MYU Tokyo

S & M 4148

# Design of Real-time Interactive Healthcare System for Acupoint Analysis Based on Augmented Reality

Yu-Xiang Zhao,* Chung-Yen Wei, and Bo-Yuan Xu

Department of Computer Science and Information Engineering, National Quemoy University,
No. 1, University Rd., Jinning Township, Kinmen 89250, Taiwan

(Received August 30, 2024; accepted August 14, 2025)

***Keywords:*** interactive healthcare, edge computing, acupoint analysis, augmented reality, MediaPipe, gesture recognition, DNN, RASA, BERT

Acupressure is a valuable method for maintaining good health and also helps in the overall regulation of bodily functions and the promotion of physical fitness. In this paper, we propose a real-time interactive healthcare system for acupoint analysis based on augmented reality (AR). We utilized the edge computing of the Jetson Nano J1010 development board, and employed MediaPipe technology and an acupoint mapping algorithm to display acupoints on the screen. By using our proposed deep neural network (DNN)-based hand gesture recognition, we achieved AR-based interaction with acupoints. Users can view the locations of the corresponding acupoints on the body and the descriptions of the acupoints through gestures. Additionally, we developed a mobile app that enables interaction with the system via Bluetooth technology. Users can select the symptoms they wish to understand, and the system will display acupoints that can alleviate those symptoms. We also developed a bidirectional encoder representations from transformers (BERT)-based AI consultation, which utilizes the open-source conversational receive, appreciate, summarize, ask (RASA) natural language understanding (NLU) to determine the user's intents from input on the mobile app. When the intent is related to medical content, the information is passed to a BERT model for diagnosis analysis, determining possible diseases or symptoms and providing corresponding recommended acupoints for symptom relief. This aids users in making preliminary judgments about their own conditions. All AI diagnostic information is transmitted to the server side, where physicians can view user information through a website. By visualizing the data through charts, physicians can quickly understand the user's condition and further save medical resources. This acupoint-analysis-based healthcare system offers a safe, convenient, and efficient solution for health management, particularly valuable for individuals lacking experience and medical knowledge. We hope that by using this proposed system, users will feel like they have a virtual healthcare advisor at their disposal.

## 1. Introduction

Acupressure is a valuable method for maintaining good health, which relies on specific acupoints distributed throughout the body. These acupoints can be targeted to address a wide

*Corresponding author: e-mail: yxzhao@nqu.edu.tw

https://doi.org/10.18494/SAM5356

range of ailments while also contributing to the overall regulation of bodily functions and the promotion of physical fitness. However, for many people, especially beginners, the challenge lies in accurately locating these acupoints, often necessitating guidance from books or experienced practitioners. In this paper, our goal is to provide users with a quick and easy means to identify the precise locations and functions of acupoints on their hands and heads, empowering them to harness the therapeutic potential of acupressure efficiently. This accessibility can aid in alleviating symptoms promptly and potentially preventing the development of diseases.

Therefore, we propose a real-time interactive healthcare system for acupoint analysis based on augmented reality (AR). Our system employs the advanced capabilities of the Jetson Nano J1010 development board, specifically its edge computing technology. Additionally, we have integrated MediaPipe technology, enabling AR interaction with acupoints. Users can now visualize the exact locations of these acupoints on their own bodies, enhancing their understanding and application of acupressure techniques.

Beyond this, we have developed a user-friendly mobile app that seamlessly connects to a smart mirror via Bluetooth. This app allows users to select specific symptoms they wish to address. Once chosen, the device provides real-time feedback by highlighting the relevant acupoints that can offer relief for those specific symptoms. This intuitive interface simplifies the acupressure process, making it accessible to a broader audience. However, our system's capabilities do not stop at visual aids and symptom management. We have taken it a step further by incorporating state-of-the-art AI diagnosis technology using receive, appreciate, summarize, ask (RASA) and bidirectional encoder representations from transformers (BERT). This means that the system can analyze user input, identify potential diseases or symptoms, and offer personalized recommendations for acupoints that can provide targeted symptom relief. We hope that by using this proposed system, users will feel like they have a virtual healthcare advisor at their disposal.

## 2. Related Work

### 2.1 Acupoints recognition and visualization

Acupoints are essential elements in traditional Chinese healthcare, and in recent years, AI technology has made significant progress in acupoints recognition. Early approaches utilized convolutional neural networl (CNN) models and mathematical calculations or edge detection to infer acupoints locations,(1,2) while current models such as MediaPipe can obtain human body coordinate points. Previous studies have used 3D morphable models (3DMMs) to display facial acupoints,(3) but their complex computational requirements limit real-time operation. MediaPipe Face Mesh is a real-time facial landmark detection system that uses machine learning to predict the 3D positions of 468 facial keypoints. Compared with 3DMMs, MediaPipe performs better and is suitable for running on various devices. Recent research has also used MediaPipe for facial acupoint display(4) but with limited functionality.

In this study, we further improve on the foundation of MediaPipe, implementing not only facial acupoint display but also hand acupoints, acupoints classification, and gesture interaction, providing users with a more comprehensive experience. In recent years, AI has made significant progress in various fields such as natural language processing (NLP), image recognition, visual retrieval, and medical image analysis.(5,6) Especially in the past few years, with the rapid improvement of computer specifications and the growth of talents in various domains, the power of AI has been strengthening year by year. In the following sections, we will introduce the relevant technologies used in this research.

### 2.2 RASA natural language understanding (NLU)

RASA is the primary tool proposed by Bocklisch *et al.* for understanding and interpreting semantic meaning.(7) RASA Core takes the output of RASA NLU and applies machine learning models to generate reply messages for virtual assistants. By analyzing keywords entered by the user, RASA effectively determines the user's intent, enabling the AI chatbot to provide appropriate responses. This approach allows users to interact with the chatbot and receive relevant information on the basis of their inputs. The implementation demonstrated how RASA can be employed to interpret user intentions and facilitate meaningful interactions within the context of a serious game designed for intraosseous access training.

In recent years, many studies have used RASA to design chatbots and apply them in many fields.(8–10) There is also research on integrating chatbots into social networks,(11) web pages,(12) and serious games.(13) Gupta *et al.* proposed a RASA-based chatbot for disease detection.(14) They used the RASA chatbot to collect user-provided symptoms and utilized machine learning classification models, such as the support vector machine (SVM) and Naive Bayes classifiers, to identify potential illnesses. Their YAML ain't markup language (YAML) system architecture effectively integrated natural language processing techniques, and testing with a symptom dataset demonstrated the chatbot's capability to accurately detect diseases on the basis of user inputs.

In this paper, we use RASA for initial symptom assessment. Different from the above, we enhance the diagnostic accuracy by incorporating BERT after RASA's preliminary judgment. BERT allows us to determine the specific disease associated with the user's symptoms in greater detail. Additionally, our system can recommend appropriate acupoints for the user to press, providing a more comprehensive and precise consultation experience.

### 2.3 BERT

BERT is a pretrained deep learning model for NLP developed by Google,(15) extending from the Transformers decoder.(16) BERT is capable of the bidirectional encoding of sentences, considering the contextual information of all words in the sentence. This allows BERT to exhibit excellent performance in various NLP tasks, including sentiment analysis, machine translation, keyword extraction, and question-answering systems.

The commonly used BERT architecture consists of 12 transformer layers, which is a neural network architecture that effectively learns long-range dependencies. BERT employs a technique called adaptive learning during training, enabling the model to dynamically adapt to different input lengths of sentences. Additionally, BERT utilizes residual connections and self-attention mechanisms, allowing the model to capture the relevance of individual words within sentences better. Through the BERT model, we aim to infer the user's diseases on the basis of conversations and provide corresponding acupoints displayed on the human body.

In recent years, there have been many studies applying BERT to disease detection and prediction, such as the detection of Alzheimer's disease,(17,18) crop diseases,(19,20) and Lyme disease.(21,22) In this paper, we use the BERT model to infer the user's disease on the basis of conversations and search for the corresponding acupoints on the human body through a database.

## 3. System Design

### 3.1 System architecture

In this paper, we propose a real-time interactive healthcare system for acupoint analysis based on AR, which can recognize acupoints on the user's hands and head, and assist users in learning and understanding acupoints knowledge. The system consists of four main components: the software, hardware, server, and mobile sides. The system architecture is shown in Fig. 1.

