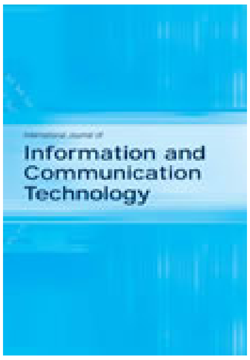


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Implementation of gesture recognition technology optimised by neural networks in OpenMV

Xilong Qu, Siyang Yu, Xiao Tan

Article History:

Received:	08 October 2024
Last revised:	16 December 2024
Accepted:	16 December 2024
Published online:	21 March 2025

Implementation of gesture recognition technology optimised by neural networks in OpenMV

Xilong Qu

College of Information Science and Engineering,
Changsha Normal University,
Changsha, 410000, China
Email: quxilong@126.com

Siyang Yu* and Xiao Tan

School of Information Technology and Management,
Hunan University of Finance and Economics,
Changsha, 410205, China
Email: yusiyang@hufe.edu.cn
Email: xxxhhhl@yeah.net
Email: tanxiao@hufe.edu.cn
*Corresponding author

Abstract: Sign language serves as a vital communication medium for the deaf and hard-of-hearing community; yet, existing gesture recognition systems face challenges such as high costs, limited accuracy in complex environments, and hardware dependence. This study presents a novel gesture recognition system leveraging the OpenMV platform, TensorFlow, and EdgeImpulse to address these issues. The proposed system achieves real-time translation of gestures into text with an accuracy exceeding 98%, demonstrating robustness in varying lighting and background conditions. By integrating machine vision capabilities with cost-effective hardware, this system overcomes the limitations of prior methods, such as the reliance on expensive equipment and poor adaptability to real-world scenarios. These findings highlight the system's potential for widespread application in assistive technologies, offering an affordable and efficient solution for improving communication accessibility.

Keywords: gesture recognition; neural network; OpenMV; feature extraction.

Reference to this paper should be made as follows: Qu, X., Yu, S. and Tan, X. (2025) 'Implementation of gesture recognition technology optimised by neural networks in OpenMV', *Int. J. Information and Communication Technology*, Vol. 26, No. 5, pp.1–21.

Biographical notes: Xilong Qu is a distinguished figure in the field of artificial intelligence and machine learning, with a keen interest in ethical AI development. His expertise encompasses the creation of algorithms that can learn from and adapt to their environment, improving decision-making processes in various sectors such as healthcare, finance, and autonomous vehicles. His work focuses on ensuring that AI systems are developed with fairness, accountability, and transparency, to benefit society as a whole.

Siyang Yu received his PhD degrees in Computer Science and Technology from Hunan University, China, in 2017. He is currently a teacher at Hunan University of Finance and Economics. His research interests include industrial internet of things, abnormal analysis, and intrusion detection.

Xiao Tan is an expert with a profound research background in chip technology, hardware security, and the internet of things. His career is focused on exploring and innovating advanced technological solutions, especially in the fields of microelectronics and information security. His research primarily involves developing more efficient methods for chip design and exploring new types of hardware security strategies, aimed at protecting data from unauthorised access and tampering.

1 Introduction

Gesture recognition, a pivotal area within modern computer science, seeks to create a harmonious environment for human-computer interaction with diverse methodologies. Gesture recognition can not only help deaf-mute people solve interactive problems, but also be applied in our daily life, education, business office and other fields, greatly improving people's quality of life.

With the development of science and technology, we should give more concern to the living conditions of some special groups in society, such as the deaf-mute group. According to the census and the survey report of disabled people, there are now 85 million disabled people, accounting for about 6% of the total population in China, which is a huge number of different types of disabled people in China (Yuan and Cha, 2018). Among them, deaf-mute people account for a large proportion, about 20 million people. According to practical experience, deaf-mute people often encounter communication difficulties in various occasions, such as when they want to buy a train ticket at a railway station, when they handle business in a bank, and when they ask for directions to go to a strange place, etc. On the one hand, the development of sign language recognition technology and research is still lagging behind. On the other hand, there are scarce social training resources of sign language teaching, so the development of sign language courses in colleges and universities across the country still remains very slow. Sign language recognition can solve the above difficulties to a certain extent, and thereby provide convenience for the daily life of the deaf-mute people.

Sign language recognition can be understood as using computer technology to convert sign language signals acquired by sensors or cameras into text signals (Li and Wu, 2022). This technology can not only well improve the communication dilemma between deaf-mute and hearing-impaired people, but also is conducive to promoting the development of human-computer interaction. With the development of computer technology in recent years, gesture recognition has transitioned from the simplest static gesture recognition to the complex continuous gesture recognition. Although a series of research results are constantly emerging, there are still many problems in sign language recognition that have not yet been well solved.

At present, most people communicate with hearing-impaired people mainly by gestures, body language and facial expressions, but less by professional sign language. Through gesture judgment and recognition, this system displays the sign language in the form of text and completes the translation function of sign language, so as to apply it into people's daily life. This has certain value and practical significance for improving the communication dilemma between deaf-mute and hearing-impaired people and popularising sign language. Therefore, sign language text translation will set off a wave of development in the future and promote the development of sign language intelligent recording.

The gesture recognition system designed in this paper is based on the official platform of OPENMV IDE, mainly to solve the communication dilemma between hearing-impaired people and normal people, to provide a sign language translation system from gestures to text, and to lay the foundation for the development of sign language intelligent recording.

Under the current condition of gesture recognition technology, recognition systems can be roughly divided into two categories (Ma, 2001): data glove-based and computer vision-based.

The first category covers data gloves, identification rings, etc. The data gloves were first researched and invented by Grimes (Jiang et al., 2007). There are many sensors on the 'data gloves', through which information is obtained and then transmitted to the computer, and gesture recognition is carried out according to relevant algorithms. This method is featured with the advantage of high recognition rate, by which 3D information of gestures can be obtained directly, but the disadvantage lies in the equipment is expensive, and wearing such hardware equipment (Li et al., 2002) will bring inconvenience to users.

