INTERACTIVE DASHBOARD WITH VISUAL SENSING AND FAST REACTIVITY

(Date received: 13.01.2022/Date accepted: 13.04.2022)

Wen Lin Yong[1], Jun-Kit Chaw[2]*, Yiqi Tew[1]

[1] *Faculty of Computing and Information Technology, Kuala Lumpur Main Campus, Jalan Genting Kelang, Setapak, 53300 Kuala Lumpur, Malaysia.*

[1] *Institute of IR4.0, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia.*

**Corresponding author: chawjk@ukm.edu.my*

ABSTRACT

These days, technology is growing rapidly, and the market has been introduced with lots of fascinating ways to interact with computers. The advancement of deep learning models and hardware technology also enables more applications with fancy features to be built. The importance of hand gesture recognition has increased due to the prevalence of touchless applications. However, developing an efficient recognition system needs to overcome the challenges of hand segmentation, local hand shape representation, global body configuration representation, and a gesture sequence model. This paper proposed an interactive dashboard that could react to hand gestures. This is also an initiative of the Tunku Abdul Rahman University College (TAR UC) Smart Campus project. Deep learning models were investigated in this research and the optimal model was selected for the dashboard. In addition, 20BN Jester Dataset was used for the dashboard development. To set up a more user-friendly dashboard, the data communication stream between the captured input stream and commands among the devices were studied. As to achieve higher responsiveness from the dashboard, evaluation on data communication protocols which were used to pass the input data were included in the study.

Keywords: Computer Vision, Human-Computer Interaction (HCI), Gesture Detection, Real-time systems, Feature Extraction

1.0 INTRODUCTION

Human-Computer Interaction (HCI) is a field of study that focuses on discovering ways of human interacting with computers. Due to the rapid growth in computer technology, touchscreen HCI has evolved in recent years and over the pandemic to interaction with gestures, motion sensors, hand interaction and other touchless interfaces. One of the well-known practices is the visual-based HCI which requires cameras as the input devices instead of computer mouse and keyboard. Although this field has been broadly studied in the computer vision field, the methods proposed in the market show vulnerability in the outdoor environment. A review done by Chakraborty *et al.* suggested that existing classifiers for vision-based gesture identification are unable to handle all types of gesture classification problems at the same time. Each has disadvantages that limit total performance [1].

The development of the interactive dashboard in this research aimed to achieve high performance in responding to user's commands which were performed using gestures movements. Thus, the gesture recognition algorithms were studied and evaluated to achieve real-time capability in returning outcomes for the gesture movements detected. Aside from that, to obtain higher usability for the interactive dashboard, the video streaming protocols were determined for faster transmission of gesture data captured from the input devices.

Every individual in this world is known to act differently in their way. Even though a lot of defined classes and algorithms for human recognition systems have been introduced in the market, there might be chances of miscalculating poses while employing the practices for their use. To fine-tune the best gesture weights, a large gesture dataset maybe required. Besides, gesture recognition performance might be affected while the system is implemented in a real-world environment due to external factors such as visual occlusions [2] and illumination [3].

The user experience while making use of the dashboard might be affected by the sensitivity of the interactive dashboard. One of the factors that might affect the interactive dashboard irritability is the transmission of gesture videos data through the IP cameras and network video recorders. While there is a delay in the video transmission, the system might not achieve responsive controls towards the dashboard. Moreover, the model used to process the gesture would also affect the interactive dashboard reactivity [4].

2.0 RELATED WORKS

2.1 Gesture Recognition with Deep Learning Models

Gesture is a widely used touchless interface that allows humans to engage with technology, and it is the next step in the evolution of motion sensors [5]. Gesture-based touchless interfaces or better real-time hand interaction are possible with devices like Leapmotion, kinect, Azure Kinect. For example, using wired gloves as input devices is known to perform well in gesture recognition. However, data collection always requires the gloves as extra equipment that caused inconvenience due to wired restrictions [6].

In this paper, RGB cameras were used as the input devices to capture the motions. Hence, only deep learning-based development are discussed here as it could extract discriminative features for classification from RGB images more effectively. Although there are other methods that may perform better in terms of preciseness or pace when detecting the gestures, deep learning approaches are more robust in the real-world environment [7]. We considered two deep learning models to be utilized for detecting gestures: Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) approaches. CNN is recognized to make use of object recognition with the ability to learn features in spatial data. By using the CNN model, gesture recognition can be performed in a fast behaviour with high precision in the results [8]. However, CNN models are incapable of learning temporal data which is crucial in continuous hand gesture recognition. Hence, 3DCNN is proposed to acquire features extraction on 3D data (eg. gesture movements' videos data) which allows detecting dynamic gestures movements. Besides, recurrent neural networks (RNN) which are known for processing sequential data are taken into consideration to incorporate with the CNN model to process longer continuous gesture data [9]. By employing RNN within the CNN model, the gestures could be captured in a continuous manner for the model to predict the next actions done by the users. However, it is challenging for this approach to correctly recognize directional motions like swipe and rotation, as well as motions like push that changes spatial information over time [10]. To maintain information in memory for long periods of time, LSTM is preferred over RNN [11]. Thus, this paper integrated 3DCNN and LTSM to recognize gestures that are commonly used in HCI.

