMO, Vedamo, viewed 20 September 2021, <https://www.vedamo.com/knowledge/what-is-virtual-classroom/>.

Rapanta, C, Botturi, L, Goodyear, P, Guàrdia, L & Koole, M 2020, ‘Online University Teaching During and After the Covid-19 Crisis: Refocusing Teacher Presence and Learning Activity’, *Postdigital Science and Education*, vol. 2, no. 3, pp. 923–945, viewed 20 September 2021, <https://link.springer.com/article/10.1007/s42438-020-00155-y>.

Ren, Z., Meng, J., Yuan, J. & Zhang, Z. (2011), Robust hand gesture recognition with kinect sensor, pp. 759–760.

Singh, S., Rastogi, R. & Sharma, P. S. (2015), Automatic lecture attendance system using face recognition, in 'MATRIX Academic International Online Journal of Engineering and Technology', Vol. III, pp. 36–40.

Yufei, L, Saleh, S, Jiahui, H, Mohamad, S & Abdullah, S, 2020, 'Review of the Application of Artificial Intelligence in Education', *International Journal of Innovation, Creativity and Change. www.ijicc.net*, vol. 12, no. 8, p. 548 - 555, viewed 20 September 2021, <https://www.ijicc.net/images/vol12/iss8/12850_Yufei_2020_E_R.pdf>.

Zhang, F., Bazarevsky, V., Vakunov, A., Thachenka, A., Sung, G., Chang, C.-L. & Grundmann, M. (2020), 'Mediapipe hands: On-device real time hand tracking'.