Optical marker method (Zhang, 2022). When in use, the optical marker is worn on the hand, and the information of the state and change of the hand is transmitted to the recognition system through optical technologies such as infrared rays. For example, Davis and Shah (Zhu et al., 2020b) take the gesture with the highlighted visual gloves as the input of the system, by which 7 kinds of gestures can be recognised. This method also has a good recognition effect. Even if the accuracy and stability of gesture recognition are upgraded, the gesture expression is still not natural enough.

The research on gesture recognition started earlier abroad. The sign language recognition system created by Huang (Fu et al., 2022) uses 3D neural network to recognise 15 different gestures. Starner (Zeng et al., 2000) and others applied Hidden Markov Model (HMM) to identify the sentences randomly composed of 40 words with part of speech in American gestures in the input video sequence, in which the HMM parameters were estimated by EM algorithm (Gao et al., 2000). The system tests the recognition of separated words and the recognition of a sentence consisting of five words (the structure of the sentence is defined as pronoun+verb+noun+adjective+pronoun), and the correct recognition rate is as high as 90%. Zhu employs principal component analysis to create a statistical structure, so as to identify the shape of active objects, that is, the identification of modelling systems and elastic objects (Han et al., 2009).

At present, the research focus of gesture recognition is mainly based on vision and depth information. Comparatively speaking, the recognition method based on depth information has better recognition effect and can recognise some complex 3D gesture information more accurately. However, the depth camera based on TOF, structured light and 3D time scanning technology is very expensive, and many researchers are studying the gesture recognition based on Microsoft Kinect depth (Mohandes et al., 2012).

The research on gesture recognition in China started late, but it has also made remarkable achievements in recent years. For example, reference (Yan et al., 2021; Wu et al., 2023a) uses neural network method and Hought transform to recognise 20 kinds of gestures in Chinese sign language.

Duan Hongwei of Shanghai University used LS-SVM algorithm (Wang et al., 2018) to recognise static gestures, and applied HMM model to recognise dynamic gestures.

In a word, the present studies on gesture recognition can be divided into two types; i.e. gesture recognition based on data glove and gesture recognition based on computer vision (Yang, 2013). For data-based glove gesture recognition system, since the size of each hand is not the same, the recognition accuracy of gloves is greatly affected. In the vision-based gesture recognition system, machine vision recognition depends on human hands, but the recognition accuracy is also ideal enough when there are human hand skin chromaticity characteristics in the background, such as the inevitable face and arms (Huang, 2020). Based on a series of problems of inaccurate recognition under technical conditions, this paper proposes a gesture recognition system based on OpenMV.

Combined with the background and significance of this research topic, the gesture recognition system is based on the programmable hardware camera OpenMV. This system is mainly designed to make the life and work of deaf-mute people more convenient and relaxed. Therefore, this system provides the translation function for the text output of gesture recognition.

The main work of this paper is as follows:

1 In the project initiation stage of the system, on the one hand, we make a detailed investigation on the current gesture recognition technology; on the other hand, we fully grasp the characteristics of the actual communication of deaf-mute or hearing-impaired people and master the existing problems in detail (Wang, 2020). Then, in view of these problems and current situation, combined with the research results of domestic related systems and the gesture recognition technology that has been applied in the current practical work, we established the goal of developing a new gesture recognition system.

2 In the system analysis stage, firstly, through the investigation and research on the technology of machine vision at this stage, according to the characteristics of different technologies, the technology to be selected for this system is finally determined. It provides theoretical support for the follow-up design, development and deployment of gesture recognition system. After data retrieve, summary and analysis, the user's demand for gesture recognition system is determined, and then the user's demand is analysed and sorted out to determine the system positioning of the system to be developed, thus laying a good foundation for the succeeding system development.

3 In the system design stage, the system is designed from top to bottom according to the demand analysis. Firstly, it is determined that the gesture recognition system adopts the architecture of programmable camera OPENMV, and is programmed in Python language. Then, according to the specific functional requirements, the three functional modules of gesture recognition, translation and human-computer interaction are established and identified, and then each functional module is analysed and designed in detail to determine the hierarchical structure of each functional module.

4 System implementation and testing stage. At this stage, we need to develop the gesture recognition system through the actual algorithm and Python code. Then, through the test case, the effect is tested to verify whether the system functions

determined in the design stage have been achieved, and if not, the algorithm shall be improved in a timely manner to meet the requirements of accurate and efficient gesture recognition (Wu et al., 2023b). Finally, it is required to point out the shortcomings of the current design, define the direction for further improvement of the system in the future, and put forward expectations for the future prospects of the system.

This paper introduces the design and implementation process of OpenMV-based gesture recognition system through five chapters (Aarthi et al., 2024). The organisation structure of the content is as follows:

- Chapter I is the introduction. This chapter is on the background and significance of this research topic, including the research status of gesture recognition at home and abroad, as well as the main work direction and content of this research project. It puts forward the design, development and implementation of a gesture recognition system based on OpenMV. Finally, the organisational structure of this paper is introduced hereunder.
- Chapter II is on the system hardware design. In this chapter, it is determined to apply a programmable camera OpenMV as the core, and the hardware structure design of the system is completed by analysing the key technical requirements of machine vision.
- Chapter III is on the system software design. This chapter completes the system software design with two parts (Ni et al., 2020). The first part is the program design of OpenMV IDE, which completes the work of parameter setting, basic code and data acquisition. The second part is the program design of Python, neural network learning.
- Chapter IV is on the overall functional test. The main work of this chapter is to test the preset functions of this system, analyse the test process and the functional effect achieved by the test, and realise the text translation function of gesture recognition (Amma and Dhanaseelan, 2021). The test results show that the OpenMV-based gesture recognition system can meet the functional requirements of the system design.
- Chapter IV is on the summary and prospect. The main content of this chapter is to summarise all the work done in this paper, and list the main achievements and innovations of this paper, the shortcomings in the work and the prospect of the next step work.

2 Related research on gesture recognition

Gesture recognition has been explored using various methods, such as data gloves (Gao et al., 2000), depth cameras (Yang, 2013), and machine learning algorithms (Fu et al., 2022). However, these approaches often face issues such as high hardware costs, limited scalability, or difficulty in handling complex backgrounds. This study builds on methods that utilise vision-based recognition, combining low-cost hardware with neural network training (Zeng et al., 2000), while addressing shortcomings in deployment feasibility and adaptability.