2.2 Data Communication Protocols

Modern communication systems and networks, such as the Internet of Things (IoT) and cellular networks, generate enormous amounts of heterogeneous traffic data [12]. As stated by Gaurav Sinha *et al.*, high bandwidth interaction allows the acceleration of the interaction between human and computer due to its capability to transfer massive data [13]. Either analog-based solutions or digital-based solutions could be used to acquire data for gesture recognition. Analog-based solutions are also known as closed-circuit televisions (CCTV) that have to process the analog signals in a digital video recorder (DVR) for video recording purposes. On the contrary, digital-based solutions use Internet protocol cameras (IP cameras) to transmit digitalized surveillance video to back-end computers for monitoring by using IP-based protocols such as RTP (Real-time Transport Protocol) or HTTP (Hyper-Text Transfer Protocol). For recording, it uses network video recorders (NVR) [14].

Due to the convenience and high resolution of video surveillance data, IP cameras were commonly used for machine learning or deep learning approaches [15], [16]. Together with other data-driven tasks, such as machine learning, it could unlock the potential of big data in many domains in the artificial intelligence era [17]. Furthermore, the goal of improving accuracy and efficiency to better promote the use of data-driven computation has remained unchanged. This paper aims to increase efficiency by utilizing the correct data communication protocol to produce high satisfactory HCI capability to the users. In conjunction with this, the responsiveness of the dashboard could be improved.

3.0 RESEARCH DESIGN AND METHODS

IP cameras were adopted as the input devices to capture the gestures from the users, then the captured data were forwarded to the dashboard for further processing and response. Our scope of work covers the exploration of the video streaming protocols used to allow communication (eg. passing media/files) between devices and the employment of the deep learning model for gesture recognition that was used to allow direct human interaction and controls to the dashboard. This work aims to develop an interactive dashboard with gesture-navigated effects based on artificial intelligence human gestures detection which brings great convenience, flexibility, and efficiency by meeting the benchmarks listed below:

1. **To attain high accuracy results in gesture recognition results.**
   The gesture recognition algorithms and methods applied should be able to achieve high precision results to secure the usability of the interactive dashboard. By acquiring exact gesture recognition, the dashboard will be able to react with proper responses according to what users perform.
2. **To achieve high responsiveness from the interactive dashboard.**
   a. The video streaming protocols which have been employed should be able to transmit the gesture data at a higher speed to reduce the delay of video streaming to the system for gesture recognition processing.
   b. The suitable deep learning models used to perform the gesture recognition should be implemented to accomplish the real-time capability of the interactive dashboard.

Figure 1 illustrates the program flow for the interaction dashboard developed. The aim of this paper is to detect the gestures for direct interaction between humans and the dashboard. Thus, gestures such as swipe left, right, up, and down were learnt by the model to interact with the dashboard. Since the gestures might be performed differently by every individual, the 20BN Jester dataset with a large amount of significant gesture data were utilized in the model learning process.

Figure 2 shows the overall process of video retrieval in our proposed video streaming module. To capture human gesture movements, IP cameras and NVRs were employed to capture the gestures that were performed by each individual. While human gesture recognition was performed in a real-time manner, the gesture data captured by those devices were expected to be transferred between those devices as quickly as possible to bring out a responsive and low latency interaction between the individuals and the dashboard.

A convolutional HIKvision's IP cameras and Network Video Recorders (NVR) were used in this research. As HIKvision's network cameras provide predefined video streaming protocols, specific settings could be configured for video streaming. The protocols provided by the HIKvision products are TCP, UDP,

