applied sciences

MDPI

Article

Research on Gesture Recognition System Using Multiple Sensors Based on Earth's Magnetic Field and 1D Convolution Neural Network

Bo Shi, Xi Chen *, Zhongzheng He, Haoyang Sun and Ruoyu Han

School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China
* Correspondence: chenxi@bit.edu.cn

Abstract: A commonly used method of gesture recognition is the use of sensor technology. Typically, technology detecting Earth's magnetic field is used in indoor positioning, and magnetic detection technology serves as a redundant method for gesture recognition devices. In this paper, we propose a novel system that utilizes multiple sensors measuring Earth's magnetic field to collect data and perform gesture recognition through a one-dimensional convolutional neural network algorithm. By applying the detection of Earth's magnetic field to gesture recognition, our system significantly improves the accuracy of recognition through a one-dimensional (1D) neural network algorithm. We conducted experiments where we collected and recognized American Sign Language standard letters, and achieved an accuracy rate close to 97%. Our experimental results demonstrate that this gesture recognition system using magnetic field sensors and a one-dimensional neural network algorithm is feasible for practical applications. Furthermore, our approach reduces the complexity of the device compared to the gesture recognition method based on artificial magnetic fields, while maintaining high recognition accuracy and not limiting the user's hand movements. This technology holds great promise for the field of human–computer interaction.

Keywords: human–computer interaction; magnetic detection; gesture recognition; deep learning; sign language recognition

check for updates

Citation: Shi, B.; Chen, X.; He, Z.; Sun, H.; Han, R. Research on Gesture Recognition System Using Multiple Sensors Based on Earth's Magnetic Field and 1D Convolution Neural Network. *Appl. Sci.* **2023**, *13*, 5544. https://doi.org/10.3390/app13095544

Academic Editor: Nunzio Cennamo

Received: 14 March 2023
Revised: 2 April 2023
Accepted: 26 April 2023
Published: 29 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Human–Computer Interaction (HCI) technology refers to methods of interacting between humans and computer systems with the primary aim of improving the efficiency and experience of such interactions. This technology encompasses a variety of interaction methods including the keyboard, mouse, touch screen, gesture recognition, voice recognition, virtual reality, and augmented reality. Gesture recognition technology is particularly noteworthy as it allows for human–computer interaction by acquiring and interpreting user gestures. Compared to other HCI technologies, gesture recognition technology boasts several advantages, such as convenience, intuition, and strong interactivity.

Currently, gesture recognition technology can be categorized into two types: computer-vision-based and sensor-based. Computer-vision-based technology captures images of hand movements using monocular cameras, binocular cameras [1], or composite principle vision devices [2]. Then, deep learning [3], Edge Orientation Histogram (EOH) [4], or neural network [1] algorithms extract the hand features from the images and recognize the gestures. Gesture recognition methods based on computer vision have been widely researched [5].

Sensor-based gesture recognition technology obtains gesture-related information through sensors attached to the user's hand and recognizes the gestures by analyzing the data collected by the sensors [6]. Commonly used sensors include myoelectric sensors [7], flexible resistive materials [8,9], optical sensors [10], acceleration sensors, gyroscopes [11], magnetic field sensors [12], etc. Multiple-sensor redundant measurement methods commonly used include computer vision, accelerometer, and flexible sensor composites [13];

accelerometer and magnetic field sensor composites [14]; accelerometer, gyroscope, and acoustic sensor composites [15]; and myoelectricity and accelerometer composites [16]. The data collected by sensors are processed through setting thresholds, machine learning [17], and other algorithms to achieve gesture recognition.

Magnetic field sensors are commonly used as a supplementary feature in gesture recognition devices [18]. Some devices are designed to turn off magnetic field sensors during actual use to conserve power consumption [19]. However, as magnetic field detection technology continues to advance, positioning [20] and gesture recognition technology [21] based entirely on magnetic field sensors have been proposed, offering practical levels of accuracy for gesture recognition. Gesture recognition technology based on magnetic field detection has two main detection systems: artificial magnetic fields and geomagnetic fields. Artificial magnetic field sources include permanent magnets worn on the user's fingers [14] and energized coils that generate specific magnetic fields [22,23]. However, a permanent magnet [14] or electromagnet [24] worn on the user's finger or wrist can complicate the system and impact the user's experience. On the other hand, a magnetic field sensor placed at a fixed location may limit the range of motion of the user's hand [21]. Technology based on detecting Earth's magnetic field is widely used for indoor positioning [20], but it is less commonly used in the field of gesture recognition. Nonetheless, with the development of magnetic field detection technology, it is possible that gesture recognition systems based on the Earth's magnetic field could become prevalent in the future.

A Convolutional Neural Network (CNN) is a deep neural network that utilizes a convolutional structure. This algorithm is widely regarded as one of the core components of deep learning and is commonly applied in diverse fields such as computer vision [25,26] and language processing [27–30]. In a classic CNN, the convolution kernel is two-dimensional, which enables it to perform a sliding window operation on both the width and height of the feature map simultaneously. This technique is commonly employed in computer vision and image processing. In contrast, a one-dimensional CNN functions similarly to a two-dimensional CNN [31] but takes in one-dimensional data and outputs one-dimensional data after convolution and pooling operations. This type of CNN is mainly used in sequence models and natural language processing. The one-dimensional CNN has fewer parameters compared to the two-dimensional CNN, making it less dependent on large-scale datasets. The one-dimensional CNN is utilized for detecting Alzheimer's disease via online handwriting analysis [32], as well as for EEG-based motor imagery classification [33].