2.1 Vision-based gesture recognition

Gesture itself has multiple complex properties, such as differences in time and space and ambiguous expressions, which can be mainly classified into static gestures and dynamic gestures (Zhu et al., 2020a; Jiang et al., 2007). It is classified into four modules i.e. data gesture image acquisition, gesture model training, real-time model matching and human-computer interaction.

Sign language translation is based on gesture recognition. As the language itself is systematic, the algorithm of general gesture recognition cannot be blindly applied. Sign language can be roughly classified into isolated words and dynamic (continuous) words. In sign language gesture recognition, the complex meaning of sign language needs to be comprehensively considered in the analysis and matching stage when referring to general gesture recognition algorithms. For example, the same gesture will be different in isolated words and dynamic words, and the recognition error between similar gestures will lead to ambiguity.

2.2 Model matching

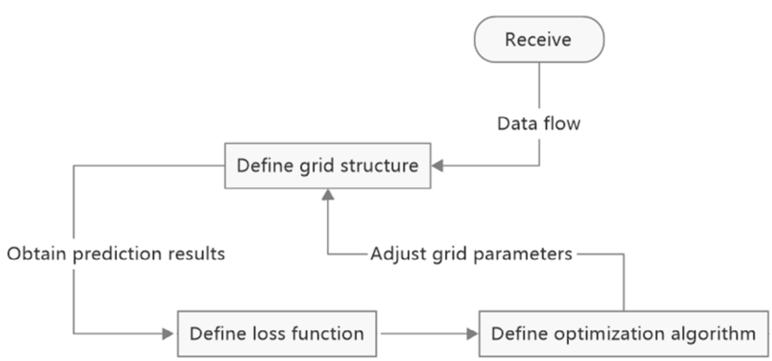
Model is established on basic data so as to better abstract and solve the problems based on dimensions, and the content of model matching mainly includes two key concepts: training and matching.

1 Model training

   The essence is to build an array-like data set internally. X has multiple columns for storing samples, which are sourced from specific data collected; Y is a column that is a kind of label for data set output.

   Training is inseparable from matching, and model training is based on data frame, which serves as the basic level of training and matching. TensorFlow framework was originally developed by Google Intelligent Team named 'Google Brain'. As a symbolic system based on data stream, TensorFlow has been used in various machine learning. Its source code was opened to the outside world in 2015. Since then, the continuous development has not only significantly improved its performance, but also made its architecture more flexible and its portability greatly enhanced.

Figure 1 Schematic diagram of frame training



The calculation result of TensorFlow is a vector structure, which holds the attributes of quantity, such as name, type and dimension, rather than a specific numerical value in mathematics. All calculations are finally transformed into nodes on the 'calculation diagram'. The flow diagram of frame training in Figure 1.

2 Model matching.

There are a variety of model matching techniques for gesture recognition, such as traditional shape matching algorithm and key point matching algorithm, and the matching based on machine learning is developing rapidly at present.

The essence of model matching is that after the real-time collected gestures are accurately obtained, one is to compare them with the samples of the data set one by one and output feedback in sequence; and the other is to match the gestures with the vectors generated by deep learning in turn after scanning calculation. Comparatively speaking, the latter overcomes the problems of the former, such as heavy workload and difficult operation control. Machine learning is an autonomous learning algorithm. For example, in a certain system, the quantities A and B are known, and Y is obtained after calculation. That is the result of machine learning, not artificially prescribed or predicted.

3 Design of system framework

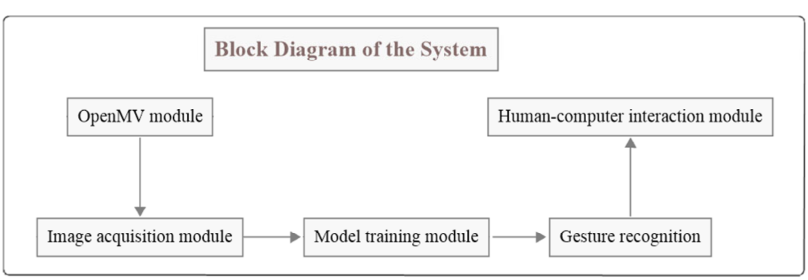
3.1 General block diagram of the system

This system employs the OpenMV IDE platform, and connects with the computer based on the OpenMV visual module to realise the gesture recognition system of sign language translation.

Firstly, the construction of software platform and the connection between hardware devices. Images are transmitted from OpenMV camera to the computer through USB cable, and clear and stable images can be acquired on the software platform. Secondly, the research on the algorithm of gesture recognition. At the same time, a large number of gesture samples from different angles were collected to form a gesture training data set. The data set is used for model training, with the recognition and detection accuracy of the training model taken into consideration.

The overall block diagram of the gesture recognition system is as shown in Figure 2.

Figure 2 Overall block diagram of the system

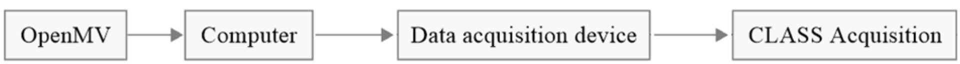


3.2 Image acquisition module

The image acquisition module is based on OpenMV IDE development software, and its own data collector is convenient and fast. The image acquisition module consists of data collector and computer, including data acquisition, analysis and template matching, etc.

Gesture model is very important for gesture recognition system, but the first step of model building is image acquisition. First of all, the number of samples of gesture images should be more than 100, and each gesture should be accurately classified. These two characteristics are realised by the data acquisition editor that comes with OpenMV IDE, which is very convenient and powerful. The image acquisition must follow certain rules. In this system, it is classified and named according to the order of ‘home’, ‘you’, ‘good’ and ‘thank you’. Here, I use Chinese Pinyin as the naming rules, namely ‘jia’, ‘ni’, ‘hao’ and ‘xie’. Among them, ‘none’ stands for the background image. Finally, the Data folder is established for model training use (see Figure 3).