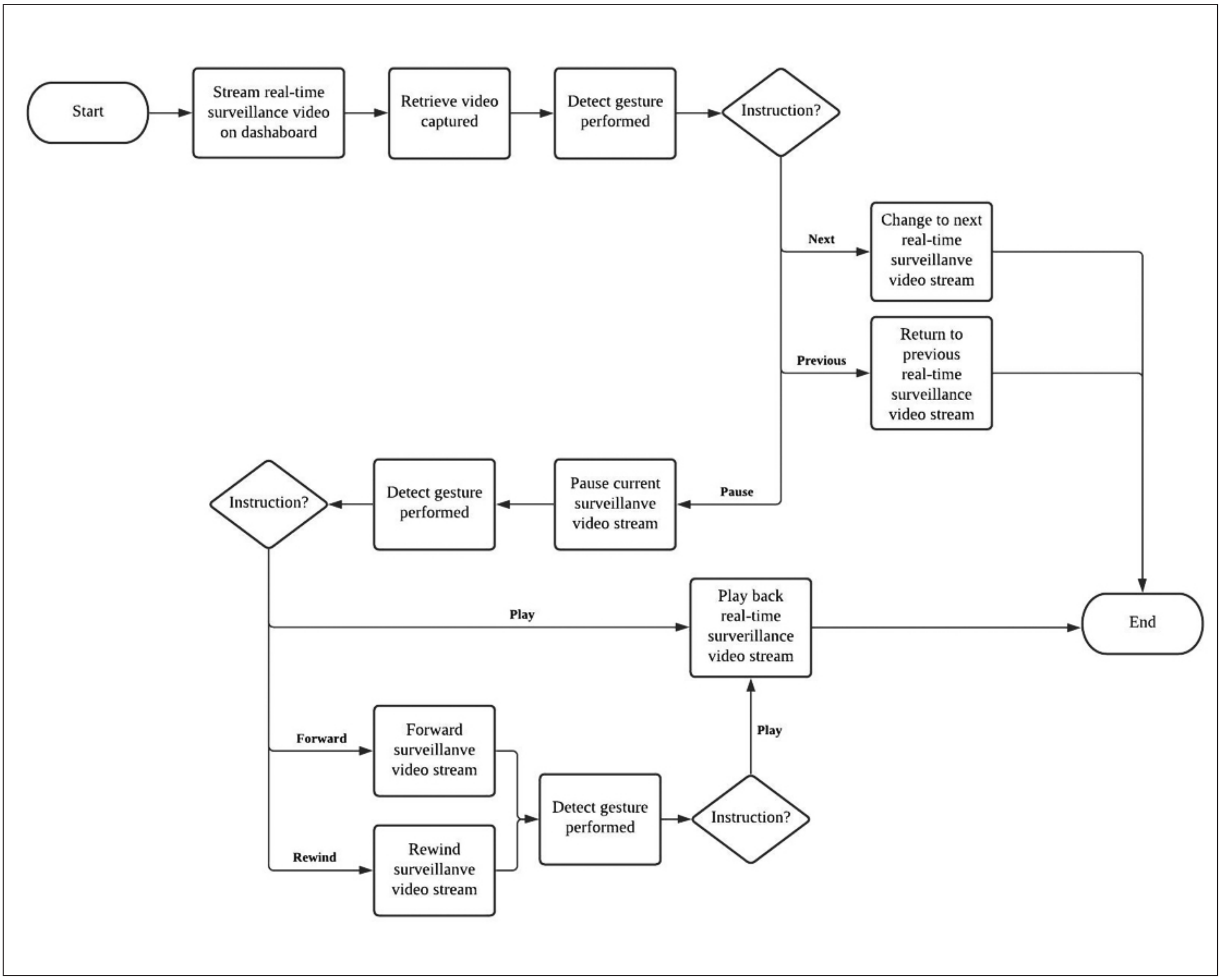


Figure 1: Flowchart of overview interactive dashboard

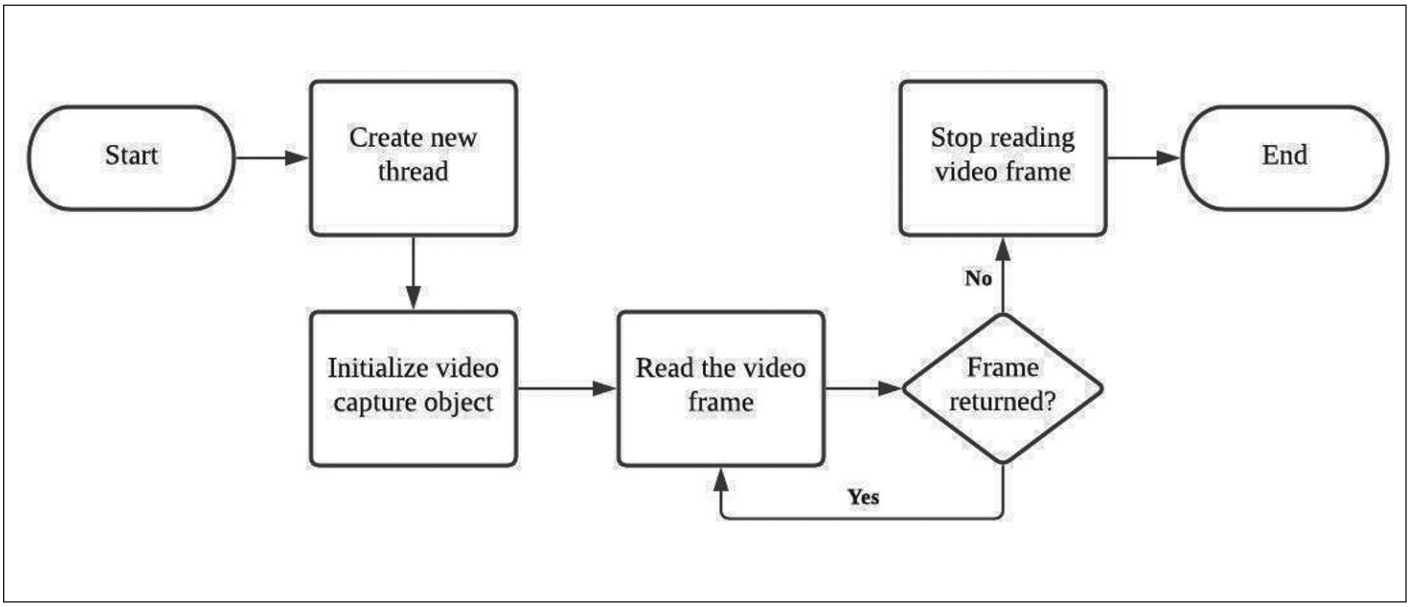


Figure 2: Flowchart of video retrieval in video streaming module

HTTP, and MULTICAST. Every live streaming protocol achieves different goals:

- TCP: Compromising video stream quality and reducing packet loss problems, however, the real-time streaming will have delays.
- UDP: Support real-time video and audio streaming.
- HTTP: Similar to the TCP protocol, yet not required ports specifications
