A Dynamic Model for the Computation of Gesture Types for Image-based Software Agents

Tse-Chuan Hsu[1], *Chih-Hung Chang*[2], *William Cheng-Chung Chu*[3*]
Shou-Yu Lee[3], *Shih-Yun Huang*[3]

[1] *Department of Computer Science & Information Management, Soochow University, Taiwan*
[2] *College of Computing and Informatics, Providence University, Taiwan*
[3] *Department of Computer Science, Tunghai University, Taiwan*
tchsu@scu.edu.tw, ch.chang@gm.pu.edu.tw, cchu@thu.edu.tw, shouyu@thu.edu.tw, shihyh@thu.edu.tw

Abstract

Computer vision technology allows the computer to interact with a human operator to quickly complete the interpretation of events and improve the operational workflow of processing events. As computer vision technology continues to evolve, the cost of the equipment continues to increase. Therefore, the stability of the system can be ensured through the design of the middleware and the calculation of the auxiliary functions of the software agents. At present, the recognition of characters for image processing is based on the technology of image recognition, which can provide a more flexible user experience. However, the dilemma of contactless design lies in the processing and calculation of images, which should reduce the inconvenience caused by delays.

This article uses a Raspberry Pi as an example of a computing proxy application. After the visual inspection, the verification operation of the software is carried out. Our system is the detection of hand position and movement, and the detection of hand mark position in reconnaissance. In addition, we simultaneously developed management and remote control events and connected to the remote edge computer. After that, we successfully completed the automatic control and serial application of two different edge computing recognition jobs, and verified the image vision computing based on the Raspberry Pi software agent, which can be used for image vision analysis and control applications.

Keywords: Computer vision, Edge computing, Handmark, Software agent, Image recognition

1 Introduction

To the gestural recognition technology. Google has released a set of palm detection model BlazePalm and hand key point model [1-2] which can further detect and record gestures according to the changes in the coordinate values of 21 points [3]. Use of dynamic receive images for real-time computing. When using the model suite, which can be combined with virtual reality or augmented reality situations to control and use gesture behavior. The technology is more flexible to offer portable devices that do not need a computer to monitor the application.

In the application of computer recognition technology, the current image computation recognition technology needs to be monitored by a computer. When it comes to image monitoring and application technology, there is also a lack of flexible application services. For example, when a user turns on his computer, it is more convenient to sit in front of the computer and use the keyboard and mouse than video vision to detect gestures. Therefore, for computer application operations, although Mediapipe provides precise deep learning calculation technology, finger positioning information is calculated publicly. However, where the application is limited to the operation in front of the computer, the value and application of technology development is limited.

With advanced IT application technology, computation tasks can be distributed to last-mile nodes, and the M2M communication mechanism can be combined to decompose computation tasks [4-5]. In the research, the image analysis and computing tasks are performed by the image computing nodes, and the computing power of the edge computing device endpoint is properly used to transfer the computing information from a single host to the distributed terminal nodes. The image processing results are returned to the MQTT broker through the MQTT communication mechanism. In the case of different remote devices, such as PCs, mobile phones or large remote control devices, the results of the visual recognition process are transmitted to different device nodes via the host. The main function is to allow users to operate without computer equipment by disaggregating the event computation tasks and the controlled devices. To improve the service of the engineering stress model, after the Edge device performs the calculation, it sends the result of the calculation task to the controlled computer device.

The structure of this thesis is as follows: Chapter 2 In Chapter 2 we have a review of relevant previous research. We consider gesture recognition and advanced computing technologies. In chapter 3 we present an experimental framework for the demonstration of gesture recognition in an edge processing environment. Chapter 4 examines the experimental results and investigates the performance differences between the treatments and the improvements

*Corresponding Author: William Cheng-Chung Chu; E-mail: cchu@thu.edu.tw

DOI: 10.53106/160792642023122407011

in computing edges. The final chapter is a discussion of the empirical application results and experimental service application models derived from the research.