Figure 3 Schematic diagram of image acquisition process



Secondly, the interference of background is really a major pain point of image processing, which leads to the decrease of recognition accuracy. Therefore, in order to solve this problem, this design brings the background into the model training group. Meanwhile, it is found that the more complex the background, the more colour blocks with similar skin colour, and the lower the success rate of model training, which will lead to the lower recognition accuracy of the system.

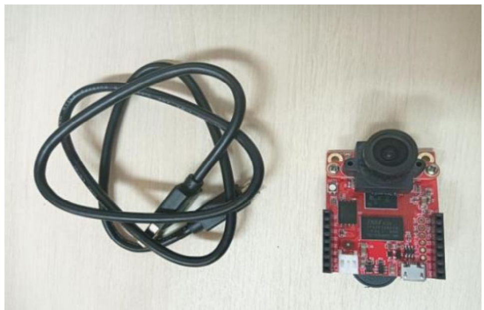
In view of this, there are two options: single colour gamut background control method and skin colour block ratio reduction method. Although the former achieves high recognition accuracy by controlling a single background colour, it is not suitable for practical life scenes. The latter can achieve a high accuracy by lowering the skin colour picture, even if training samples are added in a complex background with a non-uniform colour gamut, so the latter can solve the problem of background interference well. The background model training group is as shown in Figure 4.

Figure 4 Background image (see online version for colours)

3.3 Data analysis module

The system function of this design is to realise gesture recognition, including sign language recognition of isolated words and recognition of dynamic sign language. The hardware includes a set of computer (which can build IDE environment), a programmable camera and a data cable, the physical object is as shown in Figure 5.

Figure 5 OpenMV module and data line (see online version for colours)



OpenMV vision module is the core module of this design. OpenMV is a powerful machine vision module, which is known as ‘programmable camera’. Taking STM32F427CPU as the core, integrated with OV7725 camera core, it has powerful computing power. On the compact hardware module, the core machine vision algorithm is efficiently realised by programming language, which provides possible objective conditions for real-time translation. At the same time, it provides Python programming interface, which provides high compatibility and convenience for software programming.

With regards to OpemMV, its development environment is not limited to ‘camera’. This system design needs to install and build OpenMV IDE in the computer as the development platform (just download it from official website).

Based on the above merits and the functional characteristics of this design, OpenMV is selected as the core module.

3.4 Model training module

The model training module is the core module of the gesture recognition system in this paper. It is based on EdgeImpulse platform and TensorFlow framework.

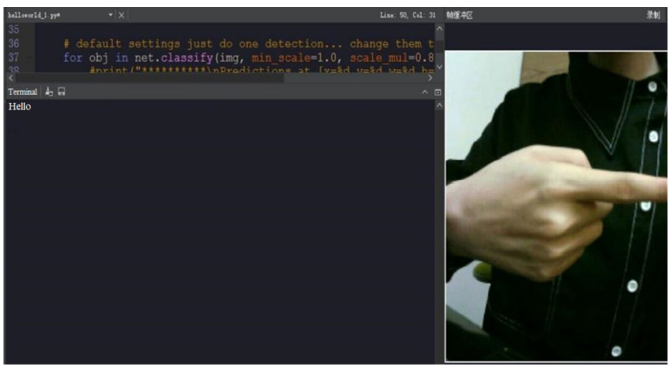
EdgeImpulse is a cloud-based AI platform that can be registered by email and used free of charge. Its cooperation with OpenMV vision module makes the whole process operation of intelligent classification and identification can be realised online, which is very convenient and fast, including: data acquisition, marking, CNN model training and optimisation deployment, etc. This paper chooses OpenMV as the platform of its deployment.

Model training adopts TensorFlow framework, which is one of the products of deep learning. Deep learning is to develop additional components similar to human computers and solve real-world problems through its special brain-like architecture (artificial neural network). Simply speaking, it is an open source framework for machine learning, which can be used to quickly build a neural network to train, evaluate and save the network conveniently and quickly.

3.5 Human-computer interaction module

With the emergence of pattern recognition technologies such as face recognition, voice recognition and eye tracking, human-computer interaction has a wider range of interpretations. As a result, it is also very important to design a more natural and smooth human-computer interaction system by combining various recognition technologies. Human-computer interaction is the information exchange between gestures and programmable cameras, in short, the translation between gestures and texts. Human-computer interaction is to realise the communication between human and OpenMV module, i.e., to realise the intercommunication between programming language and human behaviour or language through a certain interface.

Figure 6 Interactive interface (see online version for colours)



The implementation of human-computer interaction is the construction of software platform and the connection between hardware devices. Images are transmitted from OpenMV camera to computer through USB cable, and clear and stable images are obtained on the software platform. OpenMV camera acquires and transmits gesture images to the computer, and each frame of gesture is thereby recognised in real time, matched with the corresponding Chinese expression, and displayed to the software interface in the form of text in real time. For example, when a person gestures 'hello' in front of the camera, the text 'hello' will appear in the software interface within 3s. An example is as shown in Figure 6.

The human-computer interaction interface selected in this design is OpenMV IDE platform, the matching software of OpenMV, which has unique advantages. One is that the interface is simple and meets the functional requirements of text output, and the other is that it is highly compatible with OpenMV module.

4 Design and implementation of algorithm

4.1 Gesture data acquisition

The pre-preparation of image processing is to accomplish classified acquisition of the gesture images to be recognised: establish a DATA folder, and set up a parallel CLASS classification folder hereunder to provide sufficient picture samples for deep learning.

Image analysis and feature matching are compared and identified by generating models. Image analysis and feature matching are identified by comparison of generated models. The images that need to be processed are 3D-2D gestures obtained by OpenMV Cam.