The hardware side of the system utilizes the edge computing of the Jetson Nano J1010 development board and interacts with the user through a webcam. The software side is developed using PyQt and utilizes MediaPipe to obtain the coordinates of the human body. With the help of algorithms, the obtained coordinate points from MediaPipe are mapped to the acupoints on the head and hands. This allows the display screen to show the acupoints, enabling users to learn and understand acupoint knowledge. Additionally, the system includes an Android app that users can download. Through this app, users can conduct AI consultations and record their physical condition and related information. During the consultation, the app connects to the server side to perform AI consultations. The server side utilizes NLU to interpret user inputs and determine their physical condition. The user's information is stored on the server. Doctors can log in to the server through a web interface to quickly understand the user's physical condition and related information. This design enables doctors to quickly gain understanding of the user's physical condition, improving diagnostic efficiency, and allows users to conveniently access medical

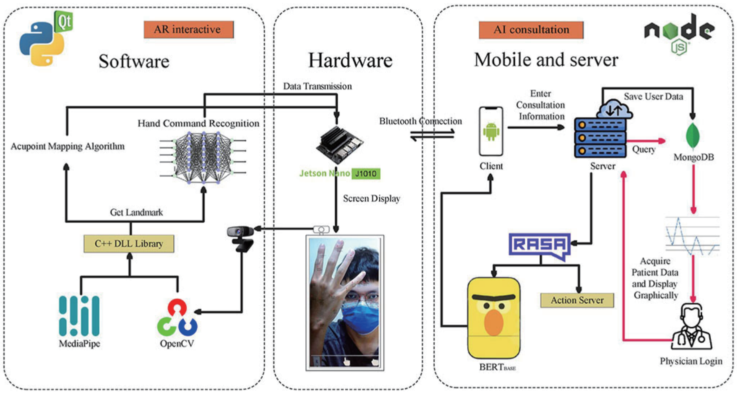


Fig. 1. (Color online) System architecture.

assistance. Overall, the developed system provides an interactive AR acupoints healthcare solution, combining hardware, software, server, and mobile components to enhance acupoints learning, AI consultations, and efficient healthcare delivery.

The system flowchart is illustrated in Fig. 2. First, we utilize a webcam to capture the user's hand data. This data is then sent to the Jetson Nano J1010 for processing. In Data Analysis, we utilize the hand and head landmark provided by the Mediapipe holistic model. Through our proposed acupoint mapping algorithm, we project the acupoints onto the human body. On the other hand, we designed a deep neural network (DNN)-based gesture recognition system that enables the user to easily operate the entire system. Users can view the desired acupoints locations or issue commands to control the product with ease. Additionally, we employ asynchronous output to provide users with a smoother visual interaction with their hand acupoints in AR.

Jetson Nano is a low-cost, low-power AI edge computing device from NVIDIA, which can provide AI system development for autonomous machines and edge computing. It is primarily targeted for creating embedded systems that require high processing power for machine learning, machine vision, and video processing applications, and is now widely used in various fields such as robotics, computer vision, intelligent image analysis, and AIoT. Since the Jetson Nano is the smallest device in the Jetson series, and owing to its low cost and low power consumption characteristics, we chose the Jetson Nano J1010 as the edge computing device for this system. MediaPipe is a multimedia machine learning model application framework developed by Google Research. It is very lightweight, making it easy to deploy on small single-board computers such as Raspberry Pi and Nvidia Jetson Nano. In practical applications of the system, the Jetson Nano mainly runs MediaPipe, MLP, and acupoint mapping algorithm calculations, which is a very ideal solution for its computing performance.

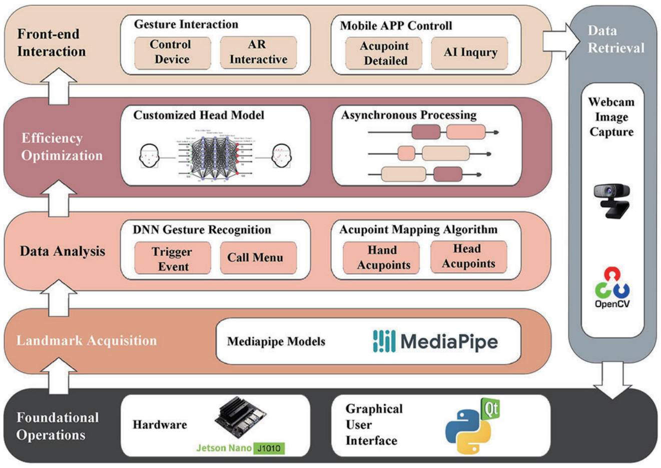


Fig. 2. (Color online) System flowchart.

### 3.2 Acupoint mapping algorithm

MediaPipe is an open-source framework designed specifically for complex perception pipelines leveraging accelerated inference (e.g., GPU or CPU) and already offers fast and accurate, yet separate, solutions for human pose, hand tracking, face landmarks, hand landmarks, and gesture recognition tasks.(23) MediaPipe Holistic provides a unified topology for groundbreaking 540+ keypoints (33 pose, 21 per-hand, and 468 face landmarks) and achieves near real-time performance on mobile devices. In this section, we describe the use of MediaPipe to detect face and hand landmarks, as shown in Fig. 3.

We use the unit "cun" as the unit of acupoint displacement.(24) Cun is a traditional Chinese unit of length (its traditional measure is the width of a person's thumb at the knuckle, whereas the width of two forefingers denotes 1.5 cun and the width of four fingers side-by-side is three cuns). We map the landmarks identified by MediaPipe to specific acupoints, as shown in Fig. 4.

Our hand and head localizations utilize position data from a database, which includes reference MediaPipe coordinate positions, relative distances to the coordinate positions, and acupoint names. The acupoint mapping flowchart is shown in Fig. 5.

To map the acupoints, we should calculate the rotation angles of the head and hands. We use the topmost landmark and the bottom landmark as the base axis to calculate the rotation angles,

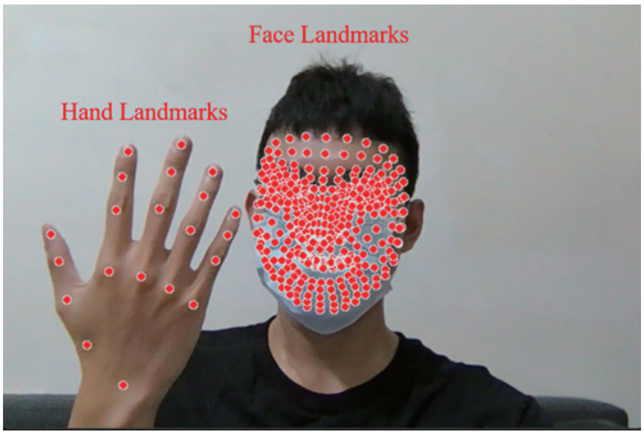


Fig. 3. (Color online) Example of face and hand landmarks.

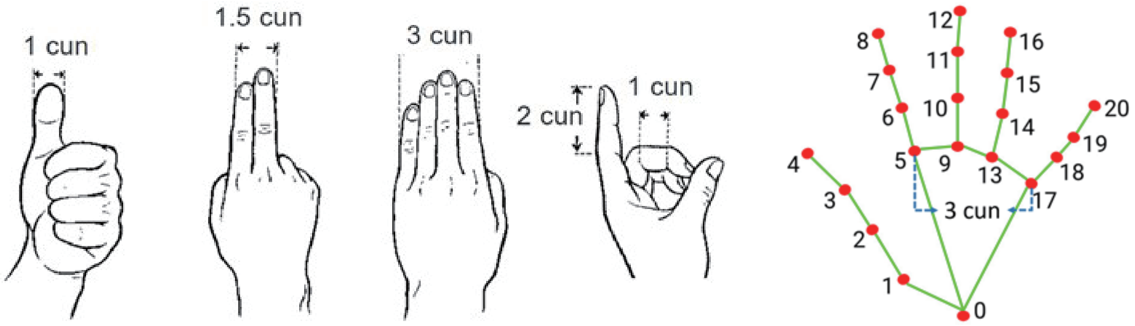


Fig. 4. (Color online) Example of cun unit measurement and landmark mapping.