2 Related Works

2.1 Deep Learning with OpenCV

Deep learning has shown good results in recognizing natural language objects. This can be text, images or natural language. Zhang, Y (2022) [6] presented initial ideas for a basic gesture recognition system using Google's TensorFlow deep learning framework and gesture recognition components from the MediaPipe and OpenCV machine vision open-source libraries. The Google-trained datasets were used to extract the skeletal key points of the cursor, and then a pre-processed database was used to train the neural network to create an initial model, which was then modified and changed.

However, successful image recognition relies heavily on supervised learning requiring extensive manual labelling. To avoid the costly collection of tagging data, and in domains where there are few standard pre-trained models, self-supervised learning is a form of unsupervised learning which allows the network to learn rich visual features. Semi-supervised learning, semi-weakly supervised learning, incremental learning, and small sample learning are presented as case studies in a study on self-supervised image recognition using deep neural networks by Ohri, K. (2021) [7].

2.2 Graph Neural Networks

The first graph neural networks, first outlined by Gori, M. (2020) [8-9], consisted of two main components, graph and neural network, combining our familiar data structures with deep learning. As convolutional CNN analysis networks evolved to incorporate graph learning, the combination of (convolutional neural networks (GCNs) for graphs) combined the learning results of social networks, crawling pen pairs, etc. Niepert, M. (2016) [10] has tried to adapt his framework in order to integrate the results of the CNN-based derivations into graph learning. Similar to image-based convolutional networks that operate on locally connected regions of the input, time series of image nodes and adjacent images are computed, and discrete and continuous node and edge attributes are learned in tandem, evolving into today's GNN framework. Deep learning can be applied in many ways. Examples include image classification, image processing, and speech recognition. The core development of the approach has been to deal with structural irregularities, mainly through machine learning methods that use graph-structured data for adaptation and prediction. Weng, X. (2020) [11] applied a feature interaction mechanism in 3D object tracking learning. In the proposed method, the image is first transformed into a graph, where nodes represent the parts of the image and edges represent the binary relationships between these parts.

2.3 Image Training MediaPipe

Developed by Google Research, MediaPipe is an application framework for multimedia machine learning models [12-14]. It uses a machine learning framework to process images and audio, using the image information to create corresponding nodes, and further linking the nodes to create data mapping lists. The technology is currently being used in a number of projects for face mesh, pose recognition, face recognition and other applications. For hand recognition, Mediapipe uses a BlazePalm to detect the palm, which is then used as hand keypoints, allowing the model to directly predict 21-point coordinates [15].

For image sampling training, Subramanian, B (2022) [16] showed that after training 100 images, the conventional standard GRU gave the accuracy indicated by the green line, and its analytical accuracy and false loss performance could be quickly corrected to give better results after more than 20 images, as we can see in Figure 1. The optimal MediaPipe-optimized gated recurrent unit (MOPGRU) proposed in the data of this study achieves the best training model results. Therefore, we can observe that the sampling characteristics of MediaPipe image connections have a certain quality for the sampling performance of deep learning and can reduce the cost of computer training and computation.

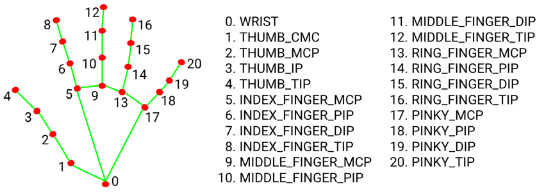


Figure 1. MediaPipe Hands 21 hand landmarks [17]

2.4 Prediction of Joint Gestures

The prediction of the key points of the hand has an impact on the gesture results that can be compared by computer computation and prediction computation. Currently, many scholars have suggested gestural training methods and established models through different dataset sampling methods to compare. Starting from deep learning, Tkach A. (2017) [18] once proposed the event analysis of image tracking, where images are reproduced through static images that correspond to the dynamic movements of the hand. The results show that, after the use of time series tracking, the modeling model and the volumetric model, respectively, have been trained with convolutional nets.

