Design of Interactive System for Acupoint Analysis Based on Augmented Reality

Chung-Yen Wei, Bo-Yuan Xu, and Yu-Xiang Zhao*
National Quemoy University, Taiwan.
*E-mail: yxzhao@nqu.edu.tw

***Abstract* — In this paper, we utilize the edge computing of Jetson Nano J1010 development board and employ MediaPipe technology and acupoint mapping algorithm to display acupoints on the screen. By using a custom deep neural network (DNN) hand recognition model, we achieve augmented reality (AR) interaction with acupoints. Users can view the corresponding acupoint locations on the body and acupoint descriptions through gestures. Additionally, we develop a mobile app that enables interaction with the system via Bluetooth technology. Users can select symptoms they wish to understand, and the system will display acupoints that can alleviate those symptoms. Our team also develops an AI-based diagnostic system using the open-source conversational AI Rasa, which utilizes 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-based healthcare system offers a safe, convenient, and efficient solution for health management, particularly valuable for individuals lacking experience and medical knowledge.**

Keywords: edge computing, acupoints, AR, AI, NLU, BERT.

I. INTRODUCTION

This system utilizes the edge computing of Jetson Nano J1010 development board and incorporates MediaPipe technology to achieve augmented reality (AR) interaction with acupoints, allowing users to view the corresponding acupoint locations on the body. We have also developed a mobile app that connects to a smart mirror via Bluetooth. Users can select symptoms they wish to understand, and the screen will provide feedback on acupoints that can alleviate those symptoms. The system also includes AI diagnosis using Rasa and BERT technology, which analyzes user input to determine possible diseases or symptoms and provides recommended acupoints for further symptom relief. This acupoint-based healthcare system offers a safe, convenient, and efficient solution for health management, particularly valuable for individuals lacking experience and medical knowledge. The innovation of this project lies in the integration of the mobile app, AI, and AR technology, which realizes a acupoint-based healthcare solution that is more convenient and efficient compared to traditional healthcare approaches.

II. RELATED WORK

Acupoints are essential elements in traditional Chinese healthcare, and in recent years, artificial intelligence (AI) technology has made significant progress in acupoint recognition. Early approaches utilized CNN models and mathematical calculations or edge detection to infer acupoint locations [1]-[2], while now models such as MediaPipe can obtain human body coordinate points. Previous studies have used 3DMM models 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 to 3DMM models, 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. This paper further develops on the foundation of MediaPipe, implementing not only facial acupoint display but also hand acupoints, acupoint classification, and gesture interaction, providing users with a more comprehensive experience. In recent years, artificial intelligence has made significant progress in various fields such as natural language processing, 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 artificial intelligence has been strengthening year by year. The following sections will introduce the relevant technologies used in this research.

A. MediaPipe

MediaPipe is an open-source, cross-platform multimedia processing framework developed by Google [7]. It provides a set of templates for image and video processing, enabling the construction of complex multimedia applications. MediaPipe supports multiple platforms, including Android, iOS, Linux, and Windows. It offers various built-in algorithms such as face detection, holistic detection, and hand tracking, while also allowing easy extension with custom algorithms. MediaPipe can be used in various environments, such as mobile and embedded devices. In this project, MediaPipe is utilized as the tool for AR display. The Holistic model provided by MediaPipe allows us to obtain corresponding body pose information, and with the help of a custom DNN calculator [8], gesture recognition functionality is achieved.

B. PyQT

PyQt is a framework for developing graphical user interface (GUI) applications [9]. It is based on the Qt framework, which is a cross-platform application development framework used for desktop, mobile, and embedded systems. PyQt can run on multiple operating systems, including Windows, Mac OS X, Linux, and embedded Linux, making program development on Jetson Nano much more convenient. Due to its foundation on the Qt framework, PyQt offers excellent performance. It provides efficient drawing and event handling through optimization and design, enabling the handling of large amounts of data and complex operations. Additionally, PyQt offers rich configuration and customization options, allowing the creation of unique user interfaces.