- MULTICAST: Establish multicast group addresses and provide stream acquisition by multiple users simultaneously.

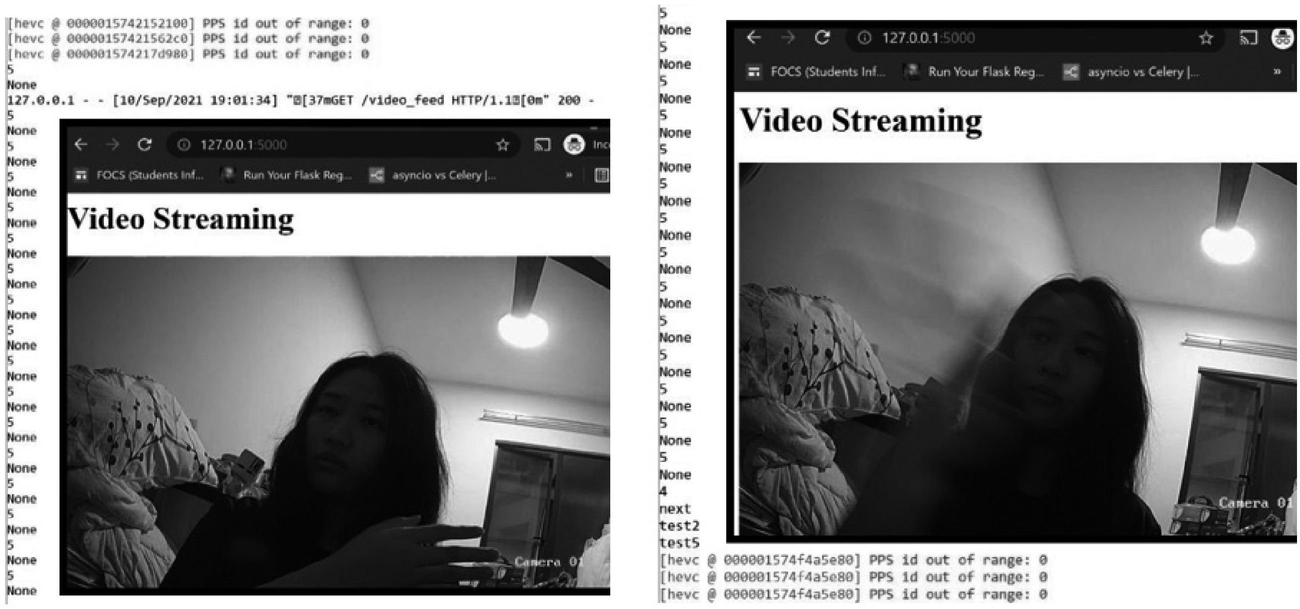
To obtain the video stream at a faster pace from the cameras to the dashboard, the UDP protocol were selected for this work.

For the experimental setup, we utilized the Jester dataset [18] to train the gesture recognition model. This dataset contains an enormous amount of gesture movements video data captured from numerous actors. The gestures which acted in those videos were suitable for HCI, e.g., swipe left, right, down, up and etcetera).