The position data of the relevant image nodes are computed by the genetic computational model. The accumulated uncertain data are corrected. For example, in Figure 2, the front end angular motion is influenced by the finger's red stop. Therefore, if the angle tends to be 40 degrees, then the front end is in the first segment, and it is impossible for the finger to leave the node after 40 degrees. Based on this model, we can refer to more accurate training results for verification, although this research is based on sampling, creating virtual gestures and automatically generating animations for use. This allows us to use the OpenCV and MediaPipe libraries to find the key points in the hand painting, accurately integrate image processing and artificial intelligence capabilities, and create computer vision AI applications.

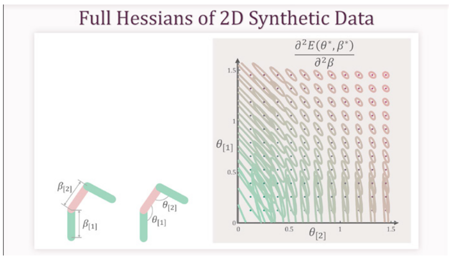
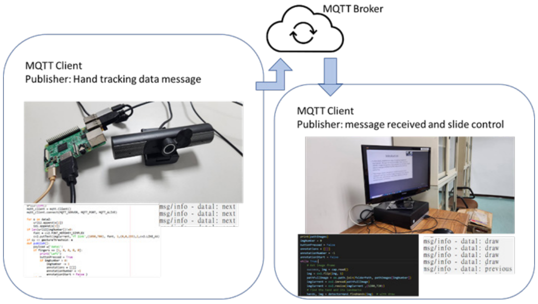


Figure 2. Results of calculations and possible finger movement scenarios [18]

3 System Architecture

In the research, the Hand Marking Model is used to process the marking. After the camera obtains the whole image for position searching and detecting, we use the hand landmark model to accurately determine the 21 3D hand joint coordinates in the detected hand area by regression. point positioning. Since the positioning information can predict the operation of the coordinate points when curling, fisting and so on occur in gestures and the joint position status cannot be detected in time by the image, it can still be replaced by calculation through machine learning, avoiding image interference and causing the system to fail to detect judging action events.

In the basic environment of the experiment, we first detect the characteristics of the hand, view and compute the image of the hand in real time through the webcam, use the media pipe to extract the joint points, construct the trajectory data of the joint point movement, and transfer the movement behavior event to a tag, notify the MQTT broker to perform subsequent broadcasts through tag events. After the transmission, the remote computer can receive the data signal through the transmission, and can process the event after the transmission, and further realize that the distributed edge computing processor only processes its own work item, and the image sampling endpoint can focus on image processing and signal distribution simultaneously. The computer end points handle the short control and computation, and in the construction of various edge computing infrastructure environments, they have better processing performance because the respective devices only exchange signals. Figure 3 is a sketch of our experiment implementation architecture.


MQTT Broker
MQTT Client
Publisher: Hand tracking data message
MQTT Client
Publisher: message received and slide control

```
msg/info - datal: next
msg/info - datal: next
msg/info - datal: next
```

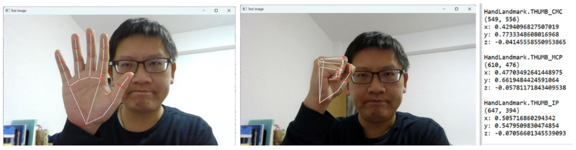
```
msg/info - datal: draw
msg/info - datal: draw
msg/info - datal: draw
msg/info - datal: draw
msg/info - datal: previous
```

Figure 3. Architecture of the experiment implementation

For another, the software's mediating agent system assesses conduct. Therefore, for a behaviour event, the operation behaviour of the agent in accordance with the event can be modified more flexibly. In this research experiment for the actual validation of the design environment, the finger positioning visual markers are used as different cues, e.g. thumb for previous page, little finger for next page, combination of index and middle finger for laser. Since the calculation and analysis of the image are performed by the Raspberry Pi, the agents can send the analysis results of the image search to all the different nodes via the MQTT publishing. For example, in two completely different space-time environments, digital signals based on image analysis results can be transmitted not only at the near-end, but also correspond to different distant simultaneous processes.