In this paper, we propose a novel gesture recognition system based on the detection of Earth's magnetic field. The advantage of the system is that it does not require the use of artificial magnetic field sources, which simplifies the system and does not limit the user's mobility. To achieve accurate gesture recognition, we employ a 1D convolutional neural network algorithm to process the sensor data. This system has demonstrated promising results and offers a promising alternative to existing gesture recognition technologies.

2. Gesture Recognition System

In this paper, we propose a gesture recognition system that utilizes sensors which detect Earth's magnetic field worn on the user's hand to capture the gesture data. This section will provide description of the system's working principle, composition, and performance tests in detail.

2.1. Working Principle

The proposed gesture recognition system employs multiple magnetic sensors installed on the user's hand to gather gesture information. The system detects Earth's stable magnetic field as its magnetic field source. This magnetic field is similar to that of a simple bar magnet, and Figure 1 illustrates that its magnetic dipole has magnetic field lines that begin near the South pole and end near the North pole. These endpoints are referred to as magnetic poles. The strength and direction of these magnetic field lines vary across the Earth's surface, and the direction and strength of the Earth's magnetic field (H_e) can be defined by its

three-axis values H_x, H_y, and H_z. As the magnetic field sensor rotates within the Earth's magnetic field, its three-axis data output changes in response to the sensor's rotation. By attaching a magnetic field sensor to a finger, the sensor can detect the finger's flexion and rotation. Different finger bending states correspond to different gestures. By analyzing and processing the data collected by the magnetic field sensor installed on the finger, we can identify the pointing and bending state of the finger and, subsequently, achieve gesture recognition.

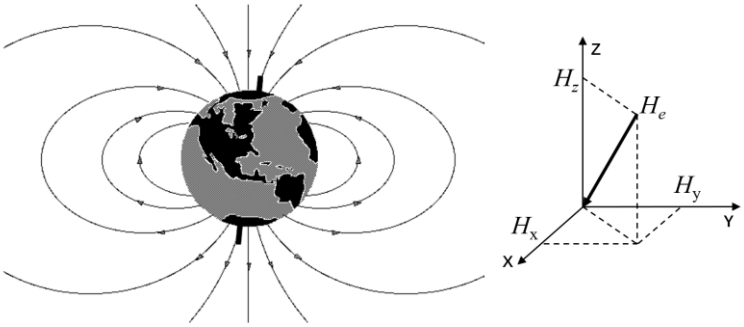


Figure 1. Earth's magnetic field.

2.2. System Composition

The system block diagram of the gesture recognition system is illustrated in Figure 2. The system's core comprises six magnetic field sensors, namely, node 1 to node 6, which are installed on the back of the hand and five fingers in a sequential manner to detect the state of hand movement. These sensors are connected to the main controller ESP32 module via the TCA9548A module, which handles the communication interfaces. The main controller collects data from the magnetic field sensors and sends it to the computer for gesture recognition processing through the USB cable. Next, we will provide detailed descriptions of the magnetic field sensor, TCA9548A module, and ESP32 module.

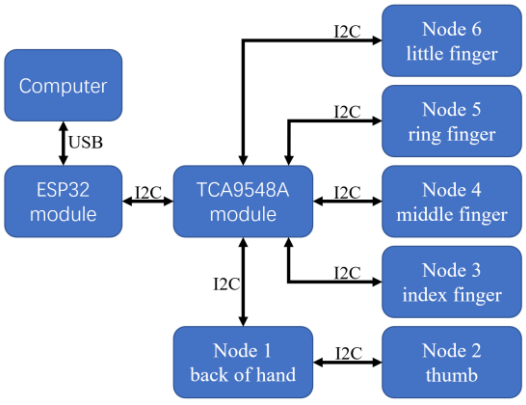


Figure 2. Schematic diagram of gesture recognition system.

The surface-level magnitude of Earth's magnetic field typically ranges between 25 to 65 μT (0.25 to 0.65 gauss). Sensors made of solenoids are bulky and have low precision, making them unsuitable for measuring the Earth's magnetic field. Instead, the Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. HMC5883L is a three-axis magnetic field sensor with a range from −8 gauss to +8 gauss and a resolution of 5 milligauss. The performance of the HMC5883L sensor meets the measurement requirements of our application scenario. HMC5883L only needs simple peripheral circuits to work, which simplifies circuit design and facilitates system integration. HMC5883L provides an I2C interface for hardware initialization and data transmission, which is a general data transmission interface. Additionally, its small 3.0 × 3.0 × 0.9 mm

LCC surface mount package allows for easy miniaturization of the magnetic field detection node and enables wearable designs. For our system, we employed six HMC5883L modules, as depicted in Figure 3.

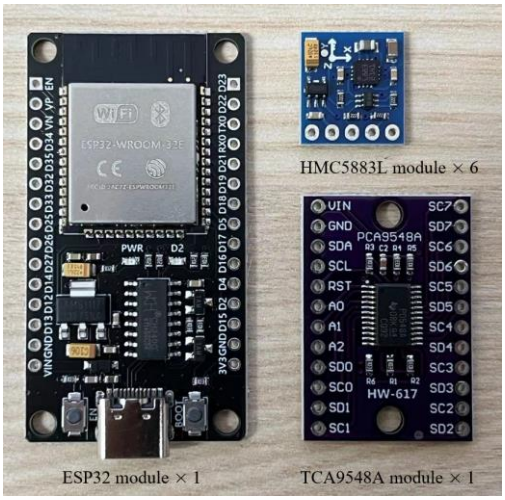


Figure 3. Gesture recognition system core module.