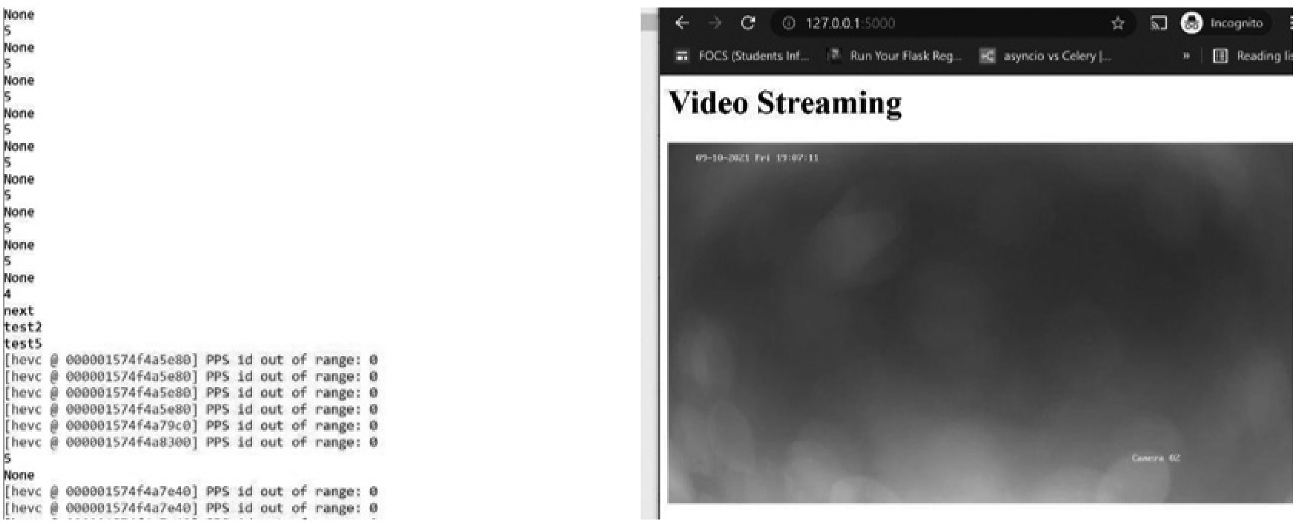
4.0 RESULTS AND DISCUSSION

Before the construction of the interactive dashboard, the gesture recognition approach was tested, and the results (gesture commands detected) were used as input to interact with the dashboard subsequently.

Figure 3 shows the test screens of the gesture recognition implementation in the flask application which was planned to be used for the dashboard set up. The gestures were used to decide which camera view to be shown on the screen. For example, a swipe left gesture was performed and this gesture was expected to change to another video stream, as captured in Figure 3 (left). After a few seconds, the program returned the gesture results which were displayed on the command prompt screen as shown in Figure 3 (right). Another few seconds passed, the gesture result was successfully passed to the controlling section, and action was carried out. Then, the video stream has changed to another in Figure 3 (bottom).



(left) Swipe Left action is performed, (right) Delay response in results



(bottom) Command occurred and the stream has changed

Figure 3: Sample output of testing on gesture recognition capabilities

However, in Figure 3 the whole testing for the program was not running smoothly, there was lagging, and the program took time to return the result. This might be due to the reason that all the functions were run in one individual function, so the previous task has to be completed before proceeding to the next task. Therefore, tasks have been separated into several functions' definitions.

Figure 4 shows the output after the tasks have been separated into individual functions. The screen output in the right shows the results with the detected gesture, returns values 5 and "None" mean that there is no gesture detected, the values 4 and "Next" means that "Swipe Left" gesture detected, the values 3 and "previous" means that "Swipe Right" gesture detected. Yet while running the application, the dashboard keeps on showing a loading screen and is unable to prompt out the dashboard. Meanwhile, gesture recognition is running normally. This might be due to the separated functions, they were not running concurrently. This means that it remains the same that it requires to wait for one task to be done only then proceed to the next task.