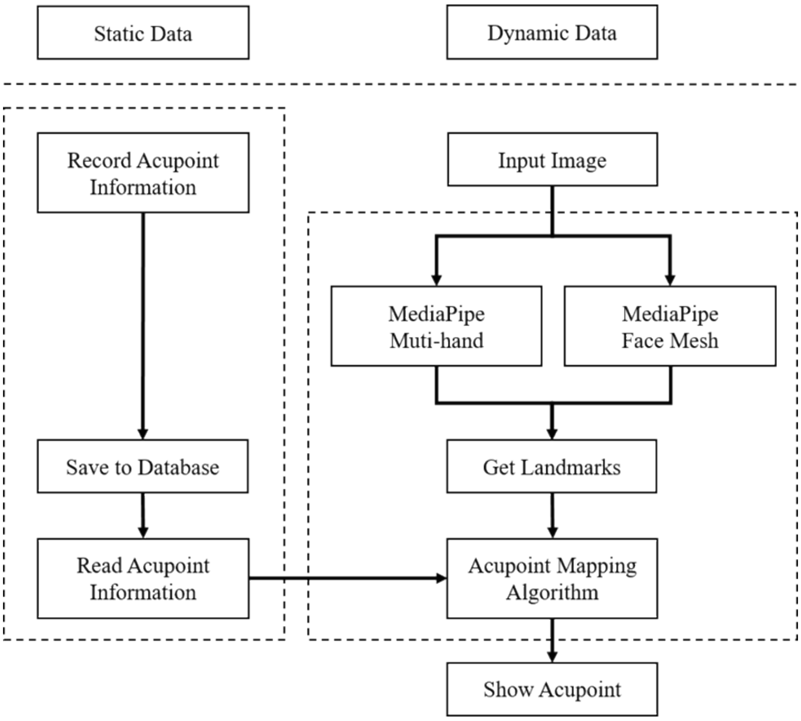


Fig. 5. Acupoint mapping flowchart.

as shown in Figs. 6(a) and 6(b). We also adjusted the distance of the acupoints according to the bones of different people, as shown in Fig. 6(c). Since the acupoints on the front and back sides of the human hand are different, to display the acupoints correctly, we take the bottom of the palm, the thumb, and the pinky of the three points of hand landmarks 0, 5, and 17 to create two vectors and calculate their normal vectors, which are used to determine the front and back sides of the hand, as shown in Fig. 6(d).

Our proposed acupoint mapping algorithm is shown in the following equations. The rotation angles of the head and hand are calculated using Eq. (1) through the base axis. The front or back side of the hand is determined by calculating the normal vector using Eq. (2). When the head or hand rotates, a transfer matrix is generated on the basis of the amount of rotation of the base axis, as shown in Eq. (3). The other acupoints can be rotated according to this transfer matrix and precisely mapped to the corresponding acupoints. An example of the hand acupoint mapping is shown in Fig. 7.

$$\varphi = \cos^{-1}\frac{\overrightarrow{v_1}\cdot\overrightarrow{v_2}}{\left\|\overrightarrow{v_1}\right\|\left\|\overrightarrow{v_2}\right\|} \tag{1}$$

$$\vec{n}=\overrightarrow{s_1}\times\overrightarrow{s_2} \tag{2}$$

$$\left[x', y'\right]=\left[x+c\cdot T_x, y+c\cdot T_y\right]\cdot\begin{bmatrix}\cos\varphi & -\sin\varphi\\ \sin\varphi & \cos\varphi\end{bmatrix} \tag{3}$$

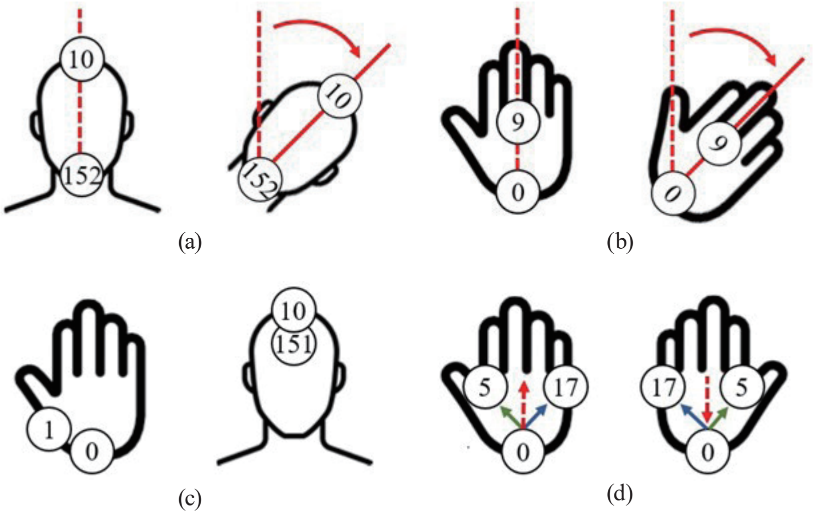


Fig. 6. (Color online) Calculation of rotation angle and palm direction: (a) head rotation, (b) hand rotation, (c) cun unit, and (d) palm direction.

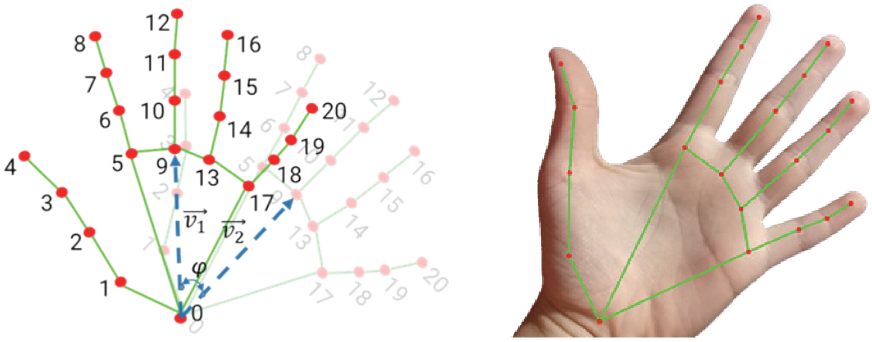


Fig. 7. (Color online) Example of hand acupoint mapping.

Here, $\varphi$ is the rotation angle, $\overrightarrow{v_1}$ is the vector of the base axis before the rotation, $\overrightarrow{v_2}$ is the vector of the base axis after the rotation, $\vec{n}$ is the normal vector that determines the front or back side of the hand. $\overrightarrow{s_1}$ and $\overrightarrow{s_2}$ are the vectors of the hand landmarks 0 to 5 and 0 to 17, respectively. $x'$ and $y'$ represent the coordinate points after acupoint mapping, $x$ and $y$ represent the coordinate points from MediaPipe, $c$ denotes the unit cun calculated from hand landmarks 5 and 6, and $T_x$ and $T_y$ represent the offsets of the acupoints relative to the landmark. Simply put, Eq. (1) is used to calculate the rotation angle, Eq. (2) is used to determine the front or back side of the hand, and Eq. (3) is used to calculate the coordinates by scaling and offsetting. Finally, by comparing the relative positioning points from the database with the hand localization points provided by MediaPipe, the acupoints can be accurately displayed on the screen and scaled on the basis of distance. The development of this customized algorithm ensures reliable acupoint mapping.

### 3.3 Hand gesture recognition

The Hand Gesture Recognition Image Dataset (HaGRID) is a dataset for hand gesture recognition images.(25) In this section, we use the data from HaGRID to extract the six types of gesture image to be used by our system, and use MediaPipe to find the hand landmarks of these images. The extracted dataset contains a total of 28,304 samples. We built a DNN model to train these data, as shown in Fig. 8. The input for this model consists of 42 features, including 21 points of landmark coordinates for each hand.

Since the input features are coordinate positions, a simple three-layer DNN model is sufficient to quickly classify the current gesture into its corresponding type. The output layer uses the Softmax function and consists of six classes: menu, select, cancel, right, left, and no gesture. The hand gesture recognition functionality enables users to interact more conveniently with the device, enhancing the overall user experience of the product.

### 3.4 Mobile-based remote control

We provide Bluetooth connectivity between the mobile device and our system, where the system is the server side and the mobile device is the client side, as shown in Fig. 9. Through Bluetooth pairing, users can conveniently view the corresponding acupoints for their symptoms and access detailed information about the desired acupoints. When the user activates the Bluetooth pairing function, the mobile device acts as an RFCOMM server and generates a specific UUID for identification during the connection. This UUID, along with the device name, is encoded into a QR code and displayed on the mobile device. The user simply needs to place the QR code displayed on the mobile device within the field of view of the webcam. The Jetson

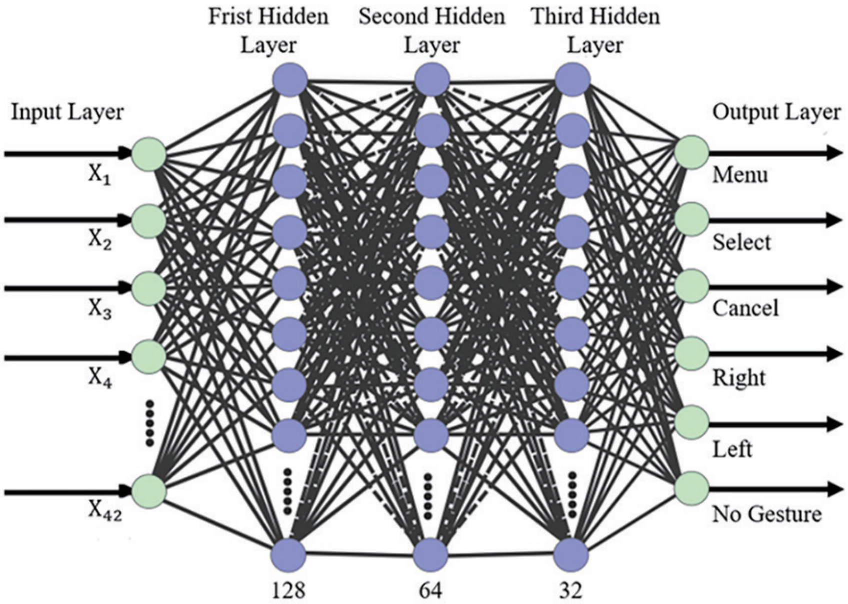


Fig. 8. (Color online) DNN model architecture of hand gesture recognition.