C. Jetson Nano

The Jetson Nano J1010 development board is a high-performance single-board computer introduced by NVIDIA, designed specifically for machine learning and artificial intelligence applications. Jetson Nano J1010 features NVIDIA's quad-core ARM Cortex-A57 processor and a 128-core NVIDIA Maxwell GPU, providing powerful computational capabilities. This allows it to excel in handling complex machine learning and visual computing tasks, capable of real-time processing of high-resolution images and videos. The NVIDIA GPU on Jetson Nano J1010 provides additional computing resources for accelerating machine learning and deep learning tasks. This enables the execution of complex neural network models on Jetson Nano and facilitates real-time face detection and hand recognition tasks as proposed in this paper. J1010 is also an optimized hardware platform for AI and machine learning applications. It supports popular deep learning frameworks such as TensorFlow, PyTorch, and Caffe, as well as GPU acceleration libraries like CUDA and cuDNN, enabling efficient utilization of custom models.

D. BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained deep learning model for natural language processing (NLP) developed by Google [10], extending from the Transformers [11] decoder. BERT is capable of 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 architecture of BERT used in this project consists of 12 transformer layers (see Figure 1), 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 [11], allowing the model to capture the relevance within sentences better. Through the BERT model, we aim to infer user diseases based on conversations and provide corresponding acupoints displayed on the human body.

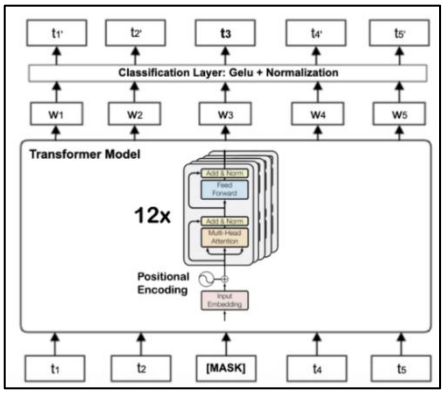

Fig. 1 BERT basic architecture diagram.

E. MVC and State Machine

This paper adopts the Model-View-Controller (MVC) architecture on the software side to organize and separate different components of the application, aiming to achieve better code structure and maintainability. The Model represents the data structure, logic, and state of the application. It is responsible for handling operations, validation, persistence, and other data-related tasks. The Model does not directly interact with the user interface (View) or user interactions (Controller) but provides interfaces for them to interact. The View is responsible for presenting the interface, allowing users to intuitively view and manipulate data. It typically represents user interface elements such as buttons, forms, charts, etc., and updates accordingly based on the state of the Model. The Controller serves as an intermediary between the Model and the View. It receives user inputs, processes user requests, and then passes the results to the Model or View as needed. It determines how to update the Model and View based on the logic of the application. In addition to the MVC architecture, this paper also utilizes a state machine to manage the different states and transitions within the application. The state machine helps define the behavior and flow of the application by representing various states and specifying the conditions and actions associated with state transitions. By adopting the MVC architecture and employing a state machine, this paper achieves a well-structured and maintainable software solution for the intended application.

III. Approach

A. System Architecture

In this paper, an interactive AR acupoint healthcare system is developed, which can recognize acupoints on the user's hands and head and assist users in learning and understanding acupoint knowledge. As shown in Figure 2. The system consists of four main components: the software side, the hardware side, the server side, and the mobile side. The hardware side utilizes the edge computing of 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 natural language understanding (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 gain a faster understanding of the user's physical condition, improving diagnostic efficiency, and allows users to conveniently access medical assistance. Overall, the developed system provides an interactive AR acupoint healthcare solution, combining hardware, software, server, and mobile components to enhance acupoint learning, AI consultations, and efficient healthcare delivery.