Each HMC5883L module is equipped with an I2C data interface, and a total of six independent I2C interfaces are available for the six sensors. However, since the I2C communication address for each HMC5883L is 0x1E, connecting all six sensors to the same I2C interface of the controller would result in I2C address conflicts and data transmission failures. Despite the controller having only one I2C interface, we are able to connect all six magnetic sensors to it using a TCA9548A module. This module features eight bidirectional transfer switches that can be controlled through the I2C bus. The upstream pair of serial clock/serial data (SCL/SDA) can be expanded to eight downstream pairs or lanes. The programmable control register enables any individual I2C pair to be selected, resolving I2C slave address conflicts. We connected the six magnetic field sensor modules in turn to channels 0–5 of the TCA9548A module.

The system controller is an ESP32 module, which is a low-power, single-chip microcomputer module with a wide range of interfaces and a competitive price. The ESP32 offers convenient development tools and a variety of development routines, which simplifies usage. The module is equipped with an I2C interface for obtaining sensor data and a USB port for both power supply and data transmission. As illustrated in Figure 2, the I2C interface of the ESP32 module is connected to the I2C control interface of the TCA9548A module. Through the TCA9548A module, each sensor is configured to gather magnetic field data. The ESP32 module then sorts the acquired three-axis magnetic field data based on the node order and transmits the sorted data to the computer via the USB cable. The sensor's acquisition frequency is set at 10 Hz.

Based on the system block diagram, the prototype of the gesture recognition system is constructed using functional modules, as depicted in Figure 4. To detect finger bending and rotation, five magnetic field sensors are installed on the fingertips, while one magnetic field sensor is installed on the back of the hand to detect palm rotation without hindering the wearer's use. The six magnetic field sensors are numbered sequentially from 1 to 6, as shown in Figure 4. For ease of installation and wearing, the magnetic field sensors and other devices are attached to a glove. The output data of the magnetic field sensors are dependent on their installation position. It is specified that, when the palm is placed horizontally and facing downwards, the positive Y-axis direction of the magnetic field sensor points towards the fingertip, while the positive *Z*-axis direction is vertically upwards.

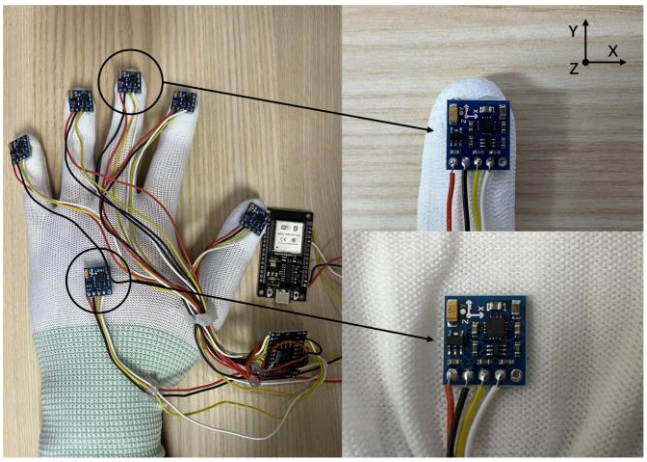


Figure 4. The prototype of the gesture recognition system.

2.3. Experimental Test

In order to verify the functionality of the gesture recognition system, the prototype was tested. The magnetic field sensors used were of the same model and each sensor's performance was found to be acceptable within the same range. To test the system functionality, the magnetic field sensor on the index finger was used. The fingertip direction θ was calculated using Equation (1), based on the magnetic field sensor three-axis data.

$$\theta = \begin{cases} 90 - \left(\arctan(\frac{H_y}{H_z})\right) \times \frac{180}{\pi} & H_z > 0 \\ 270 - \left(\arctan(\frac{H_y}{H_z})\right) \times \frac{180}{\pi} & H_z < 0 \\ 180 & H_z = 0, H_y < 0 \\ 0 & H_z = 0, H_y > 0 \end{cases} \quad (1)$$

As illustrated in Figure 5, the experimenter wore a magnetic detection glove on his left hand, formed a fist, and placed it on the table. Initially, the index finger was pointing northward horizontally, as illustrated in Figure 5a. The experimenter then rotated the index finger clockwise until it pointed due south, as shown in Figure 5c.

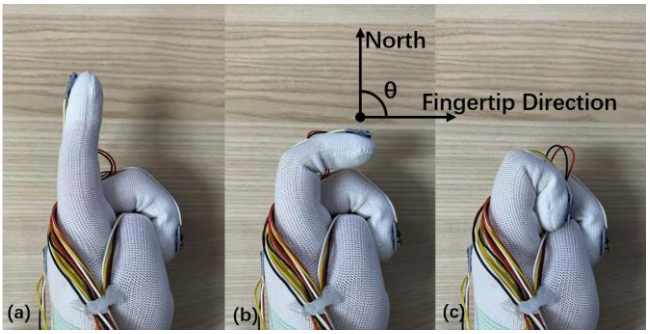


Figure 5. Experimental test. (**a**) The index finger pointing north; (**b**) pointing east; (**c**) pointing south.

Figure 6 depicts a comparative chart that presents the magnetic field sensor's calculated direction of the fingertip and the real direction of the fingertip.

The test results indicate that data collected by the magnetic field sensors can accurately reflect the degree of pointing and bending of the fingers. Since distinct gestures encompass diverse finger bending states, analyzing the magnetic field sensor data can enable us to determine the user's gestures accurately. However, the position and direction of the user's hand are continually changing during operation, making traditional threshold-based judgment unsuitable for magnetic field sensor date analysis. To address this issue, we propose a method based on a 1D convolutional neural network to process the data.