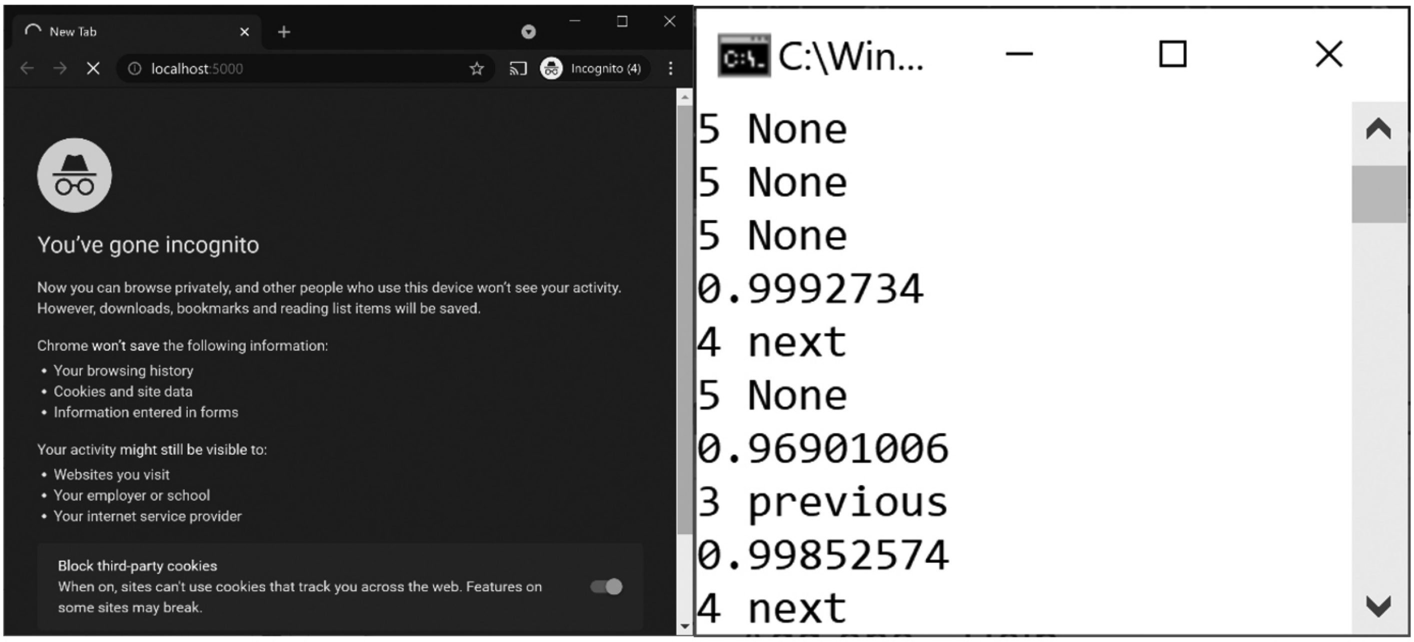


Figure 4: Only run the long-run task (gesture recognition) and dashboard unable to be prompted

To eliminate the blocking of calling other functions that were caused by the long-run task (gesture detection), the functions are expected to run asynchronously. With a focus to conduct the functions concurrently, several task queue management frameworks have been investigated.

Two of the task queues that have been tested to run two tasks concurrently. The two tasks are: (1) Sleep for 3 seconds and print a beer mug; (2) Sleep for 1 second and print a coffee mug. Figure 5 shows the result when two tasks were run concurrently using the Celery task queue. In this testing, the 2 tasks were able to run concurrently and the time elapsed for the program is around 3 seconds in total.