3.1 Computing based on the Recognition of Gestures

In this way, the coordinates of the x-, y- and z-axes can be drawn at a fixed point by means of computer calculation. Once we have obtained the relevant numerical information, we can then carry out operations to calculate the data. Taking the image in Figure 4 as an example, our joint position moves from the coordinate 0.42 to the coordinate 0.5, which can record the gesture behavior, and further establish the corresponding endpoint task control and analysis events on the behavior coordinates.



```
HandLandmark.THUMB_CMC
(549, 556)
x: 0.4294096827507019
y: 0.7733348608016968
z: -0.04145558550953865

HandLandmark.THUMB_MCP
(610, 476)
x: 0.47703492641448975
y: 0.6619484424591064
z: -0.05781171843409538

HandLandmark.THUMB_IP
(647, 394)
x: 0.505716860294342
y: 0.5479509830474854
z: -0.07056601345539093
```

Figure 4. Image joint sampling and joint values

3.2 Mobile Computing and Image Tracking

In the trial, it is necessary to record the coordinates of the two nodes, then calculate the action, which can modify or move the action. Thus, in the gesture positioning research, see Zhang, F (2020) [13], by first detecting the position, after further calculating the joint position through the neural network, first obtain the color image RGB, and then predict the hand skeletal structure. There are two different blocks shown in Figure 5, two models that are used in the study: 1) a palm detector, and 2) a hand model such as the concept of object tracking, which captures the state of movement behaviors.

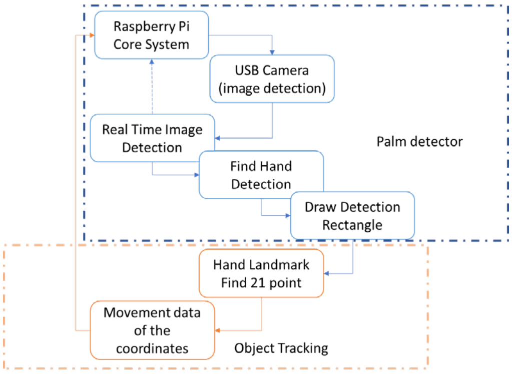


Figure 5. Palm recognition and motion detection

We calculate Palm recognition and motion detection independently of the software agent that processes the images.

1. The software agent has two different tasks. The first is to get the boundaries of the intended analysis. In the case of real-time image sampling analysis, we must first define the boundaries of the analysis so that the analysis can be more accurate. Therefore, we first detect the hand's position through the camera and set the hand's position as a rectangular boundary so that image analysis can be restricted to this rectangle. Under the framework of deep learning for recognition computation.
2. After the calculation of the frame is completed, we will perform the node detection on the image, and we will use the hand landmarks to capture the positions of the nodes. Since the change in node position affects the work item, the object movement event is detected by calculating the difference before and after the node. When an object is removed and a moving node occurs, the software agent on the Raspberry Pi endpoint converts the behavior to a label, broadcasts the label, and synchronizes all other nodes.

3.3 Division of Labor between Agents

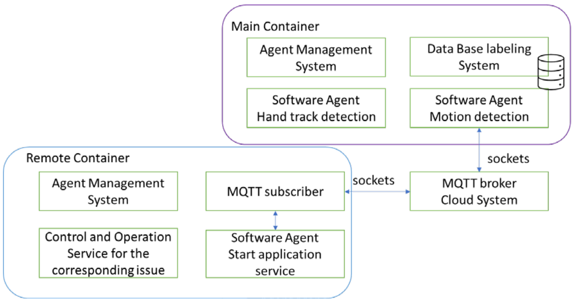

Main Container
Agent Management System
Data Base labeling System
Software Agent Hand track detection
Software Agent Motion detection
sockets
Remote Container
Agent Management System
MQTT subscriber
sockets
MQTT broker Cloud System
Control and Operation Service for the corresponding issue
Software Agent Start application service