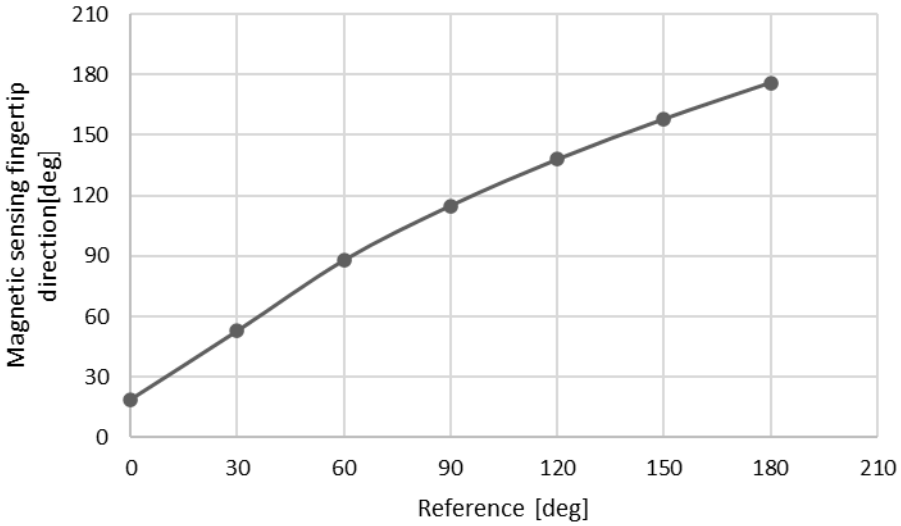


Figure 6. Comparison of measured and reference direction.

3. Deep Learning Algorithm

3.1. CNN-1D (Composition of Convolutional Neural Network)

A Convolutional Neural Network (CNN) is a deep neural network that utilizes a convolutional structure. The structure of a one-dimensional CNN comprises an input layer, convolutional layer, pooling layer, fully connected layer, and output layer, as shown in Figure 7.

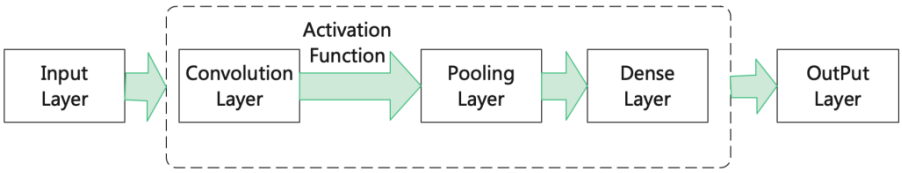


Figure 7. One-dimensional convolutional neural network (CNN-1D) structure.

The convolutional layer is the core of the convolutional neural network. Its main function is to extract the input feature information. It is composed of convolution units. The input information is moved regularly through the receptive field pair, and the features of the corresponding area are extracted. Low convolutional layers can only extract low-level features, and deep convolutional layers can extract deep features. In the convolution layer, the convolution kernel performs convolution operation on the output of the previous layer and outputs the convolution result, and its formula is expressed as:

$$y_j^l = \sum_{i=1}^{M} x_i^l \otimes k_{ij}^l + b_j^l \tag{2}$$

y_j^l is the vector obtained after the j-th convolution calculation of layer l; M is the number of input feature vectors; x_j^l is the i-th input feature vector of layer l; $\otimes$ means correlation operation; the i-th input feature vector is convolved with the j-th convolution kernel; and k_{ij}^l is the j-th bias vector of the l layer.

The convolution layer belongs to the hidden layer. After the convolution operation, the activation function is usually used to perform affine transformation on the logits output by each convolution to increase the nonlinear characteristics of the network model. Considering the convergence speed and overfitting problems, this paper uses Rectified Linear Unit (ReLU) as the hidden layer activation function. This activation function does not have the problem of gradient scattering. Compared with the Sigmoid function, it has a

faster convergence speed, and at the same time can improve the sparsity of the network and effectively prevent the overfitting problem. The function expression can be defined as:

$$ReLU(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{3}$$

After the activation function, the pooling layer [34] is usually used to reduce the dimensionality of the features, so as to reduce the amount of parameters and improve the fault tolerance rate of the model. The more commonly used pooling operations are mainly Max Pooling and Average Pooling. The former takes the maximum value of the local receptive field as the output, and the latter takes the average value of the local receptive field as the output. In this paper, the max pooling is used for downsampling, the pooling window is 2×1, the movement is performed with a stride of 2, and the input is traversed. The pooling operation of the l + 1 layer can be expressed as:

$$P_{i,m} = \max q_{i,(m-1)S+n} \tag{4}$$

Among them, $P_{i,m}$ represents the i-th feature map in the m-th layer; $q_{i,(m-1)S+n}$ represents the value of the $(m-1)S+n$ unit in the i-th feature map; and S is the size of the overlapping portion of adjacent sampling windows.