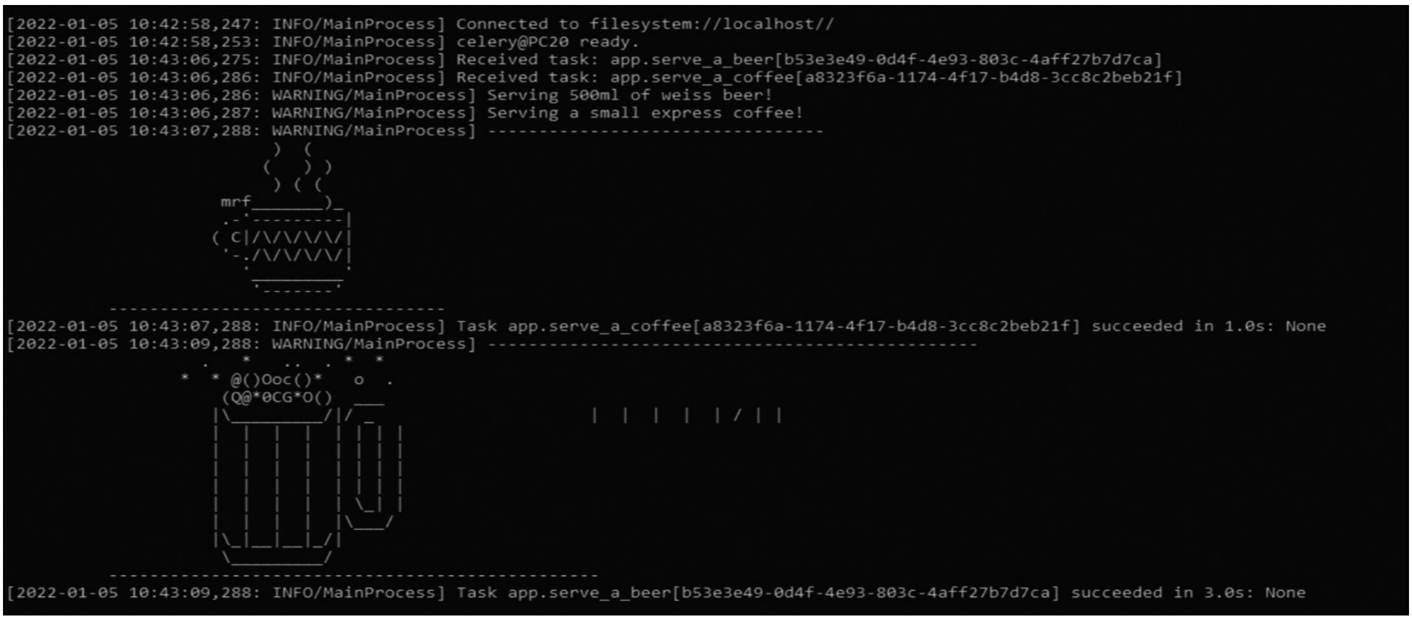


Figure 5: Running 2 tasks concurrently using Celery task queue

Figure 6 shows the result while two tasks were run concurrently using the Redis Queue task queue. In this testing, the tasks failed to run concurrently and the time elapsed for the program is around 4 seconds in total. This might be due to the failure of running them in the background using Redis Queue worker. The Redis Queue worker consists of making use of fork(), which is not available on Windows systems.

Figure 6: Running 2 tasks concurrently using Redis Queue task queue

The testing in Figure 5 and 6 were done in the Windows system, but Redis Queue is more compatible with Unix operating systems due to the usage of fork(). Therefore, the results might not be as precise as they should be. Thus, we recommend to find another suitable approach to replace the celery library to run gesture detection as a background task.

5.0 CONCLUSION

As one of the sub-projects of the TARUC Smart Campus initiative, this research project is aimed to provide artificial intelligence capabilities for the development of the dashboard employed in the i2hub, also known as the integrated innovation hub, located in the TARC cyber centre which collaborates with works such as Industry 4.0, Agriculture 4.0 and more. The dashboard that will be developed is in the pipeline to cooperate with these initiatives. For example, monitoring the robot motions from the video captured in the Industry 4.0 manufacturing site's cameras and more. This project also envisaged the dashboard with manipulation functionalities to be performed using human gestures detection.

Although there are lots of products with similar functionalities on the market, this research outcome aims to give a better solution for resolving the implementation issues faced in the real-world environment. Especially during this Covid-19 pandemic outbreak, the healthcare industry advocated making use of the interactive dashboard in better remote monitoring of the patients. This reduces unnecessary physical contact with patients or objects to control the dashboard which may be contaminated with viruses. Besides, the interactive dashboard could also be advantageous for the manufacturing industry during this pandemic. By adopting an interactive dashboard for remote monitoring of machines, it could possibly take over the worker's positions and continue the productions during the pandemic [19].

Considering this project as one of the research projects done in TARUC and the sub-project of the TAR UC smart campuses project, this project is aimed to bring new findings and knowledge that are beneficial to related topics. For instance, the video streaming protocols that have been studied in this research may be beneficial for further workings in the TARC Smart Campus's IoT initiative to expand the infrastructure and merge with more systems and other related projects on the campus. For future works, the zero-shot learning capabilities will be explored when dealing with gestures that are not learnt by the model. ■

REFERENCES

[1] B. K. Chakraborty, D. Sarma, M. K. Bhuyan, and K. F. MacDorman, 'Review of constraints on vision-based gesture recognition for human–computer interaction', IET Comput. Vis., vol. 12, no. 1, pp. 3–15, Feb. 2018, doi: 10.1049/IET-CVI.2017.0052.

[2] S. Liao *et al.*, 'Occlusion gesture recognition based on improved SSD', Concurr. Comput. Pract. Exp., vol. 33, no. 6, p. e6063, Mar. 2021, doi: 10.1002/CPE.6063.

[3] D. Y. Huang, W. C. Hu, and S. H. Chang, 'Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination', Expert Syst. Appl., vol. 38, no. 5, pp. 6031–6042, May 2011, doi: 10.1016/J.ESWA.2010.11.016.

[4] V. Gokul, G. P. Balakrishnan, T. Dubnov, and S. Dubnov, 'Semantic Interaction with Human Motion Using Query-Based Recombinant Video Synthesis', Proc. - 2nd Int. Conf. Multimed. Inf. Process. Retrieval, MIPR 2019, pp. 379–382, Apr. 2019, doi: 10.1109/MIPR.2019.00075.

[5] M. Z. Iqbal and A. G. Campbell, 'From luxury to necessity: Progress of touchless interaction technology', Technol. Soc., vol. 67, p. 101796, Nov. 2021, doi: 10.1016/J.TECHSOC.2021.101796.

[6] X. Han, X. Miao, X. Chen, G. Jiang, and L. Niu, 'Research on finger movement sensing performance of conductive gloves':, https://doi.org/10.1177/1558925019887622, vol. 14, Nov. 2019, doi: 10.1177/1558925019887622.

[7] S. Sharma and S. Singh, 'Vision-based hand gesture recognition using deep learning for the interpretation of sign language', Expert Syst. Appl., vol. 182, p. 115657, Nov. 2021, doi: 10.1016/J.ESWA.2021.115657.