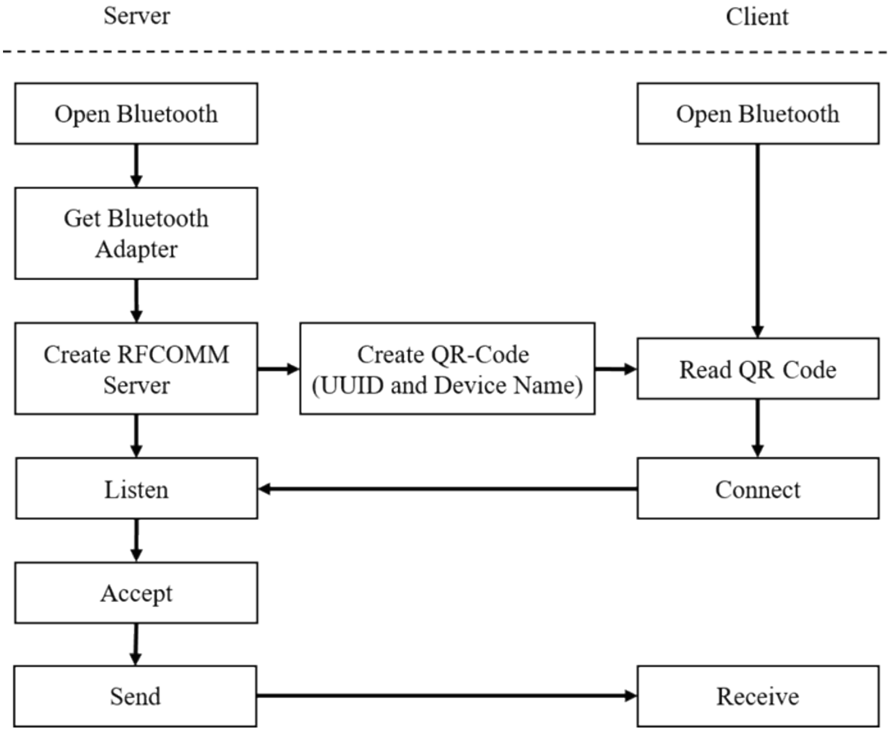


Fig. 9. Bluetooth connection flowchart.

Nano, equipped with built-in Bluetooth functionality, scans and decodes the QR code, thereby obtaining the device's MAC address and server UUID. The Bluetooth pairing functionality transforms the mobile device into a convenient control interface, allowing users to effortlessly operate and access the desired acupoints information. This design not only provides a more intuitive and interactive interface but also enhances usability by increasing convenience.

### 3.5 Interactive user interface (UI)

We input information into the system as described in Sects. 3.3 and 3.4 above, which is transmitted to the main thread of the system through event functions. This design helps prevent race conditions and critical sections between different threads, ensuring the proper handling of data. Once the main thread is reached, the system evaluates the output. This evaluation may involve operations based on predefined logic or conditions. On the basis of the evaluation result, the state will execute the corresponding method. The methods may involve updating the display content of the system, adjusting system settings, or performing other relevant operations. If specific transition conditions are met, the state instructs the state machine to perform a state transition. State transitions may involve entering another state or executing different operational flows. Additionally, the state can also notify the UI controller to switch UIs. This allows the screen display to change on the basis of the system's requirements and state variations. The system control architecture is shown in Fig. 10. The state can also send events to other threads.

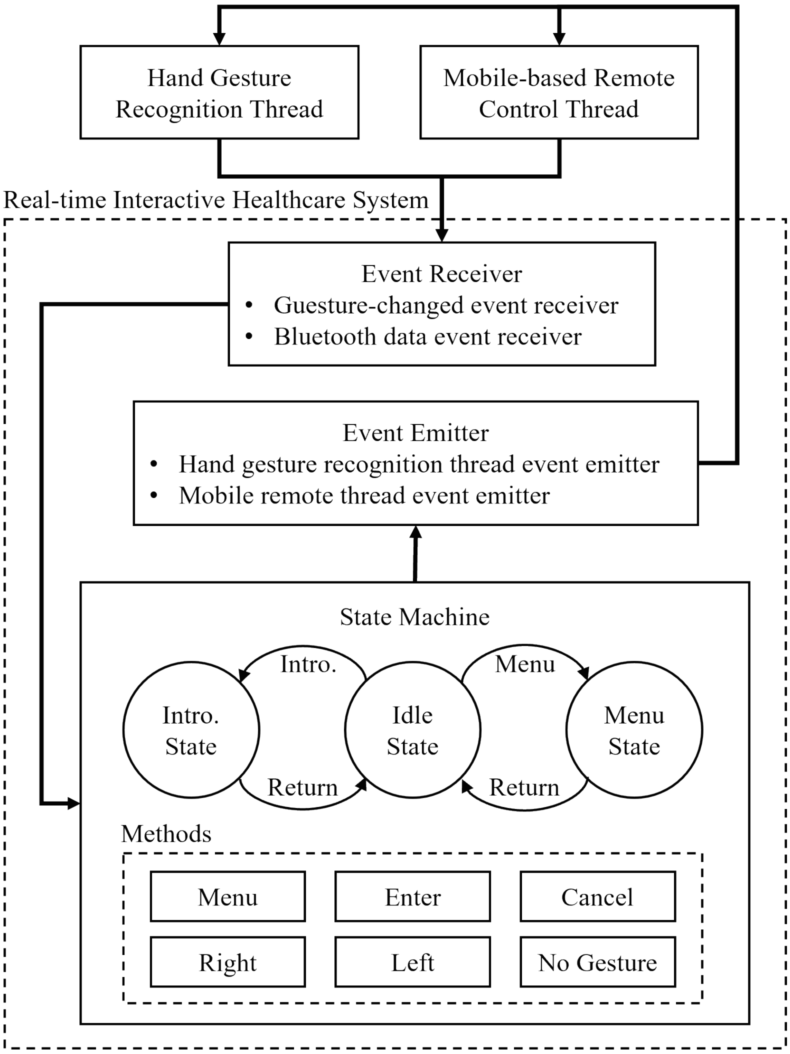

Hand Gesture Recognition Thread
Mobile-based Remote Control Thread
Real-time Interactive Healthcare System
Event Receiver
• Guesture-changed event receiver
• Bluetooth data event receiver
Event Emitter
• Hand gesture recognition thread event emitter
• Mobile remote thread event emitter
State Machine
Intro.
Menu
Intro. State
Idle State
Menu State
Return
Return
Methods
Menu
Enter
Cancel
Right
Left
No Gesture

Fig. 10. System control architecture with the interactive UI.

These events can trigger specific operations in other threads or perform specific tasks. For example, these events can be used to control the model's activation, notifying relevant threads to execute model-related operations.

### 3.6 BERT-based AI consultation

We used BERT to build our AI consultation server and constructed a RASA chatbot on the server as shown in Fig. 11. When a user asks about a disease or symptom with an intention on a mobile device, the trained BERT model will provide corresponding effective acupoints on the basis of the user's inputs and display these points via AR, allowing the user to further alleviate their symptoms through acupressure.

For the design of the AI consultation function, we use the BERT pretrained model of BERT-Base-Chinese to build our BERT-based AI consultation.(26) Google provides two basic models of

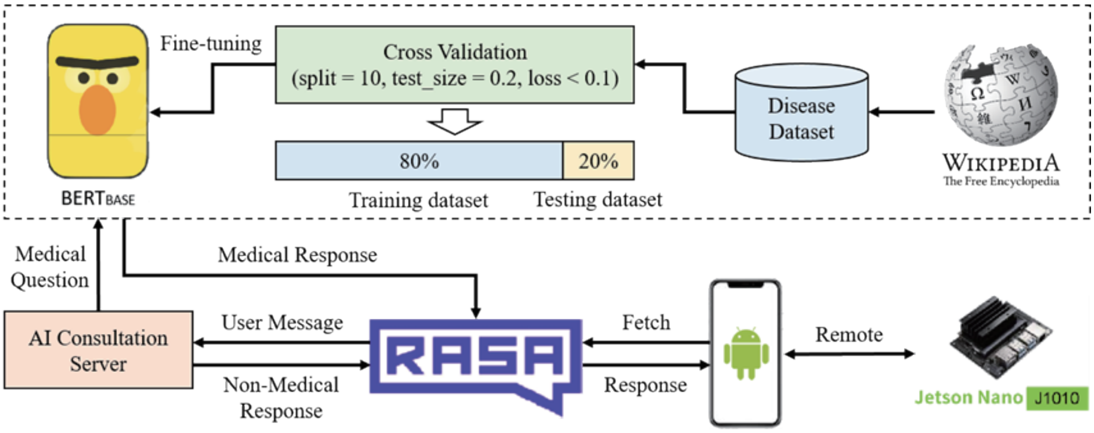


Fig. 11. (Color online) Architecture of BERT-based AI consultation.

