EVALUATION OF VIRTUAL CLASSROOM WITH ARTIFICIAL INTELLIGENCE COMPONENTS

(Date received: 11.01.2022/Date accepted: 18.04.2022)

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

[1] Faculty of Computing and Information Technology,
Tunku Abdul Rahman University College, 53300 Kuala Lumpur, Malaysia

**Corresponding author: yiqi@tarc.edu.my*

ABSTRACT

The recent advancement of information technology allows educators and students to interact with Artificial Intelligence (AI) through smart classroom channels. This channel is one of the latest technology-enhanced learning (TEL) that provides a learning environment with educators and students interaction during the learning process. Currently, smart classrooms are believed to change current dull teaching methods and enhance the students' learning experience. Hence, this paper shows a comprehensive investigation of applying AI components to an intelligent classroom system (a.k.a virtual classroom system) that provides hand gestures and face detection through e-learning classrooms. Machine Learning libraries are implemented and compared on three machines with varying hardware specifications and capabilities. As a result of this study, Tensorflow Handpose provides more accuracy than MediaPipe Hands, although it requires higher computational capabilities. Face-api.js outperforms TensorFlow and MediaPipe when it comes to executing face detection functions. In addition to the study, the presented face and hand APIs can be adopted in a real time implementation for smart classroom systems.

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

1.0 INTRODUCTION

Classroom plays an imperative role in the modern technological world, as it is an important growing environment for one's better future. Through the rapid development of AI and Internet technology, virtual classrooms have been utilized for modern education in order to provide better teaching and learning services [1]. A virtual classroom differs from a traditional classroom in that it takes place in real-time and is synchronized. There are several software used to conduct virtual classrooms, such as Zoom, Google Meet, Microsoft Teams, etc. Although online education often entails viewing pre-recorded asynchronous content, virtual classroom environments entail real-time interaction between lecturers and students [2]. Through the investigation by other researchers [3], virtual classrooms that realize AI 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. AI opens many creative doors for students and teachers alike. Students' work can be unconventional; demonstrating their abilities and knowledge beyond the prescribed books, that in turn makes them more confident in their work. Then, teachers can figure out each student's tendencies from a fairly young age [4]. 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 [5]. 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

Several domains are studied and examined based on the components and feasibility of deployment to facilitate the intelligent components of virtual classrooms. These include face detection and hand gesture detection.

2.1 Face Detection

Face detection and face recognition are frequently misunderstood by most of the users. 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 [6]. Table 1 shows a list of the issues.

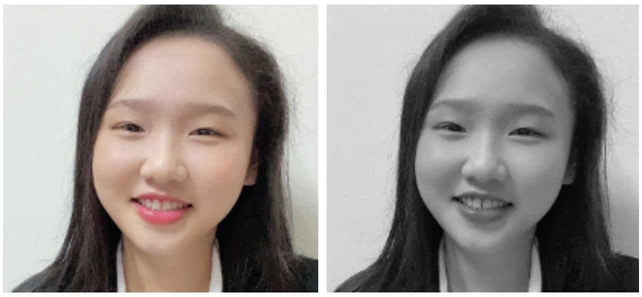
In each image, the face is detected and cropped out for further processing in Singh's work [7]. Any colored image will convert to grayscale for image pre-processing. Also, the face detected will then be aligned based on the eye's position and the scale of the image. Several publications by Akshara J. *et al.,* Arun K. *et al*. and Chintalapati, S. *et al*. advocated using histogram equalisation to facial images and preprocessing the images by scaling [8, 9, 10].

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.

In order to improve the system performance, pre-processing of input images is required [6]. 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).

In the pre-processing stage, photographs in color mode are commonly converted to grayscale mode, 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 [6]. As a result, colour photographs demand more storage space and processing power than grayscale ones [11]. 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 [12]. 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 Face Recognition

Facial recognition identifies or verifies an individual from either image, video frame or real-time. It is commonly used as access control in security systems and can be compared to other biometric techniques such as fingerprint or eye iris recognition systems [13]. The face recognition will capture the face from a group of faces, and then it will identify the details of the face. After that, it will match the faces available in the current storage location, and it will find the exact face that corresponds with the current face in the database. The process of face recognition is shown in Table 2.

Table 2: Process of Face Recognition

Process	Explanation
Face Detection	Face detection is a process of detecting the face that is located in a frame or image by finding the landmarks of the face such as eyes and nose.
Face Alignment	Face alignment plays an important role during the image pre-processing. The face and eye regions are automatically detected and faces are aligned according to translation, scaling and rotation. The face alignment is essential in image pre-processing as it is able to make the face detected more readable for the system to track the features.
Features Extraction	The facial recognition will be producing 128 dimensions embeddings once the pre-processed face is ready. The face that passes into the system, the system will check the face and generate 128 numbered measurements from the image in order to let the computer read the image as it cannot actually observe the complete visual appearance of the image. The number generated is used by the computer for comparison.
Face Recognition or Classification	The system will compare the measurements that are obtained from the input image to what is already in the database. After that, the score for each and every match will be generated, if the score is more than a particular threshold, then it is considered as a match face.