Figure 6. Communicate with local and distant agents

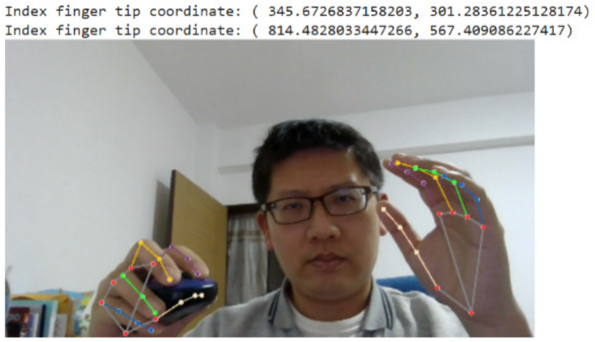
The main agent manages the tasks it needs to handle the communication services and image recognition in the main container of Figure 6. We use MQTT to broadcast the converted signal to the broker in the communication terminal service. The advantages and characteristics of broadcast are that the broker can listen online at any time, so the agent only needs to send the broadcast signal from the communication port without having to confirm the signal. Gesture recognition and identification software services are required. We need to manage the characteristics of the hand and the movement events, build a tagged database, and send the tagged database to all the nodes first. When gesture behaviour and data If the tags in the library match, send notification broadcast content tags to communicate with other remote agents of the same broker.

When the remote container agent receives the signal provided by the broker, it starts checking the database work event corresponding to the tag, and performs operations on the started or ready to start software according to the tag, such as opening, closing, and slideshow. Items such as switching pages are performed according to the tagged events.

4 Explaining Experimental Results

4.1 Deep learning Node Computing based on Image Visions

In this study, hand motion behavior is used because hand features must be obtained and node information calculated after the hand is photographed with the camera. The device itself first processes the event on the image, detects the two hands, and extracts the hand joint point frame after detection image. Figure 7.



```
Index finger tip coordinate: ( 345.6726837158203, 301.28361225128174)
Index finger tip coordinate: ( 814.4828033447266, 567.409086227417)
```

Figure 7. Mark the nodes of the hand joints

To obtain the outline of the hand posture in space, after obtaining the coordinate points of the hand frame, we independently reconstruct the coordinate information in the three-dimensional space. We can convert the coordinate-based information into image reproduced. When the image coordinates can be recreated, we can check the coordinate information and obtain the image movement event in the coordinate movement condition.

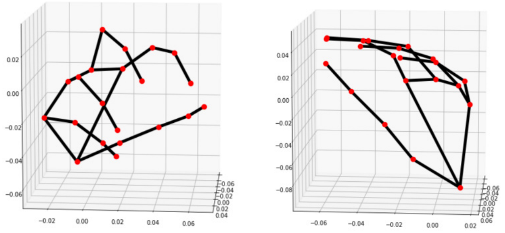


Figure 8. Compare left and right marker position detection results

The three-dimensional coordinate point of a gesture corresponding to the movement of each gesture in three-dimensional space can be reconstructed after the coordinate position conversion. As shown in Figure 8. For example, when the index finger and middle finger move in unison, the data are connected by the same node, and the different changes in the Y axis can be used to reconstruct the action label that has been converted into the action state. Therefore, we are able to create the labels we need according to the different gestures. However, from this research, if the action label is too complex, because the accuracy of the computation is not as agile as the independent operation, the system will cause identification errors and failures after the computational operation. Thus, it follows from this study that in the case of experimental validation, it will be necessary to clarify the actions of changing pages, closing applications, etc., and to focus only on the nodes of gestures that can be called accurate sampling decisions.

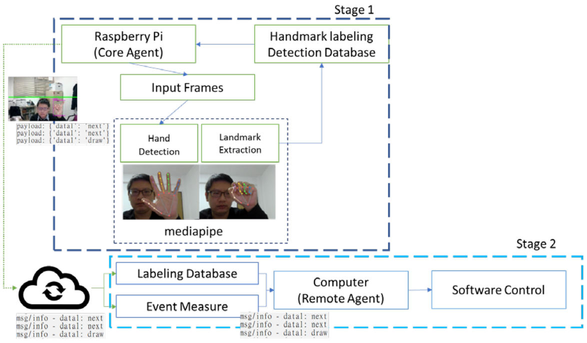
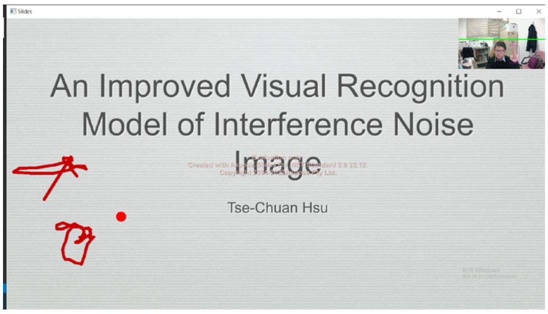