Figure 7 Picture data of the gesture 'good' (see online version for colours)

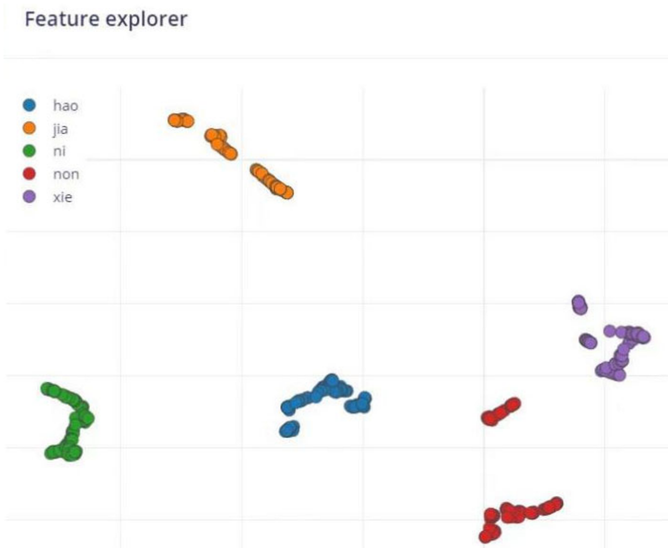
Therefore, the data collection of OpenMV IDE is mainly composed of collection and classification. Each Class is stored in its own data set editor, and more than 100 hand gestures are identified and saved for neural network training. The collected images include gestures such as 'home', 'hello', 'thank you', and some background images, which are saved in the dataset. If inappropriate pictures are found during collection, they should be deleted in time.

The dataset included over 500 images for each gesture class ('home,' 'thank you,' etc.), ensuring sufficient training data. Images were captured from diverse angles, lighting conditions, and background settings to improve model robustness. Backgrounds ranged from plain walls to cluttered scenes, simulating real-world environments. To augment the dataset, preprocessing steps such as rotation, scaling, and flipping were applied, enhancing diversity and generalisability. All images were resized to 96x96 pixels and converted to greyscale to standardise input dimensions and reduce computational load, as shown in Figure 7.

4.2 Feature extraction

The essence of feature extraction is a description method. After a large number of gesture images are obtained by the data collector of OpenMV IDE, they are uploaded to the Edge Impulse cloud platform. Uploaded gesture images will be scanned in turn to get data sets, which will be arranged, analysed and classified in turn. Feature extraction is accomplished in this way. In deep learning, feature extraction incorporates the tasks of scanning, recognition and classification. The combination of the two makes feature extraction not only ensure the accuracy but also greatly improve the effect.

Figure 8 Feature layout (see online version for colours)



Feature extraction is based on TensorFlow framework, with the final feature layout clear and distinct, without cross set. The feature layout is as shown in Figure 8.

4.3 Model training

In the construction of the algorithm model of this design, firstly, a data set is constructed by using the OpenMV IDE, which is uploaded to the Edge Impulse official website. Then, the TensorFlow deep learning algorithm is used for neural network training for modelling the features of gestures. Finally, a TensorFlow Lite convolutional neural network (CNN) is generated, which will run on the OpenMV Cam. The trained model can complete matching efficiently and quickly.

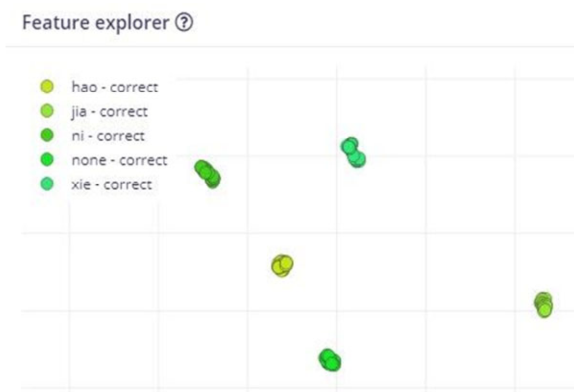
Using EdgeImpulse to train the neural network model suitable for OpenMV is implemented in four steps: image data acquisition, data uploading, model training and deployment.

Firstly, image data collection. After the OpenMV is normally connected to the computer, a sufficient number of images are taken through the OpenMV IED, and the detailed process has been stated in Section 4.1. Secondly, data uploading. Register and log in to EdgeImpuls official website (https://edgeimpulse.com/), create a new ‘project’, select ‘image’, and then import the class data in turn. Among them, 80% of the uploaded data sets are training sets and 20% are testing sets. Regarding the configuration processing module, it is required to set the image size as 96*96, set ‘learning mode selection’ to save ‘save impulse’, and the interface will appear ‘successfully’ indicating successful configuration. Set the image colour format to GRE for image preprocessing, and click ‘generate features’ to start the training, see Figures 9 and 10.

Finally, through ‘transfer learning’, it is required to set parameters, learning rate, number of cycles, and minimum confidence rate, and start the model training lasting for about 4 minutes.

Figure 9 Visualisation (see online version for colours)

Figure 10 Feature layout of training 'class' (see online version for colours)



Model testing is implemented to get the accuracy as shown in Figure 11.

Figure 11 Model testing results (see online version for colours)

Model testing results

ACCURACY
100.00%

	HAO	JIA	NI	NONE	XIE	UNCERT.
HAO	100%	0%	0%	0%	0%	0%
JIA	0%	100%	0%	0%	0%	0%
NI	0%	0%	100%	0%	0%	0%
NONE	0%	0%	0%	100%	0%	0%
XIE	0%	0%	0%	0%	100%	0%
F1 SCORE	1.00	1.00	1.00	1.00	1.00	

The training code is as shown in Figure 12.

Figure 12 Establishing neural network model code (see online version for colours)

Neural network architecture

```
sys.path.append('./resources/libraries')
import ei_tensorflow.training

WEIGHTS_PATH = './transfer-learning-weights/keras
    /mobilenet_v2_weights_tf_dim_ordering_tf_kernels_0.35_96.h5'

# Download the model weights
root_url = 'https://cdn.edgeimpulse.com/'
p = Path(WEIGHTS_PATH)
if not p.exists():
    print(f"Pretrained weights {WEIGHTS_PATH} unavailable; downloading...")
    if not p.parent.exists():
        p.parent.mkdir(parents=True)
    weights_data = requests.get(root_url + WEIGHTS_PATH[2:]).content
    with open(WEIGHTS_PATH, 'wb') as f:
        f.write(weights_data)
    print(f"Pretrained weights {WEIGHTS_PATH} unavailable; downloading OK")
    print("")

INPUT_SHAPE = (96, 96, 3)


base_model = tf.keras.applications.MobileNetV2(
    input_shape = INPUT_SHAPE, alpha=0.35,
    weights = WEIGHTS_PATH
)

base_model.trainable = False

model = Sequential()
model.add(InputLayer(input_shape=INPUT_SHAPE, name='x_input'))
# Don't include the base model's top layers
last_layer_index = -3
model.add(Model(inputs=base_model.inputs, outputs=base_model
    .layers[last_layer_index].output))
model.add(Reshape((-1, model.layers[-1].output.shape[3])))
model.add(Dense(16, activation='relu'))
model.add(Dropout(0.1))
model.add(Flatten())
```