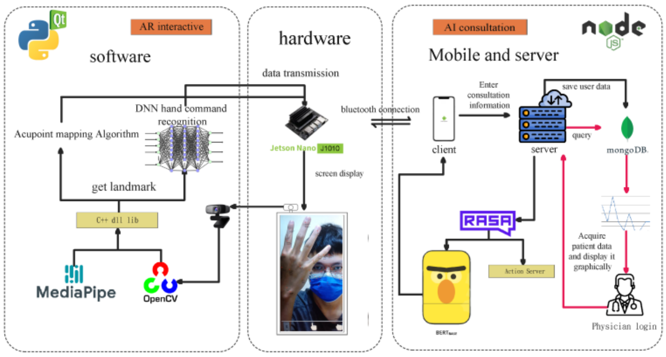

Fig. 2 System Architecture Diagram.

B. Produce Architecture

The product architecture is illustrated in Figure 3. Firstly, we utilize a webcam to capture the user's hand data. This data is then sent to the Jetson Nano J1010 for processing. We employ the MediaPipe holistic model for hand localization and obtain landmark coordinates. After data analysis, we control the output, as edge computing requires low-power and low-load solutions. We provide two facial recognition models, allowing users to choose our customized facial model for improved system performance. Additionally, we employ asynchronous output to provide users with smoother visual interaction with their hand acupoints in augmented reality.

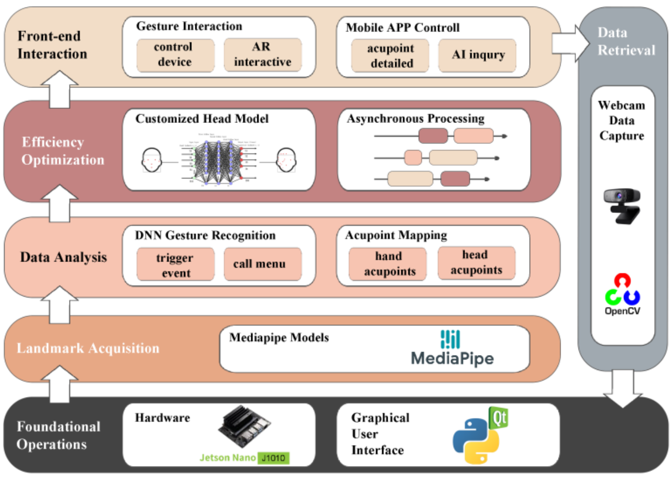

Fig. 3 Product Architecture Diagram.

C. *Acupoint Mapping*

Our hand and head localization utilize position data from a database, which includes reference MediaPipe coordinate positions, relative distances to the coordinate positions, and acupoint names. The steps for displaying acupoints can be seen in Figure 4.

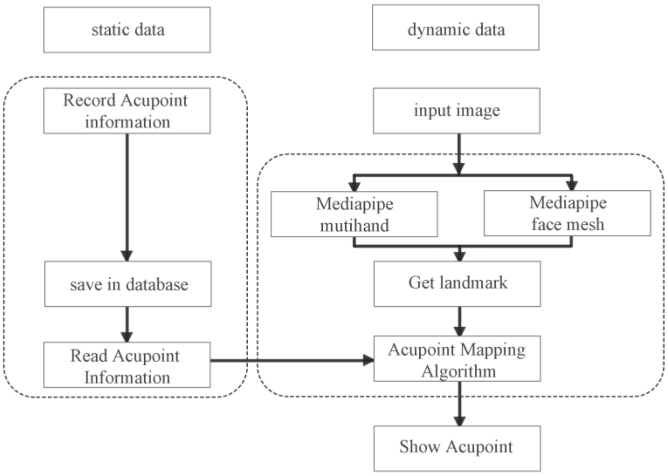

Fig. 4 Product Architecture Diagram.