Figure 9. Distribution of the video control signal using edge computing

From the study in Figure 9, an agent is used to calculate images via a media pipe, then build a tag database of different motion events, and send the tag database to a remote computing device in synchrony. When the remote computer system has a tag database, the positioning coordinates of the gesture can be used for reference matching. For example, if our gesture only signals the little finger, we can set it to the next slide. So the system gets the recognition result and sends the next page message on the next page. It is automatically broadcast to all connected remote nodes after the cloud MQTT broker receives the signal. This allows the remote computer to switch the local slide on its own, rather than having to play the picture through the computer.

4.2 Usage Verification Test Results

A screenshot of our experiment is shown in Figure 10. The upper right corner is used to check whether the behaviors of our operation is consistent with the results of the camera sampling. We took the Raspberry Pi camera image separately and placed it in the top right corner for comparison. Under the right circumstances, the user is not able to see the top right camera screen. As we only send tag information, the briefing computer does not need to be camera-equipped for demonstrations, and tag data can still be located and controlled remotely.

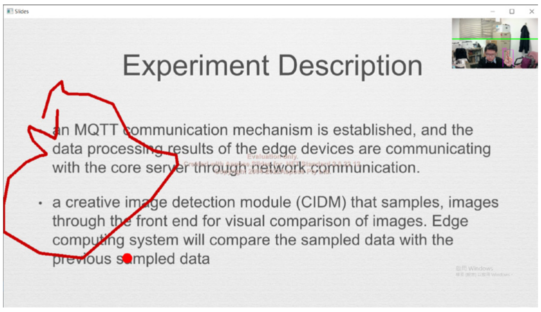

An Improved Visual Recognition Model of Interference Noise Image

Tse-Chuan Hsu

Figure 10. Distribution of the video control signal using edge computing

As a result, in Figure 10, on the screen we see a red crosshair dot, and when we mark the image in the database, we let the computer know that if we just raise our index finger, it will fire the coordinates of the briefing laser dot and send them back. The positions on all the nodes are sent back using MQTT. In this case, just the axis of the 8th point is returned. As the positioning point can be compared with the relative position of our presentation x and y axes to complete the positional comparison of the laser pointer in the presentation, we do not collect and return the z axis data.

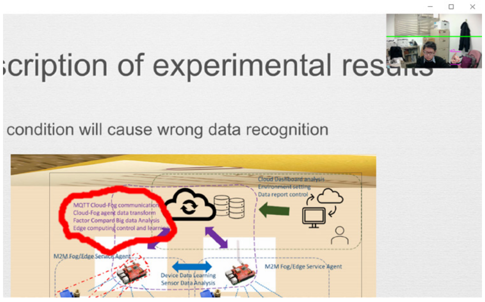
We have also set up different modes of operation in experiments to test whether the proxy signaling method that we have constructed is feasible. On the remote computer, we designed a drawing proxy event action, this action is imitating when we make a presentation, we circle the description and drawing in the presentation. When we lift index and middle finger simultaneously, it is detected that the positions of two anchor points 8 and 12 are connected simultaneously, and then draw. As shown in Figure 11, the action of this drawing is to send the coordinate information to the remote computer environment, and the demonstration computer starts the drawing operation locally through the agent, and the Raspberry Pi has no right to write these signal information. In the case of gesture marking, in addition to text marking, we have demonstrated through experiments that anchor points can mark it. Figure 12 shows that if the gesture lasts for a long time, the system automatically increases the thickness of the line for image index marking.


Experiment Description

an MQTT communication mechanism is established, and the data processing results of the edge devices are communicating with the core server through network communication.