4.4 Gesture recognition

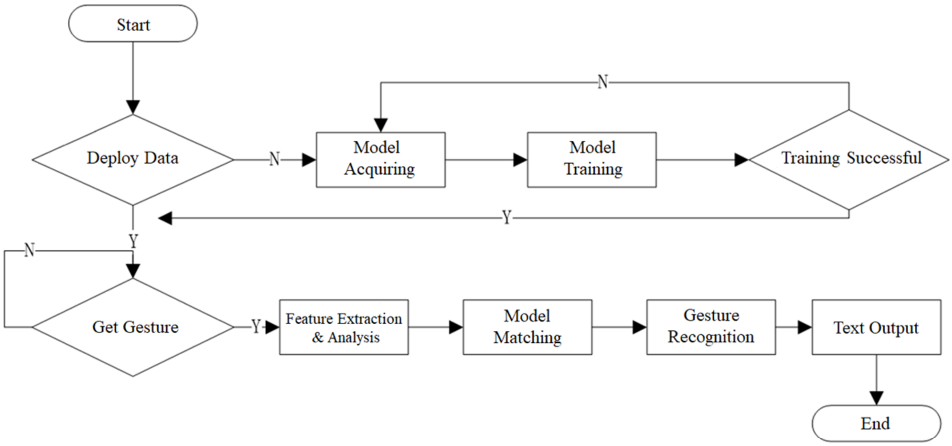
The algorithm of gesture recognition acts as the core of this system, and its main gesture recognition processes include gesture acquisition, model matching and text output.

First, gesture acquisition. Connect the computer with OpenMV Cam through USB cable, and real-time gesture images can be acquired on the OpenMV IDE platform as long as the connection and parameter settings meet normal conditions. Second, model matching. Model matching is accomplished by importing the database trained by neural network into OpenMV IDE platform and running the matching code. Third, text output. Text output is to realise human-computer interaction, and the result of model matching is used as the feedback source. When the matching similarity rate of feedback is greater than 80%, the corresponding text is output to the interactive interface.

The previous data set includes 'home', 'hello' and 'thanks'. When the gesture is displayed in the real-time image acquired by the camera, the camera calls the internal neural network library to recognise the gesture and generates the trust degree of each Class. Therefore, we feed back the gesture with the confidence in the range of 80%–99% to the Flag, thus meeting the requirements of gesture recognition.

The flow chart of gesture recognition algorithm is shown in Figure 13.

Figure 13 Flow chart of gesture recognition algorithm



4.5 Algorithm description

The neural network implemented is a CNN comprising three convolutional layers, each followed by ReLU activation and max-pooling layers. The convolutional layers extract spatial features from gesture images, while the max-pooling layers reduce spatial dimensions, minimising computational complexity. The final fully connected layer outputs gesture classifications based on the trained dataset. The network was trained using a learning rate of 0.001, a batch size of 32, and 100 epochs. To prevent overfitting, dropout layers with a rate of 0.2 were incorporated. The Adam optimiser was employed for parameter updates, with categorical cross-entropy used as the loss function.

To achieve efficient and accurate real-time gesture recognition, the system follows a structured process that integrates image acquisition, pre-processing, feature extraction, and neural network-based classification. The algorithm is designed to ensure that gestures are recognised with a confidence threshold of at least 80%, enabling reliable text output. Below, we provide the pseudocode that outlines the key steps of the gesture recognition process implemented on the OpenMV platform. This algorithm demonstrates how the various system components, including hardware and software, work together to achieve high-accuracy gesture recognition.

 # Code: gesture recognition process flow

```
# Step 1: Initialise system and load trained CNN model
initialise_system()
load_model("trained_model.tflite")

# Step 2: Start real-time image acquisition
while True:
    img = capture_image(OpenMV_camera)
```

```
    # Step 3: Pre-process image
    preprocessed_img = preprocess(img, resolution=(96, 96))

    # Step 4: Perform gesture recognition
    predictions = model.predict(preprocessed_img)
    max_confidence, recognised_class = get_highest_confidence(predictions)

    # Step 5: Check confidence threshold
    if max_confidence >= 0.80:
        # Output recognised gesture as text
        display_output(recognised_class)
    else:
        # Log unrecognised gesture for further analysis
        log_unrecognised_gesture(img)

    # Break condition for demo or testing
    if stop_condition_met():
  break
```

5 System testing

5.1 System testing method

The actual example method is used as the testing method used in this system. By comparing the testing phenomena with the preset functions, statistical analysis is made in turn according to the testing results of corresponding functions, the recognition accuracy rate is calculated, and the corresponding statistical table is plotted.

5.2 System testing process

The testing process of this system focuses on the realisation of functions. That is, through the recognition effect, the clarity, recognition accuracy and recognition efficiency of dynamic pictures are comprehensively taken into consideration.

The first is gesture recognition of isolated words: real-time images are acquired by OpenMV Cam, and clearer images are obtained by adjusting parameters, and the Chinese meaning of the gesture language is displayed in the software interface within 3s. As shown in Figure 14, The Testing of Gesture ‘home’.

Translation function of isolated words in gesture language: in this system, the translation of isolated word ‘home’ in gesture language follows the processes below: OpenMV camera acquires the gesture and transmits the gesture images to the computer, and outputs the feedback through the model matching algorithm of the gesture, and then the meaning expressed by the gesture is displayed on the software interface through text language. For example, when a person makes gestures ‘home’ in front of the camera, the character ‘home’ will appear on the software interface within 3s. The gesture language ‘home’ is as shown in Figure 15.