We employ a custom hand acupoint mapping algorithm that utilizes various mathematical functions such as cross products, dot products (Equation (1)), and generates transformation matrices based on the rotation amount of the base axis (Equation (2)). 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 based on distance. The development of this custom algorithm ensures reliable acupoint mapping.

$$\varphi=\frac{\overrightarrow{v_{0to9}}\cdot\overrightarrow{nv_{0to9}}}{\sqrt{(x_0-x_9)^2+(y_0-y_9)^2}\times\sqrt{(nx_0-nx_9)^2+(ny_0-ny_9)^2}} \quad (1)$$

$$where \quad \overrightarrow{v_{0to9}}=(x_0-x_9, y_0-y_9),$$
$$\overrightarrow{nv_{0to9}}=(nx_0-nx_9, ny_0-ny_9)$$

$$[\alpha, \beta]=[x,\ y]\cdot\begin{bmatrix}\cos\varphi & -\sin\varphi\\ \sin\varphi & \cos\varphi\end{bmatrix}\times z \quad (2)$$

D. *DNN Hand Gesture Recognition*

HaGRID (Hand Gesture Recognition Image Dataset) [12] is a dataset for hand gesture recognition images. In this project, we utilize the data from HaGRID as well as custom hand data to train a DNN (Deep Neural Network) model. The dataset consists of a total of 28,304 samples. Figure 5 illustrates the DNN network. The input for this model consists of 42 features, including the X and Y coordinates of 21 hand localization points returned by MediaPipe. 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 consists of four classes: menu, choose, cancel, right, left, and no gesture. This hand gesture recognition functionality enables users to interact more conveniently with the smart mirror, enhancing the overall user experience of the product.

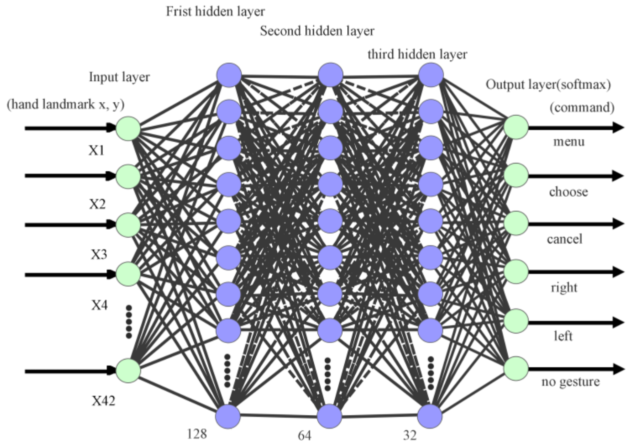

Fig. 5 DNN hand recognition model

E. *State Machine*

Figure 6 illustrates the main operational logic of the state machine in the software code of this project. The purpose of this state machine is to perform a series of operations and decisions based on the landmark data returned by the hand tracking model of MediaPipe. A more detailed explanation of this operational logic is provided below. Firstly, the system retrieves the landmark data as input from the hand tracking model of MediaPipe. Next, these input data are passed to a custom DNN model for processing. The output of the DNN model is transmitted to the main thread (smart mirror - main thread) through event functions. This design helps prevent race conditions and critical sections between different threads, ensuring the proper handling of data. Once the output of the DNN model reaches the main thread, the smart mirror evaluates the output. This evaluation may involve operations based on predefined logic or conditions. Based on the evaluation result, the state will execute the corresponding method. These methods may involve updating the display content of the mirror, adjusting mirror 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 UI interfaces. This allows the screen display to change based on the system's requirements and state variations. The state can also send events to other threads. 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.