- a creative image detection module (CIDM) that samples, images through the front end for visual comparison of images. Edge computing system will compare the sampled data with the previous sampled data

Figure 11. We draw a coherent region for testing


scription of experimental results

condition will cause wrong data recognition

Cloud Dashboard analysis
Environment setting
Data report control
MQTT Cloud-Fog communication
Cloud-Fog agent data transform
Factor Compard Big data Analysis
Edge computing control and learning
M2M Fog/Edge Service Agent
Device Data Learning
Sensor Data Analysis
M2M Fog/Edge Service Agent

Figure 12. Set parameters of analysis process

5 Conclusions and Future Works

In this study, a multi-point distributed communication mode based on edge computing will be redesigned, and the analysis of image visual computing and the transmission of messages will be redesigned by a single host. It uses fog computing to reduce the hardware resources needed by the device, combined with the characteristics of edge computing, to provide each edge node device with a unique way of computing and processing event distribution task. Considering the fact that current image processing computations require a large number of computer cores for computation, the mature media pipe model is used in the study for deep learning computation of endpoint connections. Through research and experimental findings, we find that regardless of the real-time analysis and return of events, we are able to schedule the computing hardware requirements for analysis to the motherboard of a small computer (raspberry pi) that has low real-time performance.

The experimental results of this research were successfully passed, changing the mode of centralized computing analysis, using software agents to support the computing resources required for edge computing, completing multi-point computing, and feedback computing events in real time. The study initially uses edge computing to handle the newsletter's paging and background checking operations. In the future, there will be innovative breakthroughs in edge computing-based vision value-added service applications, including intelligent factory robot arm operation and real-time remote visual broadcasting systems. Several applications, especially in the field of computer vision, combined with deep learning models to integrate image vision operations into the edge computing node environment, break through the current technology to control machine interfaces requiring human gesture and posture.

Because of current experimental results, more complex dynamic behavioral and edge computing hardware performance stress tests were not performed. In future research, we will experiment and test the behavioral computing performance of various complex operations to understand the edge computing environment. The next step is to investigate and verify the degree of message delay in the real-time operation of multi-node systems. Based on the software agent service framework of this study, more detailed discussions and investigations are conducted in the subsequent work.

Acknowledgment

This research was supported in part by National Science and Technology Council of the Republic of China under contract MOST 111-2221-E-031-006.

References

[1] H. Maurya, C. Chauhan, H. Tiwari, A. Jain, Analysis on hand gesture recognition using artificial neural network, *Ethics and Information Technology (ETIT)*, Vol. 2, No. 2, pp. 127-133, 2020.

[2] A. S. B. Pauzi, F. B. M. Nazri, S. Sani, A. M. Bataineh, M. N. Hisyam, M. N. Jaafar, M. N. A. Wahab, A. S. Mohamed, Movement estimation using mediapipe blazepose, *7th International Visual Informatics Conference, IVIC 2021*, Kajang, Malaysia, 2021, pp. 562-571.

[3] Indriani, M. Harris, A. S. Agoes, Applying hand gesture recognition for user guide application using MediaPipe, *2nd International Seminar of Science and Applied Technology (ISSAT 2021)*, Video Conference, 2021, pp. 101-108.

[4] S. Patil, P. Gokhale, Systematic Review of Resource Allocation Methods Using Scheduling for M2M (Machine to Machine Communication) in IoT Network, in: P. N. Mahalle, G. R. Shinde, N. Dey, A. E. Hassanien (Eds.), *Security Issues and Privacy Threats in Smart Ubiquitous Computing*, Springer, Singapore, 2021, pp. 213-224.

[5] R. Prasad, V. Rohokale, Internet of Things (IoT) and machine to machine (M2M) communication, in: *Cyber Security: The Lifeline of Information and Communication Technology*, Springer Series in Wireless Technology, Springer, Cham, 2019, pp. 125-141.

[6] Y. Zhang, X. Pu, X. Wang, H. Guo, K. Liu, Q. Yang, L. Wang, Design concept of sign language recognition translation and gesture recognition control system based on deep learning and machine vision, *Second International Conference on Optics and Communication Technology (ICOCT 2022)*, Hefei, China, 2022, pp. 124731X-1

[7] K. Ohri, M. Kumar, Review on self-supervised image recognition using deep neural networks, *Knowledge-Based System*s, Vol. 224, Article No. 107090. July, 2021.

