





Research Article

Software Simulation in English Translation Platform Based on Gaussian Hybrid Model

Nan Peng ¹, Yingying Wei ², Lu Liu ¹, and Junzhong Zou ³

¹Department of Basic Teaching and Research, East University of Heilongjiang, Heilongjiang, Harbin 150066, China

²Department of Management, East University of Heilongjiang, Heilongjiang, Harbin 150066, China

³Department of Human Resources, East University of Heilongjiang, Heilongjiang, Harbin 150066, China

Correspondence should be addressed to Lu Liu; liulu@hljeu.edu.cn

Received 26 June 2022; Revised 22 July 2022; Accepted 4 August 2022; Published 21 August 2022

Academic Editor: Shadi Aljawarneh

Copyright © 2022 Nan Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this article, we will first try to create a general development platform for embedded systems. The goal of this step is to establish an experimental platform that can support various peripheral modules and can be reused. The connection between the modules can be reconfigured to meet the different needs of embedded research and learning. On this basis, the system uses the audio frequency spectrum program as an input function to represent the dispersion pattern of signal energy, which can be tested with characteristic nodes simulated by convolutional neural network software. In addition, the auxiliary neural network software simulation core can continuously learn the detailed characteristics of the audio frequency spectrum, making it easy to recognize environmental sounds. In addition, the sound signal and the neural cycle are related in time. The neural network can study the relationship between different frames in the time domain to compensate for the defects caused by the complex neural network in modeling time series. Finally, this article focuses on the process of building an English translation platform based on the mobile cloud data model. The client is targeted at the Android platform, while the server is based on the IaaS system. According to the model of mobile cloud data processing, the calculation of mobile phones in computer-intensive programs is studied. Through the hardware design and distribution of the system, we are able to use mobile cloud technology as a desktop-intensive program to solve the problem of effectiveness in the solution. The article promotes the development of an English translation platform by applying the research results of environmental sound recognition based on embedded system software simulation to the design of the English translation platform.

1. Introduction

This article first analyzes the actual application background of the embedded system and then develops and designs a general experimental platform composed of embedded processor core modules, custom connection modules, and external facility modules based on application requirements [1]. How to select, design, write, and test the hardware peripheral modules of the platform, and transplant basic drivers for each module is the focus of this article [2]. The common experimental platform is built with an embedded STM32 processor [3]. It has the advantages of high performance and low cost. It can be used as the core module of the experimental system. We have also

developed a motherboard supporting circuit for this core module [4]. On this basis, it has deeply studied how to improve the data of environmental sound recognition and proposed an improvement plan for the problem of environmental sound recognition in the online mode [5]. At present, the amount of data contained in the public environmental sound data set is relatively small, and it is difficult for the model to obtain good generalization performance in limited training [6]. In this article, we describe the data expansion method of existing software simulation and then propose an online data expansion program based on this technology [7]. The improved scheme directly processes the spectrogram of the input sound in the training phase, which not only provides a wide range of

training samples, but also improves the flexibility of the system, and significantly improves the recognition performance of the proposed extension scheme on several public data sets [8]. Finally, due to the high development cost of traditional server-side equipment, there are cumbersome upgrades, maintenance, management, and expansion complexity, as well as other key issues that affect the performance of computing-intensive mobile device applications. Finally, this article introduces the English translation platform based on OpenNebula's open source platform in detail [9]. The author provides a mobile cloud computing environment for testing in combination with an online translation system [10]. Through basic research on the development model and system architecture of mobile-intensive applications, we can optimize the development, operation, deployment, and use of traditional mobile applications and effectively manage virtual resources. Finally, it provides network services with excellent scalability and realizes a dynamically scalable system model. On this basis, combined with the key technologies discussed in this article, we make full use of the advantages of mobile cloud computing and finally provide an English translation platform in the mobile cloud computing model [11]. In order to verify the effectiveness of the development model and system architecture proposed in this article, the platform will be tested by simulation testing [12].

2. Related Work

The literature introduces the overall framework of computing-intensive mobile application development based on the mobile cloud computing model. According to the analysis of system requirements, the key technologies involved in the mobile cloud computing model are integrated with OpenNebula, and the specific implementation methods and methods of the online translation system are given to process and verify its feasibility [13]. The literature introduces the theoretical basis of environmental sound recognition and deep learning [14]. First, the typical system framework, common feature extraction methods, and typical features, common classification algorithms, and evaluation indicators of environmental sound recognition tasks are analyzed and described in detail, and then the theoretical basis of deep learning is introduced [15]. The literature introduces the advantages and disadvantages of deep learning basic network structure applied to environmental sound recognition tasks, and it designs a convolutional recurrent neural network structure to learn from the characteristics of sound spectrograms [16]. The literature introduces existing sound data enhancement methods and proposes an online data enhancement scheme, including mask enhancement and hybrid enhancement, which directly act on each batch of data in the model training process. According to different parameter settings, we have designed several enhancement strategies and conducted an experimental analysis [17]. The literature introduces the hardware circuit design of the experimental platform's network connection module, audio signal receiving

module, TFT liquid crystal display module, data storage card module, and the general layout of the embedded system's general experimental platform [18].

3. Embedded System Software Simulation and Environmental Sound Recognition

3.1. Embedded System. The processor of the embedded system is the central component of the entire system, and it is the key to check the operation of the entire system. At present, the main embedded processors on the market can be divided into the following categories:

Chip programming system: it is a hardware programming system in which the software core of the microcontroller can be loaded into the chip. The system will automatically call other required hardware IP modules.