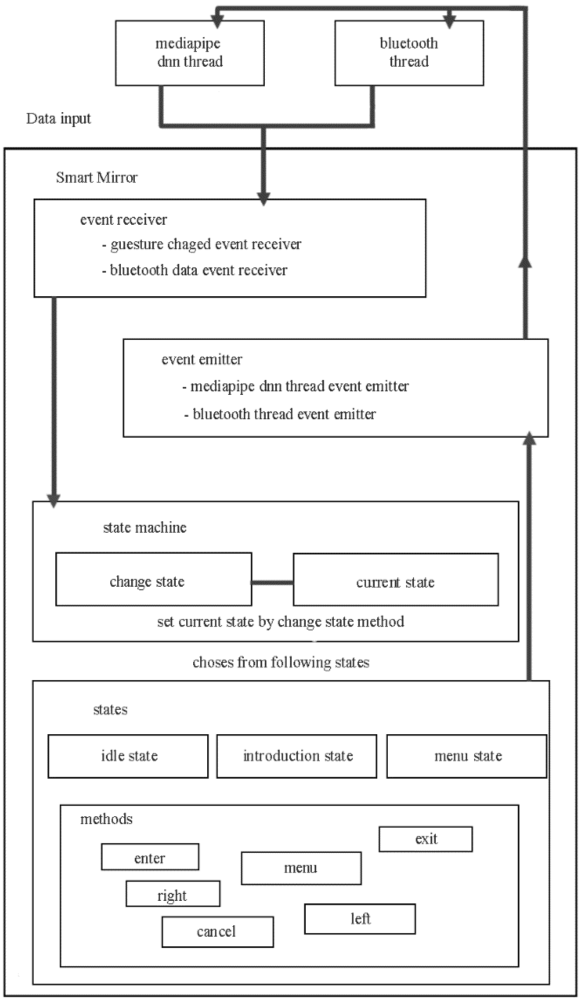

Fig. 6 finite state machine architecture

F. Mobile Bluetooth Pairing

We provide Bluetooth connectivity between the mobile device and the software side (as shown in Figure 7). 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 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 acupoint information. This design not only provides a more intuitive and interactive interface but also enhances usability by increasing convenience.

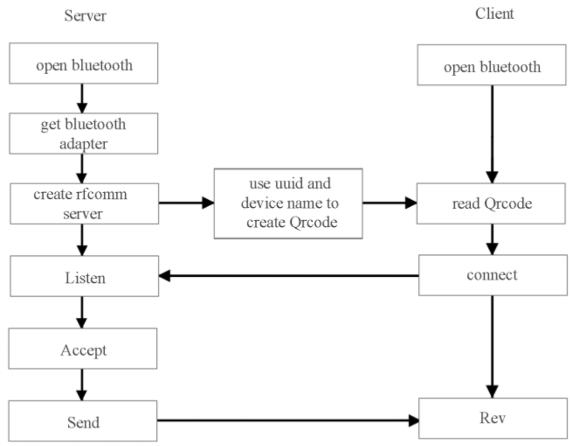

Fig. 7 Mobile phone bluetooth connection

G. BERT AI Consultation

BERT AI Figure 8 illustrates the AI consultation architecture of this project. We have set up our own server and built a simple chatbot on the server. We utilize Rasa to create the chatbot, and when the user's intent on the mobile device is to inquire about a disease or symptom, the BERT model trained in this project provides corresponding effective acupoints based on the user's input. The smart mirror then displays only these acupoints, allowing the user to further alleviate their symptoms.

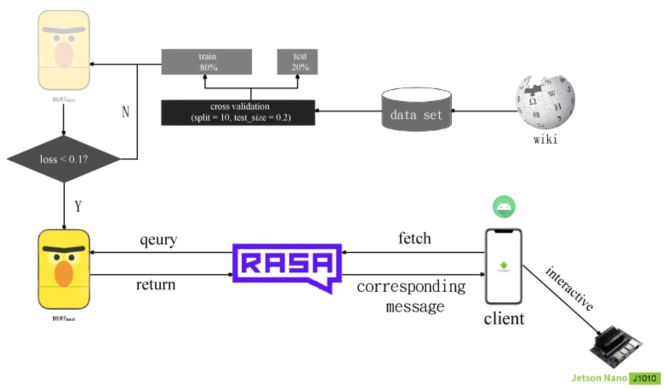

Fig. 8 Architecture Diagram of BERT AI Consultation

In this work, we employ the pre-trained BERT model, specifically the bert-base-chinese variant [13]. This model consists of 12 layers, 768 hidden units, and 12 attention heads, with a parameter size of 110MB. Before training the model, we initialize the token input of the BERT model and incorporate data collected from Wikipedia. Currently, our dataset includes dialogue information associated with 109 disease categories and symptoms related to these diseases, totaling 623 records. We use this data for fine-tuning the model, which involves connecting a new simple classifier to the last layer of BERT to identify the disease category based on the user's input.