BERT: BERT-Base and BERT-Large. The BERT-Base-Chinese pretraining model proposed by Google uses the BERT-Base version of the framework, and owing to too many BERT-Large parameters, the pretraining cost is too high, so in most experiments, the BERT-Base Chinese version of the model structure is used. This BERT-Base-Chinese model consists of 12 layers, 768 hidden units, and 12 attention heads, with a parameter size of 110 MB. Before training the model, we initialized the token input of the BERT model and incorporated data collected from Wikipedia. Our dataset included dialogue information associated with 109 disease categories and symptoms related to these diseases, totaling 623 records. We used this data to fine-tune the model, which involved connecting a new simple classifier to the last layer of BERT to identify the disease category on the basis of the user's input.

## 4. Experimental Results

### 4.1 System demonstration

In this paper, we propose a real-time interactive healthcare system for acupoint analysis based on AR. Our system utilizes the Jetson Nano J1010 development board with edge computing technology as shown in Fig. 12. Users can interact with the acupoints on their own bodies through the AR technology and enhance their understanding and application of acupressure techniques. Additionally, by scanning a QR code, the device pairs with a mobile phone via Bluetooth. This allows users to view acupoints related to specific symptoms on their mobile phone or access detailed information about each acupoint.

With our proposed acupoint mapping algorithm and AR technology, we can ensure that the acupoints are accurately displayed on the screen as shown in Fig. 13, and those on the head are not obscured by a mask. The system recognizes the user's gesture and operates the system through our proposed hand gesture recognition as shown in Fig. 14. At the same time, the system interacts with the acupoints and views their functional descriptions through AR technology as shown in Fig. 15.

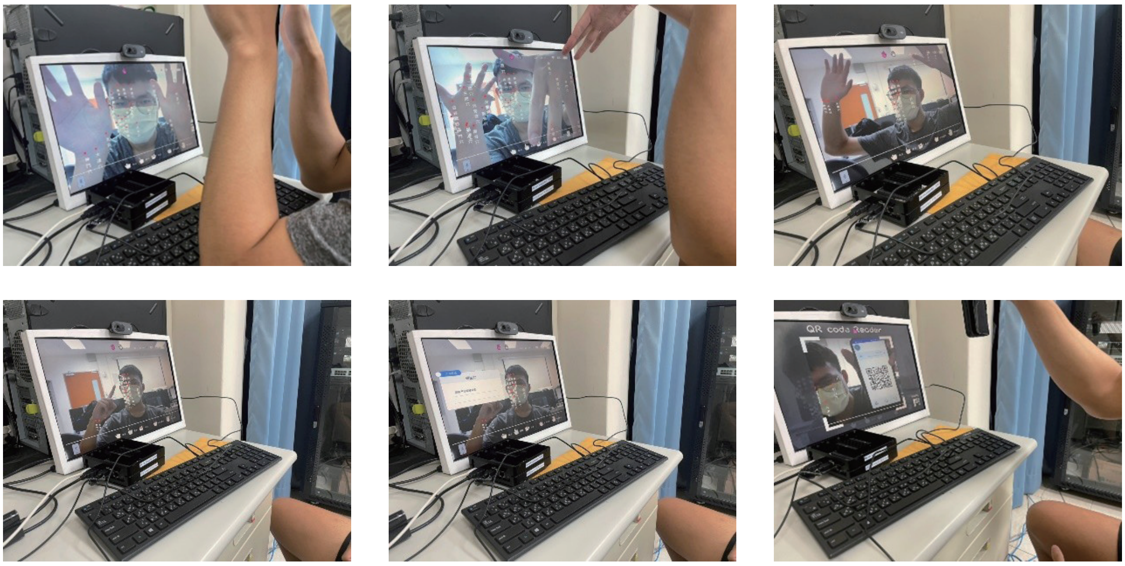


Fig. 12. (Color online) System demonstration with Jetson Nano J1010.

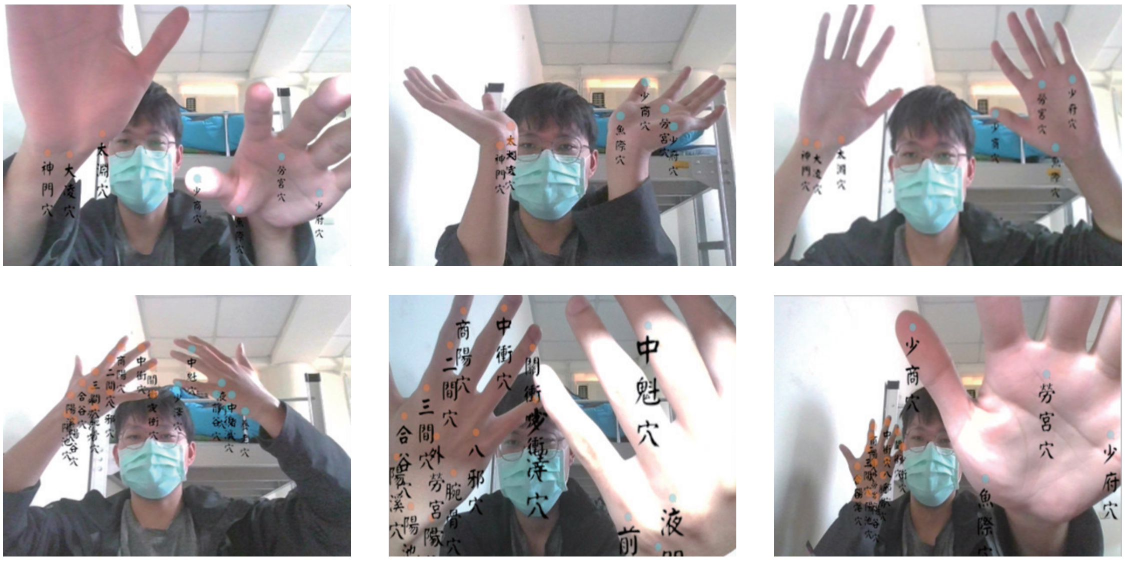


Fig. 13. (Color online) Examples of acupoint display based on AR.

The proposed BERT-based AI consultation requires both responses and inputs to be in Chinese. Users can perform AI consultation through the app, which queries the RASA server to obtain relevant information and updates the database with the consultation details. If the BERT model is uncertain about the disease based on the entered symptoms, the RASA will return the top three possible diseases associated with those symptoms, as shown in Fig. 16.

The results of the AI consultation are stored in a cloud database, and the physician can view the patient's data by accessing the database, as shown in Fig. 17. We categorize the user's symptoms and organize symptom information into graphical information. This graphical

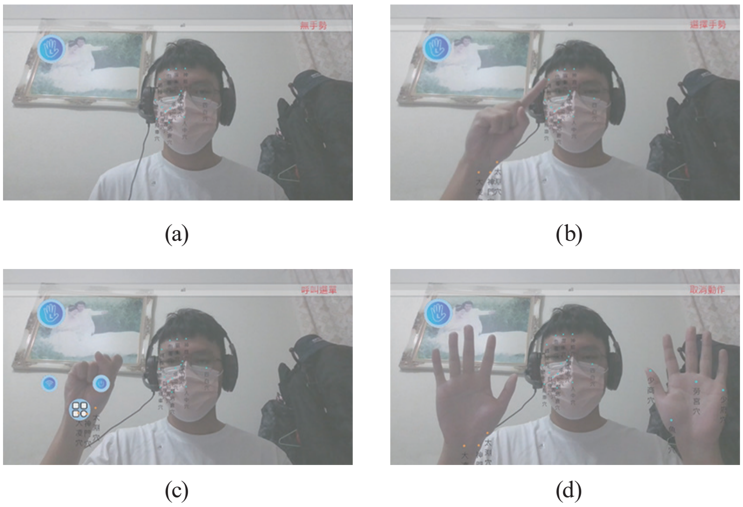


Fig. 14. (Color online) Demonstration of system operation on the proposed hand gesture recognition: (a) no gesture, (b) select gesture, (c) menu gesture, and (d) cancel gesture.