2.3 Hand Gesture Detection

Hand gesture can be described 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 [14]. 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 [15]. 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 that result in different accuracy and efficiency of Hand Gesture Detection.

3.0 PROPOSED METHOD

We use Google Meet as the main platform for the virtual classroom. Plugins are required for our prototype development using Google Meet through Google Chrome extension. Figure 2 shows the results of using hand gesture detection and face detection in Google Meet Platform where the hand and face landmarks will be drawn while the feature is detected.

Figure 2: Snapshot of hand gesture and face detection in Google Meet

Our experimental work utilizes the following hardware and software as shown in Table 3 for realizing the virtual classroom prototypes. For instance, a higher hardware specifications machine, i.e., machine C, as shown in Table 3 is utilized to evaluate the prototype performance and the software used in the tested proposed solution. In addition, a 0.922 megapixel 1080P high-definition webcam is used in this work.

Table 3: 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, RTX 2080	

4.0 RESULT AND DISCUSSION

We collected the results on libraries' efficiency used for face detection and hand gesture detection. For hand gesture detection, results of each Tensorflow and MediaPipe models in detecting hand landmarks are recorded. 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 counts to calculate the accuracy of the model's recognition of Hand Landmark in the recorded video.

The model load time of Tensorflow Handpose and MediaPipe Hands are collected in machine A, as shown in Figure 3. The model is loaded for 10 times and its average value is calculated. It is analysed 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, 333.5 times faster than TFJS, because it requires very less Graphic Processing Unit (GPU) computing power to execute. However, higher hardware specifications, i.e., machine C, can reduce the time required to load Tensorflow Handpose, as shown in Figure 4.

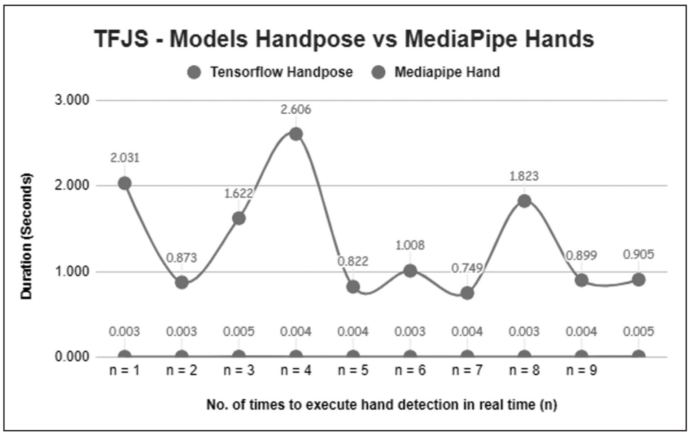


Figure 3: Results of TFJS Handpose vs MediaPipe Hands in Machine A

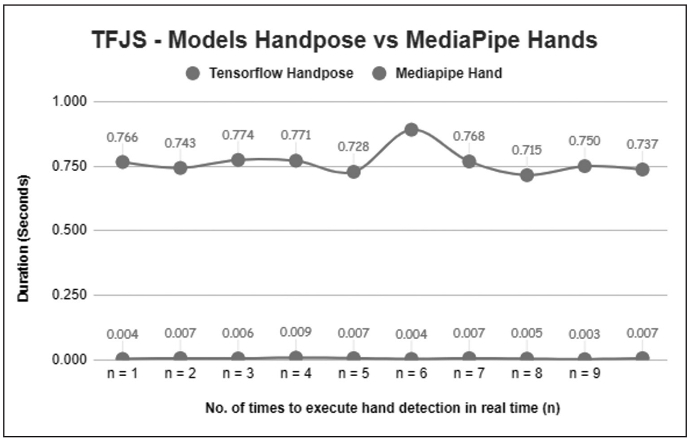


Figure 4: Results of TFJS Handpose vs MediaPipe Hands in Machine C

Table 4 shows the backend library, FPS, confidence level and accuracy of the model used in detecting the hand of the user. Under the low and high hardware specification requirements, i.e., machine A and C, both of them showed that the performance of MediaPipe Hands has a slightly higher FPS than Tensorflow Handpose due to less computational resources required for execution. Nevertheless, due to GPU support, the average accuracy of Tensorflow Handpose is higher than MediaPipe Hands.

The results of time taken for the face detection in three different libraries that 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 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 library 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 that 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 are affected by the hardware limitations where the performance for the libraries in machine C is better than in machine B.