IV. RESULT

A. *Acupoint Mapping Effect*

The effectiveness of acupoint mapping can be observed in Figure 9. The hand acupoints adhere to fixed positions and do not shift due to hand rotation or other gestures. This demonstrates the accuracy and effectiveness of our acupoint mapping algorithm.

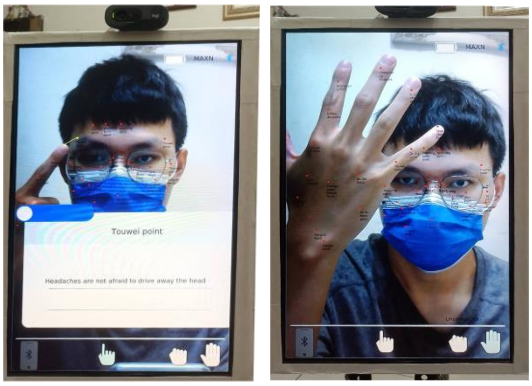

Fig. 9 Acupoint Mapping Effect Display

B. *DNN Performance Evaluation*

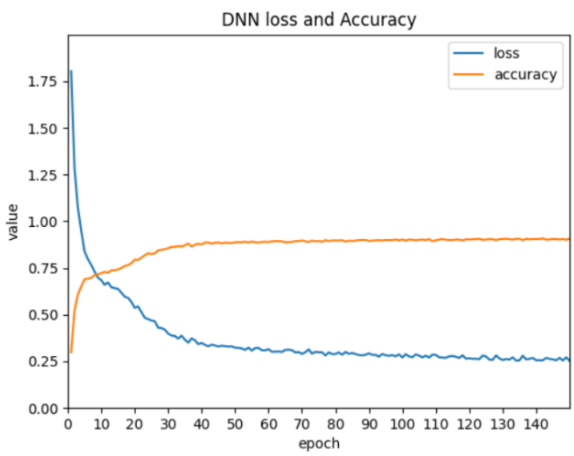

Fig. 10 DNN Hand Gesture Recognition Loss and Accuracy

$$\text{Loss}= -\sum_{i=1}^{output\ size} y_i \cdot \log \hat{y}_i \quad (3)$$

We employed DNN for hand gesture recognition with 150 epochs, a batch size of 16, and the categorical cross-entropy loss function (Equation 3). 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. 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, cancel, choose, 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. Figure 10 intuitively illustrates the progress and effectiveness of our model training. This technology exhibits high accuracy in hand gesture recognition, providing a solid foundation for the successful implementation of this project.

C. *BERT Performance Evaluation*

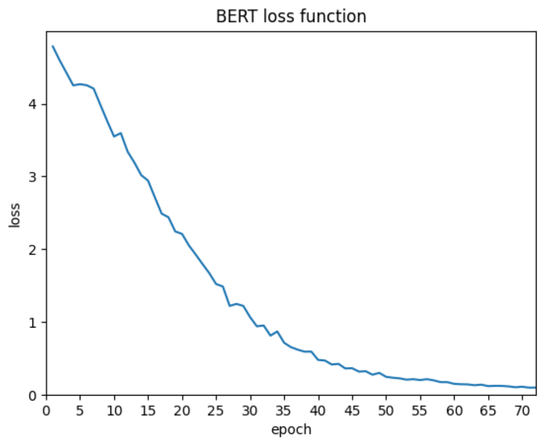

Fig. 11 BERT Loss Fuction

In this paper, we divided the dataset into an 80% training set and a 20% testing set for fine-tuning the BERT model. During the training process, we used 72 epochs and a batch size of 10. By observing the variation of the loss function in Figure 11, we can see a gradual decrease in the loss function, indicating an improvement in the model's learning performance. We terminated the training when the loss function was below 0.1. After each training session, we obtained four model metrics: Accuracy, Precision, Recall, and F1-Score. To ensure the credibility of the experiment, we performed ten-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. Based on our results, 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. The results in Figure 12 further validate the effectiveness and reliability of our trained BERT model in the AI chatbot for medical consultations.。