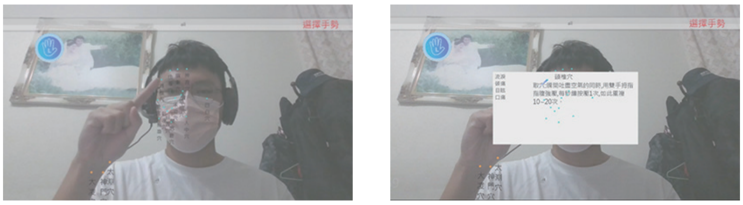
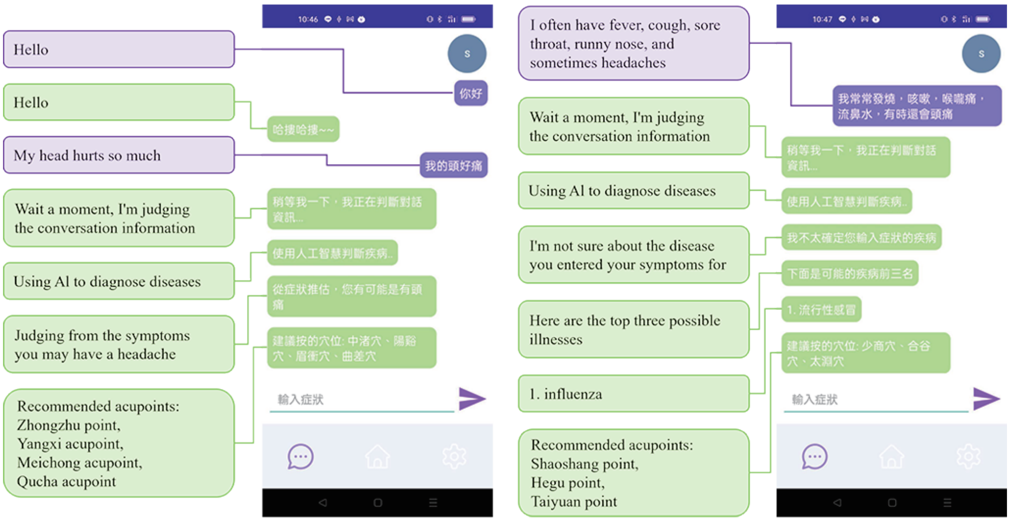


Fig. 15. (Color online) Demonstration of AR-based interaction with acupoints.



Hello

你好

Hello

哈摟哈摟~~

My head hurts so much

我的頭好痛

Wait a moment, I'm judging the conversation information

稍等我一下，我正在判斷對話資訊...

Using AI to diagnose diseases

使用人工智慧判斷疾病..

Judging from the symptoms you may have a headache

從症狀推估，您有可能是有頭痛

Recommended acupoints: Zhongzhu point, Yangxi acupoint, Meichong acupoint, Qucha acupoint

建議按的穴位: 中渚穴、陽谿穴、眉衝穴、曲差穴

輸入症狀

I often have fever, cough, sore throat, runny nose, and sometimes headaches

我常常發燒，咳嗽，喉嚨痛，流鼻水，有時還會頭痛

Wait a moment, I'm judging the conversation information

稍等我一下，我正在判斷對話資訊...

Using AI to diagnose diseases

使用人工智慧判斷疾病..

I'm not sure about the disease you entered your symptoms for

我不太確定您輸入症狀的疾病

Here are the top three possible illnesses

下面是可能的疾病前三名

1. influenza

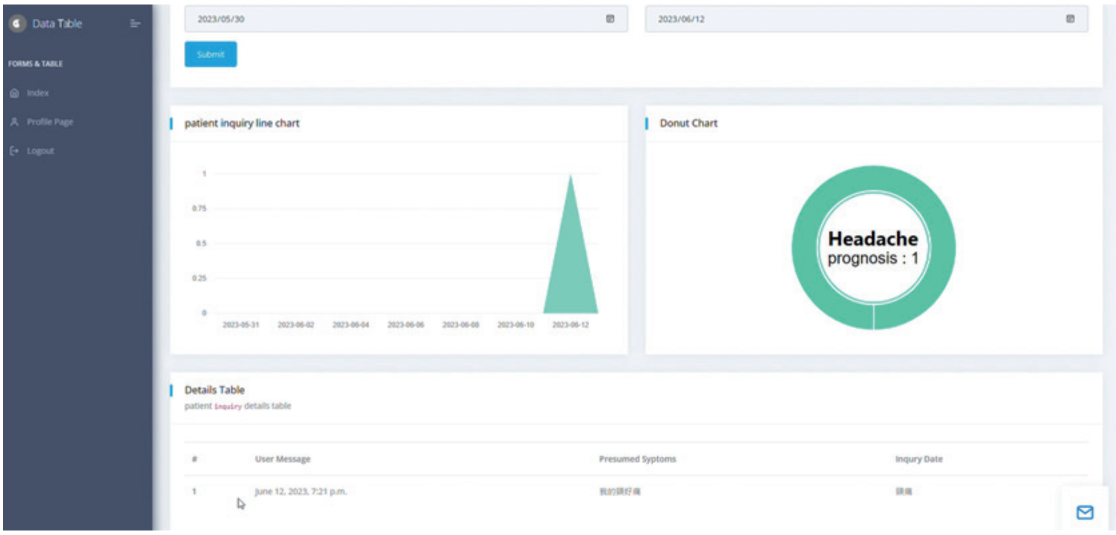
1. 流行性感冒

Recommended acupoints: Shaoshang point, Hegu point, Taiyuan point

建議按的穴位: 少商穴、合谷穴、太淵穴

輸入症狀

Fig. 16. (Color online) Demonstration of BERT-based AI consultation with RASA.



2023/05/30

2023/06/12

Details Table

patient inquiry details table

| # | User Message | Presumed Syptoms | Inqury Date |
|---|---|---|---|
| 1 | June 12, 2023, 7:21 p.m. | 我的頭好痛 | 頭痛 |

Fig. 17. (Color online) Cloud data management platform for AI consultation.

information provides the physician with a quick overview of the patient's current condition, which saves the physician time in organizing and analyzing the data and provides a more intuitive and clear visual presentation.

## 4.2 Optimization of acupoint mapping algorithm

Our proposed acupoint mapping algorithm is based on MediaPipe and applied to hand and head acupoint mapping. Taking hand landmark detection as an example, the hand landmark detection provided by MediaPipe uses a single-shot detector model called BlazePalm, which achieves an average accuracy of 95.7% in palm detection. Its training data was manually annotated from ~30 K real-world images with 21 3D coordinates, rendering a high-quality synthetic hand model over various backgrounds and mapped to the corresponding 3D coordinates. Therefore, it can ensure that the hand landmarks can be correctly found under different backgrounds and light sources. The effectiveness of the acupoint mapping algorithm is shown in Fig. 18. The hand acupoints adhere to fixed positions and do not shift owing to hand rotation or other gestures. This demonstrates the accuracy and effectiveness of our acupoints mapping algorithm.

Although the MediaPipe Face Mesh can provide 468 features, which are sufficient for us to quickly correspond them to the face acupoints, this method also consumes a lot of time for computing performance. Therefore, we would like to propose an optimized way to save more computing resources and improve the system speed under the edge computing based on real-time computing. In this section, we propose an accelerated detection of face acupoints. We use the six features provided by the MediaPipe Face Detector and map these features to face acupoints by training a DNN model as shown in Fig. 19.

The input of the DNN model for accelerated acupoint mapping consists of six features, including the landmarks of the right eye, left eye, nose tip, mouth center, right ear tragion, and

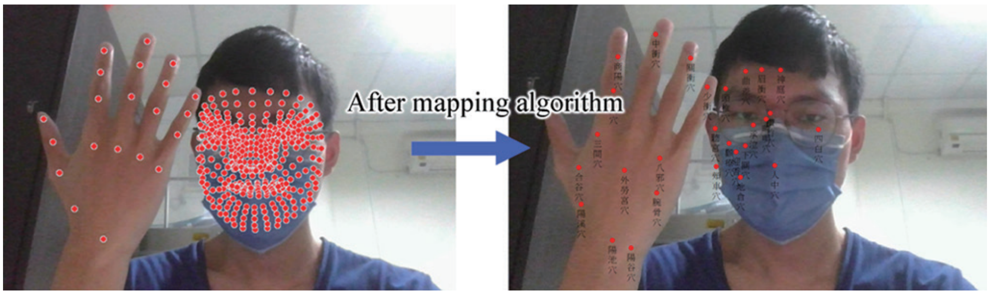


Fig. 18. (Color online) Example of acupoint mapping algorithm.