[8] A. Mujahid *et al.*, 'Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model', Appl. Sci. 2021, Vol. 11, Page 4164, vol. 11, no. 9, p. 4164, May 2021, doi: 10.3390/APP11094164.

[9] E. Tsironi, P. Barros, C. Weber, and S. Wermter, 'An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for gesture recognition', Neurocomputing, vol. 268, pp. 76–86, Dec. 2017, doi: 10.1016/J.NEUCOM.2016.12.088.

[10] H. R. Lee, J. Park, and Y. J. Suh, 'Improving Classification Accuracy of Hand Gesture Recognition Based on 60 GHz FMCW Radar with Deep Learning Domain Adaptation', Electron. 2020, Vol. 9, Page 2140, vol. 9, no. 12, p. 2140, Dec. 2020, doi: 10.3390/ELECTRONICS9122140.

[11] G. Zhu, L. Zhang, P. Shen, and J. Song, 'Multimodal Gesture Recognition Using 3-D Convolution and Convolutional LSTM', IEEE Access, vol. 5, pp. 4517–4524, 2017, doi: 10.1109/ACCESS.2017.2684186.

[12] M. Abbasi, A. Shahraki, and A. Taherkordi, 'Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey', Comput. Commun., vol. 170, pp. 19–41, Mar. 2021, doi: 10.1016/J.COMCOM.2021.01.021.

[13] G. Sinha, R. Shahi, and M. Shankar, 'Human Computer Interaction', Proc. - 3rd Int. Conf. Emerg. Trends Eng. Technol. ICETET 2010, pp. 1–4, 2010, doi: 10.1109/ICETET.2010.85.

[14] C. F. Lin, H. T. Chiao, R. K. Sheu, Y. S. Chang, and S. M. Yuan, 'A fault-tolerant ONVIF protocol extension for seamless surveillance video stream recording', Comput. Stand. Interfaces, vol. 55, pp. 55–72, Jan. 2018, doi: 10.1016/J.CSI.2017.04.005.

[15] H. C. Kaşkavalci and S. Gören, 'A Deep Learning Based Distributed Smart Surveillance Architecture using Edge and Cloud Computing', Proc. - 2019 Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. Deep. 2019, pp. 1–6, Aug. 2019, doi: 10.1109/DEEP-ML.2019.00009.

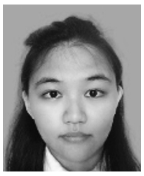
[16] M. Bugeja, A. Dingli, M. Attard, and D. Seychell, 'Comparison of Vehicle Detection Techniques applied to IP Camera Video Feeds for use in Intelligent Transport Systems', Transp. Res. Procedia, vol. 45, pp. 971–978, Jan. 2020, doi: 10.1016/J.TRPRO.2020.02.069.

[17] Y. Wu *et al.*, 'Efficient maliciously secure two-party mixed-protocol framework for data-driven computation tasks', Comput. Stand. Interfaces, vol. 80, p. 103571, Mar. 2022, doi: 10.1016/J.CSI.2021.103571.

[18] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, 'The jester dataset: A large-scale video dataset of human gestures', Proc. - 2019 Int. Conf. Comput. Vis. Work. ICCVW 2019, pp. 2874–2882, Oct. 2019, doi: 10.1109/ICCVW.2019.00349.

[19] T. Chen and C. W. Lin, 'Smart and automation technologies for ensuring the long-term operation of a factory amid the COVID-19 pandemic: an evolving fuzzy assessment approach', Int. J. Adv. Manuf. Technol., vol. 111, no. 11–12, pp. 3545–3558, Dec. 2020, doi: 10.1007/S00170-020-06097-W/FIGURES/8.

PROFILES



WEN LIN YONG is pursuing her postgraduate studies in Computer Science at Tunku Abdul Rahman University College, Kuala Lumpur. At the same time, she was engaged as a research assistant which contributed to the TARUC Smart Campus initiative. She has completed her undergraduate studies with Diploma in Business Information Systems and bachelor's degree (Honours) in Enterprise Information Systems.
Email address: yongwl-wa16@student.tarc.edu.my

JUN-KIT CHAW received the BEng (Computer) and PhD (Electrical Engineering) from Universiti Teknologi Malaysia (UTM), currently serving as lecturer in the Institute of IR4.0, Universiti Kebangsaan Malaysia (UKM, Bangi), Malaysia. Previously, he served as Senior Lecturer in the Faculty of Computing and Information Technology at Tunku Abdul Rahman University College (TAR UC). He is a SAS certified predictive modeller and machine learning specialist as well as NVIDIA Deep Learning Institute certified for demonstrating competence in deep learning for computer vision. His primary research interests include computer vision and deep learning. His research projects include optimization of manufacturing processes, predictive maintenance and medical data analytics.
Email address: chawjk@ukm.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