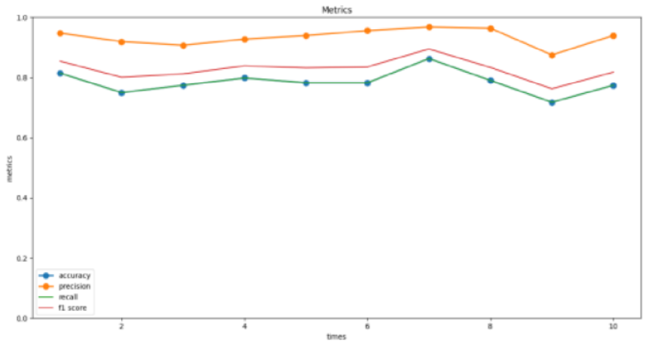

Fig. 12 Architecture Diagram of BERT AI Consultation

V. CONCLUSIONS

In conclusion, this paper has established algorithms for accurate mapping of acupuncture points, demonstrated augmented reality visualization using MEDIAPIPE, achieved hand gesture recognition using a simple neural network based on DNN, and proposed a disease analysis technique based on BERT. We have discussed the accuracy of disease inference in our study and successfully developed an embedded system, integrating the technology into JETSON NANO. This allows users to interactively explore and understand acupuncture knowledge by observing acupuncture points on the human body through our developed product.

ACKNOWLEDGMENT

This work was supported by the Ministry of Science and Technology (MOST), Taiwan, R.O.C., under the Grant MOST 111-2221-E-507-006.

REFERENCES

[1] Gee-Sern Hsu, Hsiao-Chia Peng and Kai-Hsiang Chang, Landmark Based Facial Component Reconstruction for Recognition Across Pose, June 23-28 2014, pp. 34-39.

[2] Menglong Chang, Qing Zhu, Automatic location of facial acupuncture-point based on facial feature points positioning April 1 2017, pp. 545-549.

[3] Yi-Zhang Chen, Corky Maigre , Min-Chun Hu and Kun-chan Lan, Localization of Acupoints using Augmented Reality, June 20 2017, pp. 239-241.

[4] Menghe Zhang, Jürgen P. Schulze, Dong Zhang, FACEATLASAR: ATLAS OF FACIAL ACUPUNCTURE POINTS IN AUGMENTED REALITY, Mar 29 2022.

[5] Baohua Sun, Lin Yang, Wenhan Zhang, Patrick Dong, Charles Young, Jason Dong, Michael Lin, Demonstration of Applications in Computer Vision and NLP on Ultra Power-Efficient CNN Domain Specific Accelerator with 9.3TOPS/Watt, 2019, pp.1213-1223.

[6] Andreas S. Panayides, Amir Amini, Nenad D. Filipovic, Ashish Sharma, Sotirios A. Tsaftaris, Alistair Young, David Foran, Nhan Do, Spyretta Golemati, Tahsin Kurc, Kun Huang, Konstantina S. Nikita, Ben P. Veasey, Michalis Zervakis, Joel H. Saltz, Constantinos S. Pattichis, AI in Medical Imaging Informatics: Current Challenges and Future Directions, 2020, pp.1-15.

[7] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, Matthias Grundmann , MediaPipe: A Framework for Building Perception Pipelines,Jun 14 2019, pp.1-8.

[8] Google(n.d.), MediaPipe Holistic,Retrieved December 2 , 2022, from https://google.github.io/MediaPipe/solutions/holistic.html.

[9] Joshua Willman, Apress Berkeley, Hampton, USA, December 8, 2020.

[10] Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Oct 11 2018 , pp.1~16.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need, Jun 12 2017, pp.1-15.

[12] Alexander Kapitanov, Andrey Makhlyarchuk, Karina Kvanchiani, HaGRID -- HAnd Gesture Recognition Image Dataset, Jun 16 2022 [CrossRef][PubMed].

[13] Hugging Face (n.d.), BERT model pre-trained for Chinese ,Retrieved December 2 , 2022, from https://huggingface.co/bert-base-chinese.