Figure 14 The gesture 'home' (see online version for colours)

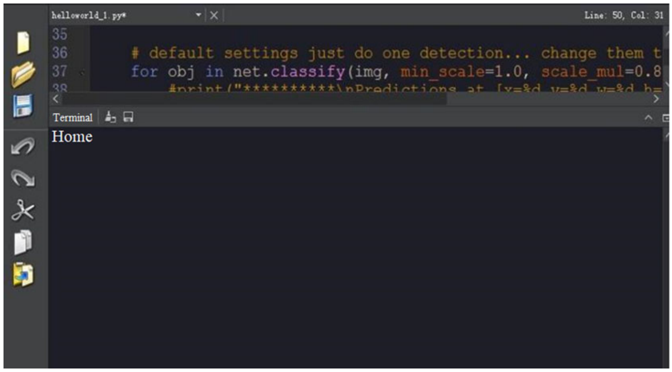


Figure 15 Gesture language 'home' (see online version for colours)

In this design, dynamic gestures are polite expressions in gesture language. The recognition of dynamic gestures is the same as that of isolated words in gesture language, i.e. OpenMV camera acquiring the gesture and transmitting the gesture images to the computer, but it is more complicated than isolated words, and it is more difficult to grasp the accuracy of recognition. Therefore, each frame of gestures should be subject to background recognition and matched with specific sign language polite expression model, and displayed to the software interface in real time in the form of text.

After confirming the normal connection between the computer and the OpenMV, when a person makes gestures 'hello' and 'thanks' in front of the camera lens of the OpenMV, and the text 'hello' and 'thanks' will appear in the software interface of the OpenMV IDE within 3s. The gestures 'hello' and 'thanks' are as shown in Figure 16.

Figure 16 Gesture language 'hello' and 'thanks'

Dynamic courtesy gesture language recognition is as shown in Figure 17 example test of the gesture 'hello'.

Figure 17 Gesture 'hello' (see online version for colours)

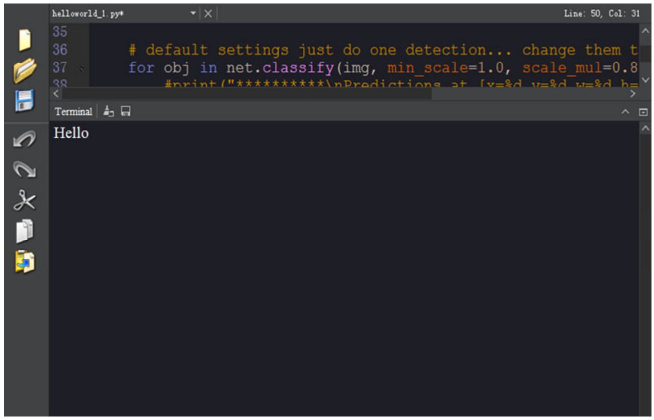
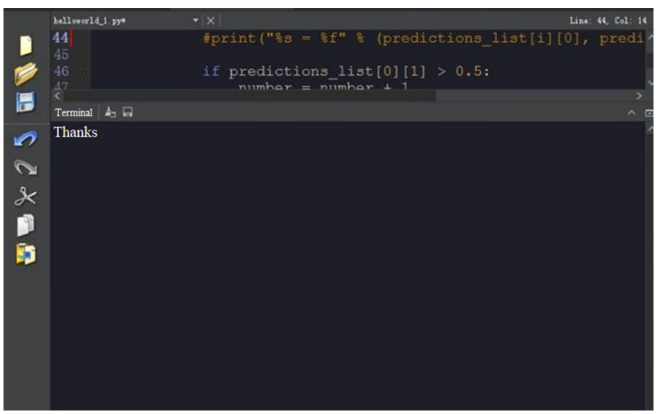


Figure 18 Gesture 'thanks' (see online version for colours)



As shown in Figure 18 is example test of the gesture of 'thanks'.

5.3 *Analysis of test results*

The testing results indicate that the system meets the expected target functions, realises gesture recognition of isolated words and dynamic words with rather high recognition accuracy, see Table 1.

Table 1 Comparison of gesture recognition systems

Gesture	*Test quantity (times)*	*Accurate recognition quantity (times)*	*Accuracy (%)*
Home	50	50	100
You	50	49	100
good	50	50	100
Thanks	50	50	98

While the system achieved high accuracy (>98%) in controlled environments, its performance dropped to ~85% under complex conditions, such as cluttered backgrounds with colours resembling skin tones. Similarly, extreme gesture angles and poor lighting reduced recognition accuracy. These challenges highlight the need for further model optimisation and additional data collection to improve robustness. Addressing these issues will enhance the system's ability to handle real-world variability.

Table 2 Comparative analysis

Method	*Accuracy (%)*	*Strengths*	*Weaknesses*
OpenMV-based gesture recognition	98.4	High accuracy, real-time translation, affordable hardware	Small dataset, limited performance in complex backgrounds
3D neural network	97.1	Effective for continuous gestures	High computational cost, hardware dependent
Hidden Markov Model (HMM)	93.8	Sentence-level recognition possible	Parameter tuning complexity
HMM/ANN combination	96.5	Robust for isolated gestures	Limited to predefined vocabularies
VPL data glove	99.2	Extremely high precision	Expensive equipment, inconvenient for users
Stereo vision algorithm	98.5	Recognises 3D gestures	High hardware cost, issues with dynamic gestures

5.4 Comparative analysis

To better contextualise the performance of our gesture recognition system, we conducted a comparison of our results with previous studies in the field of gesture recognition. Table 1 highlights the accuracy rates, strengths, and weaknesses of various methods, including those based on 3D neural networks, HMMs, and vision-based algorithms. Our study demonstrates superior accuracy (exceeded 98%) and practicality with the use of the OpenMV platform, outperforming several prior methods in terms of real-time capability and hardware affordability. However, it is important to note that our system faces challenges in handling complex backgrounds, a limitation shared with some other vision-based approaches.

6 Conclusions

Gesture recognition represents a transformative avenue for bridging communication gaps in the deaf and hard-of-hearing community. Here, we present a gesture recognition system leveraging the OpenMV platform, TensorFlow, and EdgeImpulse to achieve real-time, high-accuracy (98–100%) translation of gestures into text. This cost-effective and portable solution overcomes limitations inherent to previous approaches, including reliance on expensive hardware and inadequate adaptability to real-world variability.