The classification layer consists of a fully connected hidden layer and a Softmax layer. The fully connected layer flattens the output of the previous pooling layer into a one-dimensional feature vector, and the first fully connected. The activation function of the hidden layer is ReLU, and Softmax is the generalization of logistic regression mainly used for multi-classification problems. The probability y_i of the sample belonging to the i-th category calculated according to Softmax can be expressed as:

$$y_i = \frac{e^{a_i}}{\sum_{k=1}^{C} e^{a_k}} \tag{5}$$

Among them, C is the number of categories to be predicted and a is the output unit of the fully connected network. Through the above formula, it can be guaranteed that, for the same sample, the sum of the probabilities belonging to each category is 1, that is, $\sum_{i=1}^{C} y_i = 1$; therefore, given an input sample x, the probability that the sample belongs to category k is:

$$p(y = k|x) = \text{softmax}(\theta_k^T, x) = \frac{\exp(\theta_k^T x)}{\sum_{j=1}^{K} \exp(\theta_j^T x)} \tag{6}$$

where θ are all training parameters in the regression model.

In order to make the output of the model consistent with the expected value, it is necessary to measure the distance between the output and the expected value, and the function to measure this distance is the loss function. For multi-classification problems, the cross-entropy loss function is generally used, and its mathematical expression is:

$$L(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} I\{y_i = k\} \log \frac{\exp(\theta_K^T x)}{\sum_{j=1}^{K} \exp(\theta_K^T x)}\right] \tag{7}$$

where m is the number of samples or the input batch size and $I\{\}$ is the indicator function. When the value of $\{\}$ is true, the value of the indicator function is 1; otherwise, the value of the indicator function is 0.

In order to minimize the loss function value of the model, it is necessary to optimize and adjust the weight of the neural network, and the optimizer uses the backpropagation algorithm to complete it. The mathematical description is:

$$\theta^* = \underset{\theta}{\text{argmin}} L(f(x); \theta, y) \tag{8}$$

where θ^* is the optimal parameter of the model; $L(\cdot)$ is the loss function; and $f(\cdot)$ and y are the output value and target value of the model, respectively.

3.2. Description of the Dataset

The American Sign Language (ASL) alphabet serves as a common method of gestural communication. We assess the viability of our proposed approach by accurately recognizing the ASL alphabet. In this study, one experimenter wore a glove with magnetic sensors on the left hand. The experimenter acquired knowledge of the American Sign Language alphabet by studying video tutorials. The experimenter was instructed to use hand gestures to represent the 26 letters of the American Sign Language alphabet. Figure 8 shows three gestures A, B, and C. The system used a sampling frequency of 10 times per second and recorded each gesture for 120 s, generating 1200 data points for each gesture. Each data point sequentially comprises three-axis magnetic field data for the back of the hand, thumb, index finger, middle finger, ring finger, and little finger. To analyze the data, we saved the magnetic gesture data for each letter in separate txt files, resulting in 26 files in total.

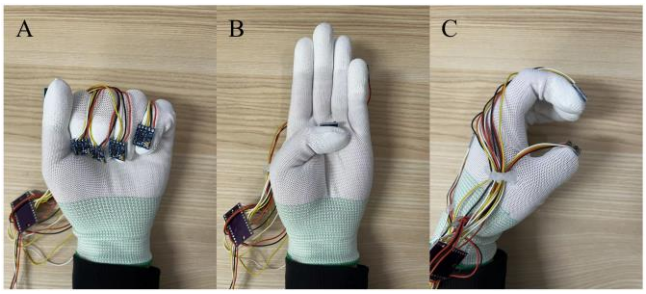


Figure 8. Gestures for three letters: A, B, and C.

After the data is preprocessed and labeled, the data set is randomly divided into a training set, validation set, and test set according to the ratio of 6:2:2. The input data size is shown in the Table 1.

Table 1. Dataset partitioning.

Set	Data	Target	Percentage
Train Set	(720, 18, 1)	(720, 1)	60%
Validation Set	(240, 18, 1)	(240, 1)	20%
Test Set	(240, 18, 1)	(240, 1)	20%

3.3. Models

Based on the above-mentioned one-dimensional convolutional neural network, this paper constructs a gesture classification model CNN-Ges-Cla, and its network structure is shown in Figure 9.

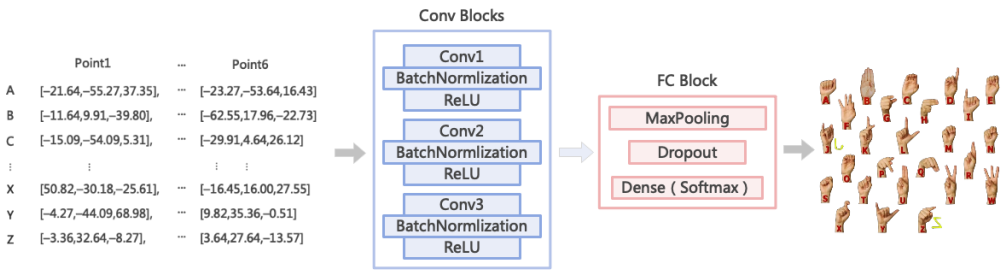


Figure 9. CNN-Ges-Cla network structure diagram.

The entire network consists of three convolutional modules—the Conv Block—and a fully connected module, Dense Block; each convolutional module contains a one-dimensional convolutional layer (Conv), batch normalization layer [35], and ReLU activation function

layer. The number of convolution kernels of the three convolutional layers are 16, 32, and 64, respectively. The fully connected module contains a max pooling layer, and the number of output neurons in the fully connected layer is 26. The parameter configuration of the trainable layer in the network structure is shown in Table 2.

Table 2. Parameter configuration of the trainable layer.