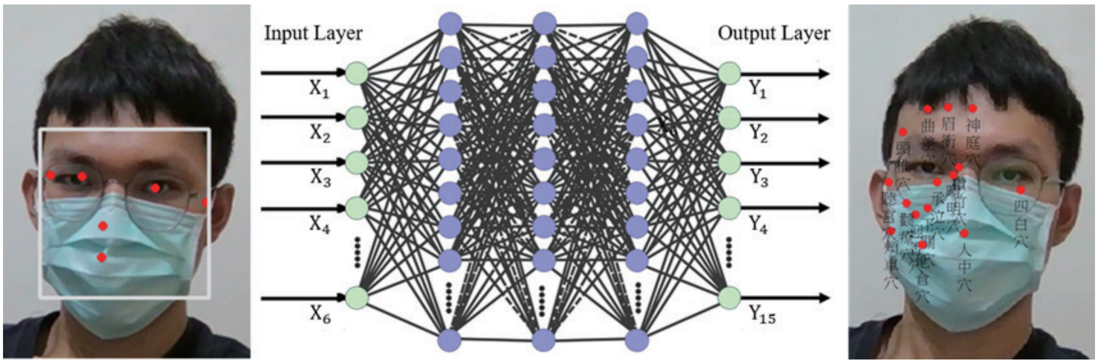


Fig. 19. (Color online) Example of accelerated acupoint mapping.

left ear tragion, while the output is the 15 points of the facial acupoints. As mentioned above, we recorded 468 mapping data to train this model. The model parameters are 150 epochs, 16 batch sizes, and a learning rate of 0.001, and use the categorical cross-entropy loss function in Eq. (4). We use the mean absolute error (*MAE*) metric to calculate the output error. The results of the network training are 1.314e−4 for the loss and 6.582e−3 for the *MAE*, as shown in Fig. 20, which is about 0.02 cun on average after our calculation. As the calculation of facial feature points is reduced from 468 to 6, the computing speed of the system is significantly improved, as shown in Table 1. Although this method can significantly increase the computing speed of the system, it also reduces the accuracy slightly, so we have designed the system to allow users to choose which method they want to use for acupoint mapping. We hope that by proposing this method, users will be able to choose the best way to use the system according to their own needs.

$$Loss = -\sum_{i=1}^{n} y_i \cdot \log\left(\hat{y}_i\right) \tag{4}$$

Here, $n$ is the number of classes, $y_i$ is the truth label, and $\hat{y}_i$ is the output for the $i$th class.

Our system uses a general webcam for video capture. General webcams have built-in automatic light correction to adapt to various light sources, so the environmental lighting has little effect on system recognition. Since the image size input to MediaPipe is 480×320, which is

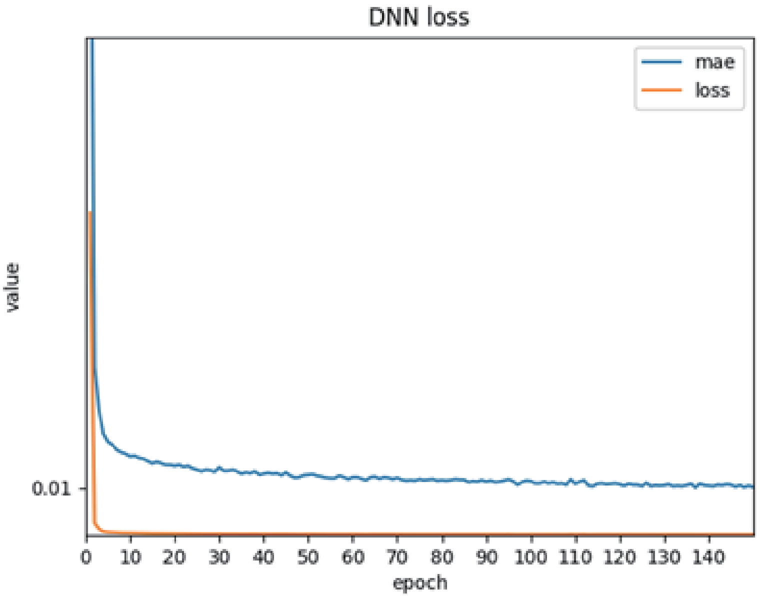


Fig. 20. (Color online) Loss curve during DNN model training.

Table 1
Performance comparison of accelerated acupoint mapping.

| Acupoint mapping | Face mesh | Face detector + DNN |
|---|---|---|
| Time per frame (TPF) | 1.96e−1 | 5.9e−2 |
| Frames per second (FPS) | 5.1 | 16.9 |

much smaller than the resolution of general webcams, the camera resolution has also little impact on system recognition. On the other hand, MediaPipe uses a large number of images with complex backgrounds during training, so it has a certain robustness in recognizing complex backgrounds.

### 4.3 Model training of hand gesture recognition

We used the network architecture and dataset from Sect. 3.3 to train the DNN model for hand gesture recognition with parameters of 150 epochs, 16 batch sizes, and a learning rate of 0.001, and used the categorical cross-entropy loss function. By observing the loss function during training, we can see a gradual decrease in loss and a corresponding increase in accuracy, indicating good learning performance of the model, as shown in Fig. 21. The DNN hand gesture recognition model in this project achieved an impressive accuracy rate of up to 90%. This indicates that our model can accurately classify the user's gestures into six predefined types (menu, select, cancel, right, left, no gesture). Such high accuracy provides reliable hand gesture recognition functionality for our application, enabling users to perform corresponding operations smoothly. Monitoring the loss function and accuracy ensures that the training process and results of the model meet the intended goals. This technology exhibits high accuracy in hand

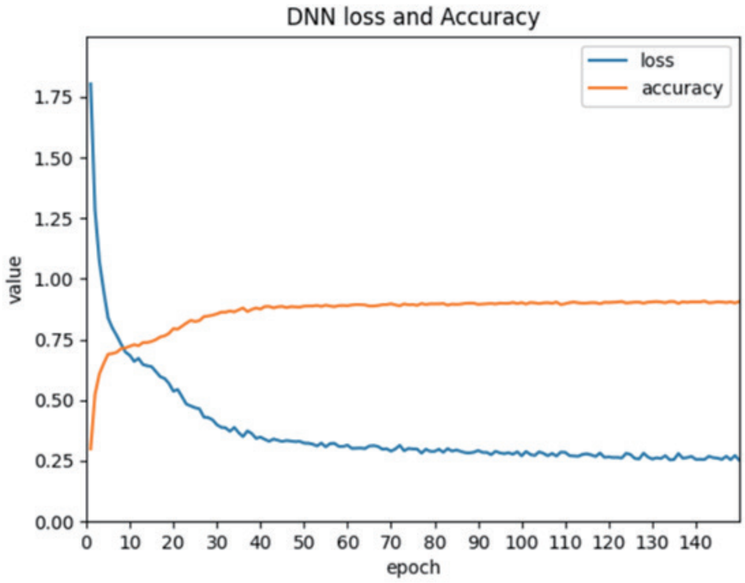


Fig. 21. (Color online) Loss curve during DNN model training.

gesture recognition, providing a solid foundation for the successful implementation of this project.

### 4.4 Model training of BERT-based AI consultation

We used the BERT pretrained model and dataset from Sect. 3.6 to train the BERT network for our AI consultation with parameters of 72 epochs and 10 batch sizes. Our disease dataset has a total of 623 records, which includes dialogue information associated with 109 disease categories and symptoms related to these diseases. Table 2 shows examples from our disease dataset. We divided the dataset into an 80% training set and a 20% testing set to fine-tune the BERT model. By observing the variation of the loss function in Fig. 22, we can see a gradual decrease in the loss function, indicating an improvement in the model's learning performance.

We set the training to terminate when the loss function is lower than 0.1. During the training, we recorded four model metrics: accuracy, precision, recall, and F1 score, as shown in Fig. 23. To ensure the credibility of the experiment, we performed 10-fold cross-validation, averaging the model metrics from the ten iterations as the final performance indicators. This approach helps reduce the impact of random variations in individual training sessions and provides a more comprehensive evaluation of the model's performance. Our results showed that our model performed impressively in disease prediction, achieving an accuracy rate of nearly 80%. This means that our model can provide accurate predictions for the symptoms and disease categories entered by the user.

Table 2
Examples from the disease dataset.

| Category | Related symptoms |
|---|---|
| Influenza | Fever, cough, sore throat, runny nose, muscle aches, fatigue, and headaches |
| Influenza | Vomiting and diarrhea |
| Acute pharyngitis | Sore throat, painful swallowing, and dysphagia |
| Heatstroke | Dizziness, headache, thirst, shortness of breath, confusion |

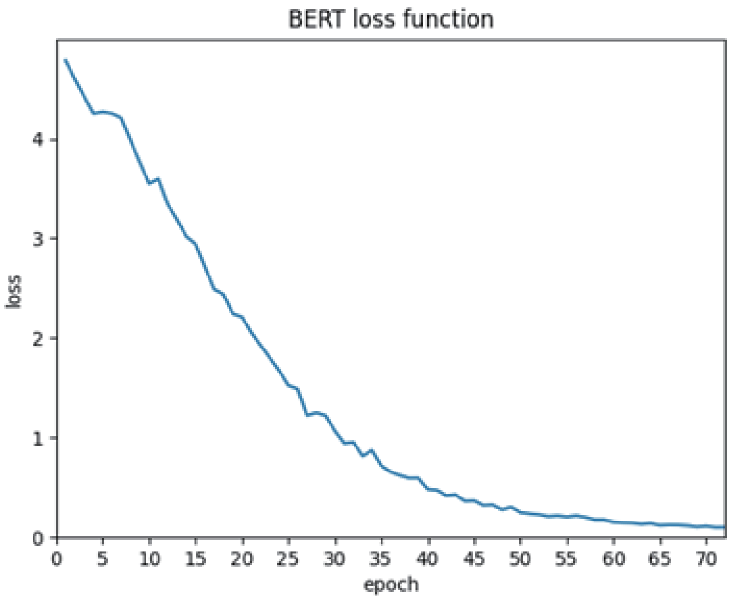


Fig. 22. (Color online) Loss curve during the BERT model training.