[8] M. Gori, G. Monfardini, F. Scarselli, A new model for learning in graph domains, *2005 IEEE international joint conference on neural networks*, Montreal, QC, Canada, 2005, pp. 729-734

[9] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE transactions on neural networks*, Vol. 20 No. 1, pp. 61-80, January, 2009.

[10] M. Niepert, M. Ahmed, K. Kutzkov, Learning convolutional neural networks for graphs, *International conference on machine learning*, New York, NY, USA, 2016, pp. 2014-2023.

[11] X. Weng, Y. Wang, Y. Man, K. M. Kitani, Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 6499-6507.

[12] C. Lugaresi, J. Tang, J. H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, M. Grundmann, Mediapipe: A framework for perceiving and processing reality, *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, USA, 2019, pp. 1-4.

[13] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. L. Chang, M. Grundmann, *Mediapipe hands: On-device real-time hand tracking*, arXiv preprint arXiv: 2006.10214, June, 2020. https://arxiv.org/abs/2006.10214

[14] T. T. Nguyen, D. T. Pham, H. Vu, T. L. Le, A robust and efficient method for skeleton-based human action recognition and its application for cross-dataset evaluation, *IET Computer Vision*, Vol. 16, No. 8, pp. 709-726, December, 2022.

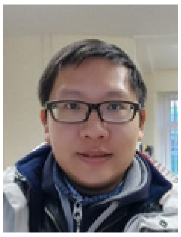
[15] M. Oudah, A. Al-Naji, J. Chahl, Hand gesture recognition based on computer vision: a review of techniques, *Journal of Imaging*, Vol. 6, No. 8, Article No. 73, August, 2020.

[16] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K. H. Park, J. Kim, An integrated mediapipe-optimized GRU model for Indian sign language recognition, *Scientific Reports*, Vol. 12, No. 1, pp. 1-16, July, 2022.

[17] MediaPipe Hands. (n.d.), Google. https://google.github.io/mediapipe/solutions/hands.html

[18] A. Tkach, A. Tagliasacchi, E. Remelli, M. Pauly, A. Fitzgibbon, Online generative model personalization for hand tracking, *ACM Transactions on Graphics*, Vol. 36, No. 6, pp. 1-11, November, 2017.

Biographies



Tse-Chuan Hsu received his M.S. degree in Computer Engineering from Tunghai University, Taichung, Taiwan and received his PhD from BathSpa University, UK. 2020. He is currently an Assistant Professor at Soochow University, Taipei, Taiwan. His research interests include Internet of Things, cloud computing, edge computing, computer vision, software engineering and data analysis systems.

Chih-Hung Chang received the M.S. and Ph.D. degrees in computer science from Feng Chia University, Taichung, Taiwan. He is an Associate Professor with the Department of Computer Science and Communication Engineering, Providence University, Taiwan. His research interests include software engineering, cloud service, and deep learning.

William Cheng-Chung Chu is a Distinguished Professor in the Department of Computer Science and the director of the Software Engineering at Tunghai University, Taiwan. Chu received a PhD in computer science from Northwestern University. He is a member of the IEEE Computer Society.

Shou-Yu Lee received the M.S. degree in computer science from Tunghai University, and the Ph.D. degree in software engineering from The University of Texas at Dallas, Richardson, TX, USA. He is currently an Assistant Professor with the Computer Science Department at Tunghai University, Taiwan. His current research interests include software fault localization, software risk analysis, machine learning applications, and self-adaptive computing.

Shih-Yun Huang has been an assistant professor at the Department of Computer Science at Tunghai University since 2022. He received his Ph.D. degree in the department of Electrical Engineering science from the National Dong Hwa University, Taiwan, in 2022. The B.S. and M.S. degree in Electrical and Electronic Engineering, National Ilan University, in 2009 and 2013, respectively. His research interests include Wireless Communications, Mobile Edge Computing, AI, Cloud Computing, and the Internet of Vehicles.