Layer Name	Kernel Size	Kernel Number	Padding	Stride
Conv1	3×3	16	0	1
Conv2	3×3	32	0	1
Conv3	3×3	64	0	1
FC	-	26	-	-

In addition to using the above-mentioned CNN-1D model, this paper also uses a long-term short-term memory network (LSTM) to construct a model LSTM-Ges-Cla for comparative experiments. The network structure of LSTM-Ges-Cla consists of two LSTM cells (LSTM Cell) and two fully connected layers, and the number of output nodes of each fully connected layer is 64 and 26, respectively. The number of hidden dimensions in each LSTM Cell is set to 20.

4. Results and Discussion

This paper employs Ubuntu18.04 as its operating system and accelerates training using CUDA10.0 and cuDNN7.4. The NVIDIA RTX 3080 is used as the hardware platform. PyTorch1.10 serves as the network development framework, with Python as the programming language. The model utilizes an Intel®Core™i5-9400F CPU@2.90 GHz as the CPU. The model trains for 80 rounds with the optimization algorithm SGD, utilizing an initial learning rate of 2×10^{-4} and a batch size of 128. Additionally, given the small data dimension, the model employs a dropout strategy with a parameter of 0.3 to prevent overfitting.

In this paper, the confusion matrix is used as the evaluation standard to evaluate the model, and the expression of the confusion matrix is shown in the formula.

$$\begin{array}{c} \\ 1 \\ \vdots \\ N \end{array} \begin{array}{c} \begin{array}{ccc} 1 & \cdots & N \end{array} \\ \begin{bmatrix} n_{11}^T & \cdots & n_{1N}^T \\ \vdots & \ddots & \vdots \\ n_{N1}^T & \cdots & n_{NN}^T \end{bmatrix} \\ \begin{array}{ccc} P_1 & \cdots & P_N \end{array} \end{array} \begin{array}{c} \\ C_1 \\ \vdots \\ C_N \end{array} \tag{9}$$

Each column of the confusion matrix represents the predicted category, each row represents the true attribution category of the data, and the data summary of each row indicates the number of data instances of this category. Among them, nA *ij* represents the number of category *i* predicted as category *j*; when *i* = *j*, it means that the prediction is accurate, A = T, and when *i* = *j*, it means that the prediction is wrong, A = F. Pi represents the total number of predicted categories as class *i* and C represents the total number of true classes *i*.

According to the confusion matrix, the precision rate (Precision), recall rate (Recall), and F1 value can be calculated separately to further evaluate the model. The F1 value is a balance point between Precision and Recall, which can measure the distribution of the test population; its calculation formula is as follows:

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{11}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (12)$$

Among them, *TP* is True Positive, *FP* is False Positive, *FN* is False Negative, and *TN* is True Negative.

The trained model is classified using the test set. The confusion matrix is shown in Figures 10 and 11, and the Precision, Recall, and F1 scores are shown in Table 3.

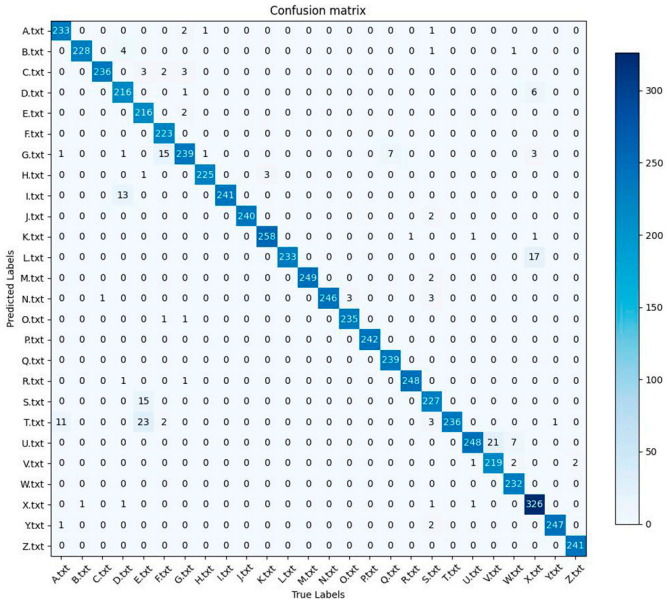


Figure 10. Confusion matrix of CNN-Ges-Cla model.

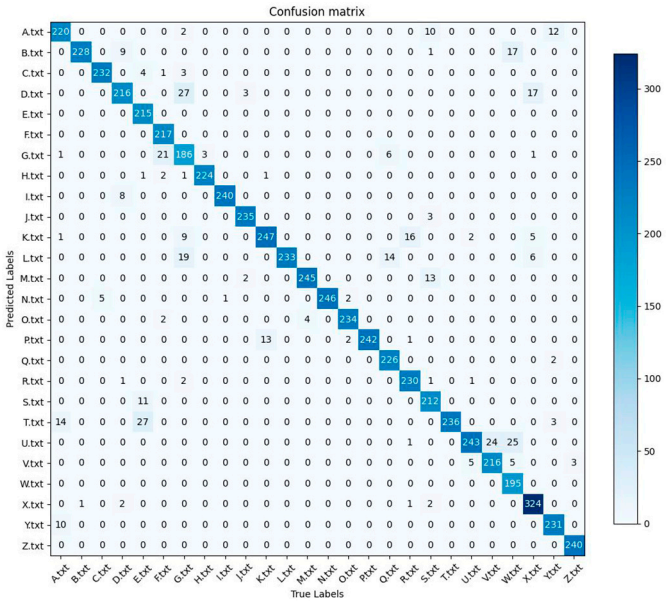


Figure 11. Confusion matrix of LSTM-Ges-Cla.

Table 3. Classification indicators for the two models.