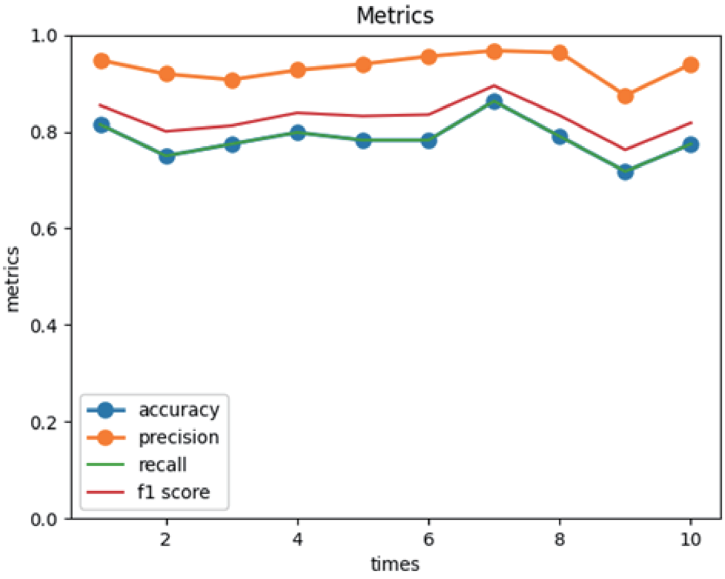


Fig. 23. (Color online) Loss curve during the DNN model training.

## 5. Conclusions

In this paper, we proposed a real-time interactive healthcare system for acupoint analysis based on AR. We utilized the edge computation on the Jetson Nano J1010 development board to display the acupoints on the screen, using our proposed DNN-based hand gesture recognition and acupoint mapping algorithms, to interact with the acupoints through AR technology. We also developed a mobile application that can interact with the system via Bluetooth technology. In addition, we developed a BERT-based AI consultation to determine possible diseases or symptoms and provided corresponding recommended acupoints for symptom relief. All diagnostic information was stored on the server, and doctors can view user information through the website, further saving medical resources. On the basis of our experimental results, we discussed the accuracy of disease diagnosis and analyzed the performance of each network. The results showed high performance and accuracy of our proposed system. We hope that the proposed system can provide a safe, convenient, and efficient solution for personal healthcare management.

## References

1 G. S. Hsu, H. C. Peng, and K. H. Chang: Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (2014) 34–39.

2 M. Chang and Q. Zhu: Proc. 2017 5th Int. Conf. Frontiers of Manufacturing Science and Measuring Technology (FMSMT 2017, Atlantis Press) (2017) 545–549.

3 Y. Z. Chen, C. Maigre, M. C. Hu, and K.-C. Lan: Proc. 8th ACM on Multimedia Systems Conf. (2017) 239–241.

4 M. Zhang, J. Schulze, and D. Zhang: arXiv preprint. https://arxiv.org/abs/2111.14755

5 B. Sun, L. Yang, W. Zhang, P. Dong, C. Young, J. Dong, and M. Lin: Proc. 2019 IEEE Int. Conf. Multimedia & Expo Workshops (ICMEW) (IEEE, 2019) 611–611.

6 A. S. Panayides, A. Amini, N. D. Filipovic, A. Sharma, S. A. Tsaftaris, A. Young, D. Foran, N. Do, S. Golemati, T. Kurc, K. Huang, K. S. Nikita, B. P. Veasey, M. Zervakis, J. H. Saltz, and C. S. Pattichis: IEEE J. Biomed. Health. Inf. **24** (2020) 1837.

7 T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol: arXiv preprint arXiv:1712.05181 2017.

8 A. Jiao: Proc. J. Physics: Conference Series (IOP Publishing) **1487** (2020) 012014.

9 T. N. T. Mai and S. Maxim: Int. J. Open Inf. Technol. **9** (2021) 31.

10 T. M. N. Pham, T. N. T Pham, H. P. T. Nguyen, B. T. Ly, T. L. Nguyen, and H. S. Le: J. Asian Financ. Econ. Bus. **9** (2022) 273.

11 Y. Windiatmoko, R. Rahmadi, and A. F. Hidayatullah: Proc. IOP Conference Series: Materials Science and Engineering (IOP Publishing) **1077** (2021) 012060.

12 L. Fauzia, R. B. Hadiprakoso, and Girinoto: Proc. 2021 4th Int. Seminar on Research of Information Technology and Intelligent Systems (ISRITI) (IEEE, 2021) 373–378.

13 F. Stancati, A. Riitano, B. Kapralos, D. Button, A. Dubrowski, and F. Lamberti: Proc. 2023 14th Int. Conf. Information, Intelligence, Systems & Applications (IISA) (IEEE, 2023) 1–4.

14 V. Gupta, A. Sood, and T. Singh: Proc. 2022 Int. Mobile and Embedded Technology Conf. (MECON) (IEEE, 2022) 94–100.

15 J. Devlin, M. W. Chang, K. Lee, and K. Toutanova: arXiv preprint. https://arxiv.org/abs/1810.04805

16 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin: Adv. Neural Inf. Process. Syst. (2017) 30.

17 A. Balagopalan, B. Eyre, F. Rudzicz, and J. Novikova: arXiv preprint. https://arxiv.org/abs/2008.01551

18 C. Mao, J. Xu, L. Rasmussen, Y. Li, P. Adekkanattu, J. Pacheco, B. Bonakdarpour, R. Vassar, L. Shen, G. Jiang, F. Wang, J. Pathak, and Y. Luo: J. Biomed. Inf. **144** (2023) 104442.

19 W. Zhang, C. Wang, H. Wu, C. Zhao, G. Teng, S. Huang, and Z. Liu: Agronomy **12** (2022) 2130.

20 J. Ding, B. Li, C. Xu, Y. Qiao, and L. Zhang: Appl. Intell. **53** (2023) 15979.

21 E. K. E. Laison, M. Hamza Ibrahim, S. Boligarla, J. Li, R. Mahadevan, A. Ng, V. Muthuramalingam, W. Y. Lee, Y. Yin, and B. R. Nasri: J. Med. Internet Res. **25** (2023) e47014.

22 T. Susnjak: Borrelia Burgdorferi: Methods and Protocols (Springer, 2024) 173–183.
23 C. Lugaresi, J. Tang, 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, and M. Grundmann: arXiv preprint. https://arxiv.org/abs/1906.08172
24 M. Coyle, M. Aird, D. Cobbin, and C. Zaslawski: Acupuncture in Medicine **18** (2000) 10.
25 A. Kapitanov, K. Kvanchiani, A. Nagaev, R. Kraynov, and A. Makhliarchuk: Proc. IEEE/CVF Winter Conf. Applications of Computer Vision (2024) 4572–4581.
26 Z. Yang, Z. Zhixiong, L. Huan, and D. Liangping: Data Anal. Knowl. Discovery **4** (2020) 41.

## About the Authors

**Yu-Xiang Zhao** received his B.S. and M.S. degrees in Electrical Engineering from Tamkang University, Taiwan, in 2000 and 2002, respectively, and his Ph.D. degree in Computer Science and Information Engineering from National Central University, Taiwan, in 2007. From 2009 to 2017, he was an assistant professor at National Quemoy University, Taiwan. Since 2017, he has been an associate professor. His research interests are in deep learning, machine learning, and image processing. (yxzhao@nqu.edu.tw)

**Chung-Yen Wei** received his B.S. degree in Computer Science and Information Engineering from National Quemoy University, Taiwan, in 2024. Since 2024, he has been pursuing M.S. degree at the Institute of Data Science and Engineering, National Yang Ming Chiao Tung University, Taiwan. His research interests are data science, deep learning, and image processing. (ericjjkkhandsome@gmail.com)

**Bo-yuan Xu** received his B.S. degree in Computer Science and Information Engineering from National Quemoy University, Taiwan, in 2024. His research interests are in deep learning and image processing.
(z22756392z10@gmail.com)