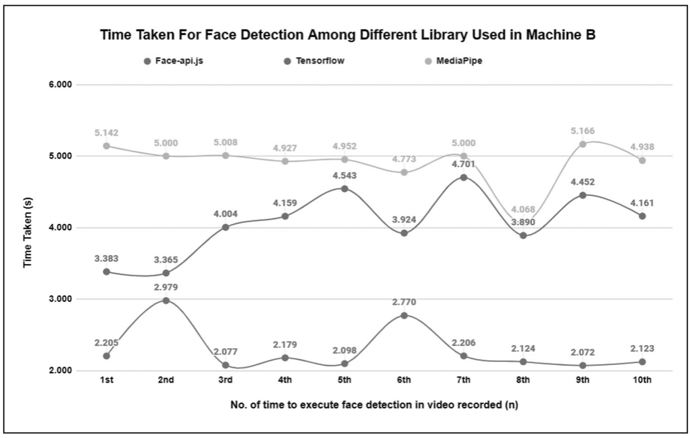


Figure 5: Results of Face-api.js, Tensorflow and MediaPipe Face Detection in Machine B

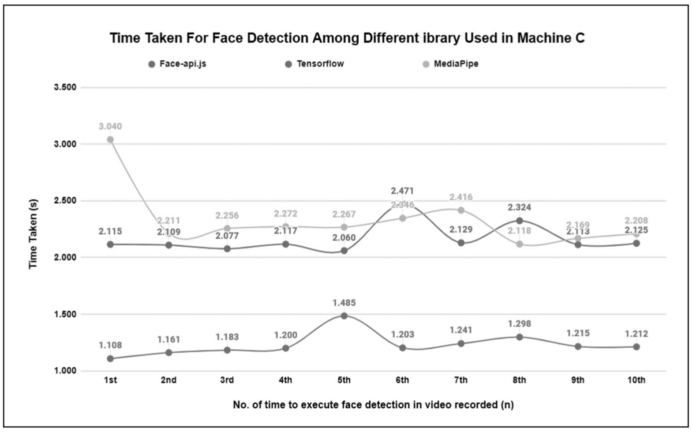


Figure 6: Results of Face-api.js, Tensorflow and MediaPipe Face Detection in Machine C

5.0 CONCLUSION

In a nutshell, through the analysis of Tensorflow Handpose and MediaPipe Hands, in terms of accuracy, Tensorflow Handpose (~99.4%) is higher than MediaPipe Hands (~95.4%). However, in terms of FPS, MediaPipe Hands is more stable in both low and high hardware specification requirements (i.e., machine A and machine C) than Tensorflow Handpose. Through the research on hand gesture detection, in order to have high efficiency on

Table 4: Comparison of Tensorflow Handpose and MediaPipe Hands

Model		TensorFlow - Handpose						MideaPipe - Hands				
Backend		Web Graphics Library (WebGL)						Web Assembly (Wasm)				
Machines		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	

low-specification hardware, it is recommended to use MediaPipe Hands. On the contrary, Tensorflow Handpose is more suitable for high-specification hardware.

Moreover, 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 addition to detecting face and hand landmarks, other AI components such as face emotion, liveness detection can be examined and included in virtual classroom systems. Besides, there are enormous face detection library resources that are able to implement the AI work. Nevertheless, each resource may come with different compatibility with recent hardware and updated version software tools and requires further analysis and investigation in order to improve the performance of the virtual classroom 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 (TAR UC) for providing the facilities at the Centre of Computational Intelligence [16] and Centre of Excellence Big Data Analytics and AI to make the project successfully implemented with preliminary results, as shown in this paper. ■

REFERENCES

[1] Yu Fei, 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>.

[2] 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>.

[3] 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. 187-192.

[4] 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/>.

[5] 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.

[6] Howard, C. (2018), ‘Face recognition based automated student system’, Universiti Tunku Abdul Rahman.

[7] 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.

[8] 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.

[9] 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.

[10] 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’.

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

[12] 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.

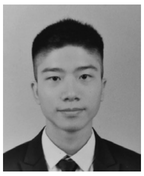
[13] Virgil Petrescu, R. V., 2019, Face Recognition as a Biometric Application. Journal of Mechatronics and Robotics, vol. 3, no. 1, pp. 237–257. <https://doi.org/10.3844/jmrsp.2019.237.257>.

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

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

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

PROFILES



KAH YEE LIM is a student of Faculty of Computing and Information Technology, Tunku Abdul Rahman University College, Kuala Lumpur. He received his Bachelor's Degree from Tunku Abdul Rahman University College in 2022. His areas of interest are web application development, game and graphic design.
Email address: limky-wm18@student.tarc.edu.my

HAU JOAN is a student of Faculty of Computing and Information Technology, Tunku Abdul Rahman University College, Kuala Lumpur. She received her BS degree from Tunku Abdul Rahman University College in 2022. In 2022, she joined the Faculty of Computing and Information Technology to pursue her MS degree at Tunku Abdul Rahman University College, Kuala Lumpur. Her research interests are video compression and video coding standard.
Email address: joanh-wm18@student.tarc.edu.my

YIQI TEW obtained his B.Eng. (Hons.) electronics degree from Multimedia University, MComSc. from National University of Malaysia, and PhD degree from University of Malaya in 2018, 2011 and 2016 respectively. He is an experienced Associate Professor and skilled engineer in Embedded Systems, Video (HEVC) Processing, Security (Information Hiding). His research involves Multi-view, Depth Sensing and Real-time Video Streaming Mechanisms in Industry 4.0 and Smart Agricultural projects. With his profession, he became the lead researcher in Computational Intelligence with publication of more than 30 papers, appointed as National Expert in IoT and Cloud Computing by the Ministry of Human Resource, Malaysia for WorldSkills events in 2018, 2020 and 2021.
Email address: yiqi@tarc.edu.my