While our results demonstrate the system's robustness under controlled conditions, challenges remain in recognising gestures in cluttered backgrounds or extreme environmental conditions. Addressing these limitations will require further advancements in model architecture, enhanced dataset diversity, and adaptive algorithms.

Comparative benchmarks reveal that this system achieves comparable or superior accuracy and speed relative to existing solutions, all while dramatically reducing costs. These findings position the system as a scalable tool for assistive technologies, with the potential to empower millions of individuals through affordable and accessible communication solutions.

Future efforts will focus on scaling the system for broader gesture vocabularies, improving robustness in dynamic environments, and exploring seamless integration with wearable and mobile platforms. Such developments will not only expand the system's applicability but also bring us closer to truly inclusive communication technologies.

Acknowledgements

The works that are described in this paper are supported by the scientific research project of Hunan Provincial Education Department (No. 24A0791 and No.24B0926 and No.24A0786) and Provincial Natural Science Foundation of Hunan (2023JJ30101).

References

Aarthi, S., Satheesh, T., Sakthivel, R. et al. (2024) 'Disturbance estimation-based resilient control design for interval type-2 fuzzy stochastic systems under multiple disturbance observer', *International Journal of General Systems*, Vol. 53, No. 6, pp.678–706, https://doi.org/10.1080/03081079.2024.2302613.

Amma, N.G.N. and Dhanaseelan, F.R. (2021) *Optimal Privacy Preserving Scheme Based on Modified ANN and PSO in Cloud*, pp.773–793, Research Anthology on Privatizing and Securing Data, IGI Global, US, Commonwealth of Pennsylvania

Fu, S., Cha, X. and Xu, C. (2022) 'Intelligent drug delivery trolley based on OpenMV vision system', *Industrial Control Computer*, Vol. 5, No. 2, pp.6–9.

Gao, W., Ma, J., Wu, J. et al. (2000) 'Sign language recognition based on HMM/ANN/DP', *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 14, No. 5, pp.587–602.

Han, J., Awad, G. and Sutherland, A. (2009) 'Automatic skin segmentation and tracking in sign language recognition', *IET Computer Vision*, Vol. 3, No. 1, pp.33–35.

Huang, S. (2020) *Research on Chinese Sign Language Recognition Method for Large Vocabulary*, pp.21–23, University of Science and Technology of China, US.

Jiang, F., Gao, W., Wang, C. et al. (2007) 'R Development in signer-Independent sign language recognition and the ideas of solving some key problems', *Journal of Software*, Vol. 18, No. 3, pp.477–489.

Li, Q., Fang, Z., Shen, M. et al. (2002) 'Gesture recognition technology and its application in human-computer interaction', *Ergonomics*, No. 1, pp.27–29+33.

Li, S. and Wu, Y. (2022) 'Research on the design of barrier-free communication assistance system for deaf-mute people', *Internet of Things Technology*, Vol. 12, No. 11, pp.113–116.

Ma, J. (2001) *Research on Statistical Model of Sign Language Understanding*, Graduate School of China Academy of Sciences, Institute of Computing Technology, US.

Mohandes, M., Deriche, M., Johar, U. and Ilyas, S. (2012) 'A signer-independent Arabic sign language recognition system using face detection, geometric features, and a hidden Markov model', *Computers & Electrical Engineering*, Vol. 38, No. 2, pp.422–433.

Ni, P., Li, Y. and Chang, V. (2020) 'Research on text classification based on automatically extracted keywords', *International Journal of Enterprise Information Systems (IJEIS)*, Vol. 16, No. 4, pp.1–16.

Wang, F. (2020) 'Design and implementation of gesture recognition based on embedded system', *Radio Communications Technology*, Vol. 46, No. 6, pp.717–721.

Wang, J., Zhu, E., Huang, S. et al. (2018) 'Dynamic gesture recognition method based on deep learning', *Computer Simulation*, Vol. 9, No. 3, pp.366–370.

Wang, S., Yang, J., Wang, Z. et al. (2022) 'Research on gesture recognition based on the detection of key points on hands', *Computer and Telecommunications*, No. 5, pp.29–33+47.

Wu, C.H., Lam, H.Y., Kong, A. et al. (2023a) 'The design and evaluation of a digital learning-based English chatbot as an online self-learning method', *International Journal of Engineering Business Management*, Vol. 15, DOI: 10.1177/18479790231176372.

Wu, X., Ding, S., Niu, B. et al. (2023b) 'Predefined-time event-triggered adaptive tracking control for strict-feedback nonlinear systems with full-state constraints', *International Journal of General Systems*, Vol. 5, No. 1, pp.1–29.

Yan, Y., Yang, H. and Yu, Z. (2021) 'Intelligent face recognition system based on OpenMV', *Journal of Physics: Conference Series*, Vol. 2137, No. 1, p.012074, DOI: 10.1088/1742-6596/2137/1/012074.

Yang, Q. (2013) *Research on Chinese Sign Language Recognition Technology Based on Apparent Modeling*, pp.82–90, Northwestern University, US.

Yuan, B. and Cha, C. (2018) 'Present situation and prospect of gesture recognition technology', *Science and Technology Innovation*, Vol. 22, No. 32, pp.95–96.

Zeng, F., Liang, B., Liu, Z. et al. (2000) 'Design of human-computer interaction environment based on data gloves', *Journal of Image and Graphics*, Vol. 6, No. 2, pp.67–71.

Zhang, F. (2022) 'Design of non-contact object size and shape measurement system based on OpenMV', *Modern Information Technology*, Vol. 6, No. 9, pp.92–95+99.

Zhu, C., Jin, L.S., Mesiar, R. et al. (2020a) 'Using preference leveled evaluation functions to construct fuzzy measures in decision making and evaluation', *International Journal of General Systems*, Vol. 49, No. 2, pp.161–173.

Zhu, W., Yang, Y. and Liang, H. (2020b) 'Intelligent car identification and chess clamping design based on Arduino', *Digital Technology and Application*, Vol. 38, No. 8, pp.132–134.