Model	Precision	Recall	F1
CNN-Ges-Cla	0.9842	0.9778	0.9757
LSTM-Ces-Cla	0.9423	0.9312	0.9345

The table above clearly indicates that both models achieve evaluation indicators exceeding 93%. However, the model based on a one-dimensional convolutional neural network performs exceptionally well, with indicators almost reaching 97%. This gives it an advantage over the LSTM-based model. The reason for this discrepancy could be that the interdependence of sample data in this paper, in the time dimension, is not strong enough to extract deep feature information based on sequence classification. Therefore, the paper proposes the use of the CNN-Ges-Cla model, which is based on CNN-1D, to classify the collected gesture signals.

The use of magnetic detection technology based on the geomagnetic field has been widely adopted in the field of indoor positioning. However, this paper proposes an innovative application of this technology to the field of gesture recognition. The proposed method involves using six geomagnetic field sensors installed on the human hand to obtain gesture data. We trained a 1D convolutional neural network using magnetic field data from predefined gestures, and the accuracy of gesture recognition reached close to 97% under experimental conditions. The results of the experiment confirm the feasibility of using magnetic detection technology based on the geomagnetic field in gesture recognition. Moreover, the application of the one-dimensional convolutional neural network algorithm compensates for the lack of stability in gesture recognition when using one single type of sensor. With the implementation of wireless communication between sensors, the practicality of the magnetic field sensor glove will be enhanced. However, it is important to note that the experiment has only tested 26 gestures defined by the American Sign Language alphabet. Further research is necessary to enhance recognition of dynamic and complex gestures.

5. Conclusions

This paper proposes a novel gesture recognition system using multiple sensors based on Earth's magnetic field and a 1D convolution neural network. Compared with traditional gesture recognition technology, gesture recognition technology based on magnetic sensors has numerous advantages, including increased accuracy in recognition and immunity to environmental interference. Our proposed system detects gestures by installing multiple magnetic field sensors on the fingertips and back of the palm, with data collected from these sensors classified using a 1D convolution neural network. To validate the system's efficacy, we conducted an empirical test using American Sign Language alphabet gestures, and the results demonstrated an exceptional recognition accuracy of nearly 97% under experimental conditions. Further investigation is required to effectively recognize more complex and dynamic gestures.

The future holds vast application possibilities for magnetic-sensing gesture-recognition technology. Its exceptional precision and reliability make it ideal for various computer graphics applications such as virtual and augmented reality, providing a more natural and authentic human–computer interaction. Moreover, this technology can enhance the ease of device control for individuals in healthcare, smart homes, and other areas. Additionally, magnetic-sensing gesture-recognition technology can facilitate barrier-free technology, assistive tools, and other fields to offer greater convenience for individuals with disabilities.

Author Contributions: Conceptualization, X.C.; methodology, B.S.; software, B.S.; validation, B.S. and Z.H.; formal analysis, H.S.; investigation, X.C. and R.H.; resources, X.C.; data curation, H.S.; writing—original draft preparation, B.S.; writing—review and editing, X.C. and R.H.; visualization, B.S. and Z.H.; supervision, X.C.; project administration, R.H.; funding acquisition, X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (51777010).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Naglot, D.; Kulkarni, M. Real Time Sign Language Recognition Using the Leap Motion Controller. In Proceedings of the International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; pp. 1–5.
2. Dong, C.; Leu, M.C.; Yin, Z. American Sign Language Alphabet Recognition Using Microsoft Kinect. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 7–12 June 2015; pp. 44–52.
3. Sharma, S.; Kumar, K. ASL-3DCNN: American Sign Language Recognition Technique Using 3-D Convolutional Neural Networks. Multimed. *Tools Appl.* **2021**, *80*, 26319–26331. [CrossRef]
4. Pansare, J.R.; Ingle, M. Vision-Based Approach for American Sign Language Recognition Using Edge Orientation Histogram. In Proceedings of the International Conference on Image, Vision and Computing (ICIVC), Portsmouth, UK, 3–5 August 2016; pp. 86–90.
5. Mohamed, N.; Member, G.S. A Review of the Hand Gesture Recognition System: Current Progress and Future Directions. *IEEE Access* **2021**, *9*, 157422–157436. [CrossRef]
6. Jiang, S.; Kang, P.; Song, X.; Member, S.; Lo, B.P.L.; Member, S.; Shull, P.B. Emerging Wearable Interfaces and Algorithms for Hand Gesture Recognition: A Survey. *IEEE Rev. Biomed. Eng.* **2021**, *15*, 85–102. [CrossRef] [PubMed]
7. Savur, C.; Sahin, F. American Sign Language Recognition System by Using Surface EMG Signal. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 2872–2877.
8. Jani, A.B.; Kotak, N.A.; Roy, A.K. Sensor Based Hand Gesture Recognition System for English Alphabets Used in Sign Language of Deaf-Mute People. In Proceedings of the IEEE SENSORS, New Delhi, India, 28–31 October 2018; pp. 1–4.
9. Ford, L.K.; Borneman, J.D.; Can, C.; Kaya, Y.; Sign, I.; Using, L.; Lite, T.; Fadlilah, U.; Mahamad, A.K.; Handaga, B. Development of a Wearable Device for Sign Language Recognition. *J. Phys. Conf. Ser.* **2018**, *1019*, 012017.
10. Wu, Y.T.; Gomes, M.K.; da Silva, W.H.; Lazari, P.M.; Fujiwara, E. Integrated Optical Fiber Force Myography Sensor as Pervasive Predictor of Hand Postures. *Biomed. Eng. Comput. Biol.* **2020**, *11*, 117959722091282. [CrossRef]
11. Wen, H.; Rojas, J.R.; Dey, A.K. Serendipity: Finger Gesture Recognition Using an off-the-Shelf Smartwatch. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 7–12 May 2016; pp. 3847–3851.
12. Fahn, C.S.; Sun, H. Development of a Fingertip Glove Equipped with Magnetic Tracking Sensors. *Sensors* **2010**, *10*, 1119–1140. [CrossRef]
13. Chan, T.K.; Yu, Y.K.; Kam, H.C.; Wong, K.H. Robust Hand Gesture Input Using Computer Vision, Inertial Measurement Unit (IMU) and Flex Sensors. In Proceedings of the IEEE International Conference on Mechatronics, Robotics and Automation (ICMRA), Hefei, China, 18–21 May 2018; pp. 95–99.
14. Friedman, N.; Rowe, J.B.; Reinkensmeyer, D.J.; Bachman, M. The Manumeter: A Wearable Device for Monitoring Daily Use of the Wrist and Fingers. *IEEE J. Biomed. Health Inform.* **2014**, *18*, 1804–1812. [CrossRef]
15. Siddiqui, N.; Chan, R.H.M. Hand Gesture Recognition Using Multiple Acoustic Measurements at Wrist. *IEEE Trans. Hum.-Mach. Syst.* **2021**, *51*, 56–62. [CrossRef]
16. Jiang, S.; Lv, B.; Guo, W.; Zhang, C.; Wang, H.; Sheng, X.; Shull, P.B. Feasibility of Wrist-Worn, Real-Time Hand, and Surface Gesture Recognition via SEMG and IMU Sensing. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3376–3385. [CrossRef]
17. Cho, S.G.; Yoshikawa, M.; Ding, M.; Takamatsu, J.; Ogasawara, T. Machine-Learning-Based Hand Motion Recognition System by Measuring Forearm Deformation with a Distance Sensor Array. *Int. J. Intell. Robot. Appl.* **2019**, *3*, 418–429. [CrossRef]
18. Zimmerman, T.G.; Lanier, J.; Blanchard, C.; Bryson, S.; Harvill, Y. Hand Gesture Interface Device. *ACM Sigchi Bull.* **1987**, *18*, 189–192. [CrossRef]
19. Bellitti, P.; De Angelis, A.; DIonigi, M.; Sardini, E.; Serpelloni, M.; Moschitta, A.; Carbone, P. A Wearable and Wirelessly Powered System for Multiple Finger Tracking. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 2542–2551. [CrossRef]
20. Pasku, V.; De Angelis, A.; De Angelis, G.; Member, S.; Arumugam, D.D.; Dionigi, M.; Carbone, P.; Moschitta, A.; Ricketts, D.S. Magnetic Field-Based Positioning Systems. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2003–2017. [CrossRef]
21. Rinalduzzi, M.; De Angelis, A.; Santoni, F.; Buchicchio, E.; Moschitta, A.; Carbone, P.; Bellitti, P.; Serpelloni, M. Gesture Recognition of Sign Language Alphabet Using a Magnetic Positioning System. *Appl. Sci.* **2021**, *11*, 5594. [CrossRef]
22. Santoni, F.; De Angelis, A.; Moschitta, A.; Carbone, P. A Multi-Node Magnetic Positioning System with a Distributed Data Acquisition Architecture. *Sensors* **2020**, *20*, 6210. [CrossRef]
23. Fahn, C.S.; Sun, H. Development of a Data Glove with Reducing Sensors Based on Magnetic Induction. *IEEE Trans. Ind. Electron.* **2005**, *52*, 585–594. [CrossRef]
24. Santoni, F.; De Angelis, A.; Moschitta, A.; Carbone, P. MagIK: A Hand-Tracking Magnetic Positioning System Based on a Kinematic Model of the Hand. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 9507313. [CrossRef]
25. Siriborvornratanakul, T. Human Behavior in Image-Based Road Health Inspection Systems despite the Emerging AutoML. *J. Big Data* **2022**, *9*, 96. [CrossRef]

26. Siriborvornratanakul, T. A New Human Factor Study in Developing Practical Vision-Based Applications with the Transformer-Based Deep Learning Model. In Proceedings of the International Conference on Human-Computer Interaction, AI-HCI, Virtual Event, 26 June–1 July 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 436–447.
27. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
28. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [CrossRef]
29. Richardson, E.; Alaluf, Y.; Patashnik, O.; Nitzan, Y.; Azar, Y.; Shapiro, S.; Cohen-Or, D. Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 2287–2296.
30. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 6999–7019. [CrossRef] [PubMed]
31. Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, K.J. Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Trans. Acoust.* **1989**, *37*, 328–339. [CrossRef]
32. Dao, Q.; El-Yacoubi, M.A.; Rigaud, A.S. Detection of Alzheimer Disease on Online Handwriting Using 1D Convolutional Neural Network. *IEEE Access* **2023**, *11*, 2148–2155. [CrossRef]
33. Liu, X.; Xiong, S.; Wang, X.; Liang, T.; Wang, H.; Liu, X. A Compact Multi-Branch 1D Convolutional Neural Network for EEG-Based Motor Imagery Classification. *Biomed. Signal Process. Control* **2023**, *81*, 104456. [CrossRef]
34. Fukushima, K. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biol. Cybern.* **1980**, *36*, 193–202. [CrossRef]
35. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning ICML, Lille, France, 6–11 July 2015; pp. 448–456.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.