DSP digital signal processor: this processor is designed to deal with the data transmission problem of digital signals. The DSP has a dedicated hardware multiplier, is capable of pipeline operation, and provides dedicated DSP instructions.

Embedded Microprocessor (MPU): it is a standard computer processor and is currently the most widely used processor. And the cost is low, and it has an excellent performance in temperature control and reliability.

Embedded microcontroller (MCU): it is also called a single-chip microcomputer, its structure and performance are not as good as MPU, and its utilization rate is low.

According to the different function types of each microprocessor, the choice of the peripheral circuit of the embedded system is also very different. The basic module of the peripheral circuit is usually mainly composed of the power management circuit, the data storage circuit, and the operating program module. The existence of all submodules is necessary for the normal operation of the microprocessor. The external hardware of the embedded system will change according to the needs of the system, and usually the input device will be changed according to the different external hardware that needs to be configured.

In some more complex embedded systems, an embedded operating system is usually required. Generally speaking, the operating system has two basic functions: to promote the correct use of system resources by applications and to manage hardware devices. In addition, multitasking is also an important function of embedded operating systems that can work in real time.

Under normal circumstances, embedded system development consists of two parts: program development and hardware development. The development process is mainly composed of 4 parts, which are

Overall structural design: it is the overall structural framework of the absolute system, including the division of system programs and hardware functions and

the assignment of target tasks, as well as the compatibility of other hardware.

Co-design of hardware and software: according to the requirements of the system structure, detailed design comparisons of the system hardware and software versions are carried out respectively.

Requirement analysis: after analyzing the characteristics of the system and the use environment, the goals and tasks are determined, and unified standards are used as the guidelines for system design.

System test: we will perform a complete machine test on the combined system after debugging to verify whether it meets the functional requirements in the setting.

3.2. Basic Principles of Environmental Sound Recognition.

Environmental speech recognition is an important issue in speech recognition. The goal of many existing systems is to be able to accurately predict the characteristics of certain types of sounds. As shown in Figure 1, we input a speech signal that can express its classification characteristics into the environmental speech recognition system. In this way, regardless of the internal working environment of the environmental speech recognition system, the speech recognition system will issue a predetermined category, such as “dog barking.”

Preprocessing the data is a preliminary process for extracting environmental acoustic characteristics. The data are imported into a unified specification by measuring the dimensionality of the original data to ensure its consistency. In the actual test, due to the fluctuation of the collection environment, the collected data are usually noisy. We can use the end marker to solve this problem.

Highlighting the particularity of environmental speech is a key step for the system to solve the problem of accuracy. In the actual test, the range of the noise configuration is very high, and the difference between different configurations has not been studied yet. Therefore, the important step is to use the signal processing source to extract more features of the sound configuration and to highlight the salient features of the entire signal with a lower dimension. The recognition performance of a system usually depends on the ability to highlight the sound features. In the work of environmental speech recognition, common features include short-range energy, sound vibration amplitude, zero-crossing level, mile spectrum, mile frequency interval coefficient, and so on.

The classification model is a necessary functional module of the environmental speech recognition system. The classification model is usually based on the existing labeled data, is trained and updated continuously in the repeated process, and finally integrated into the validation set. In addition, it requires a lot of calculations to classify the distance between different sounds and data in the sample features.

3.3. Algorithm Model. At present, most of the sound features are converted into data by the Fourier transform formula. But this article also refers to other frequency domain

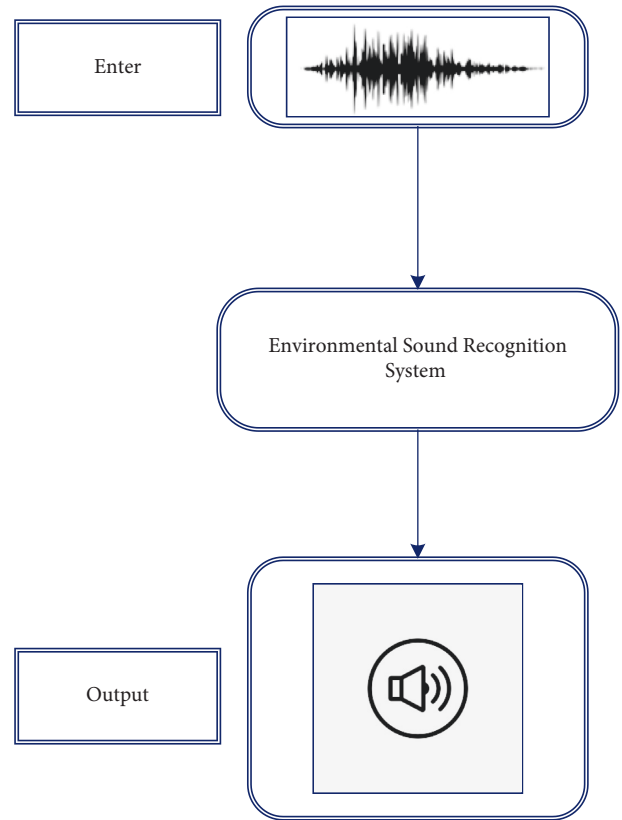


FIGURE 1: Overview of the environmental sound recognition system.

conversion methods, such as constant Q conversion (CQT) and continuous violet conversion (CWT).

Zero crossing rate (ZCR) is an important feature of instantaneous sounds (such as percussive sounds), and its calculation formula is as follows:

$$Z = \frac{1}{2(n-1)} \sum_{n=1}^{N-1} |\text{sgn}(x(n+1)) - \text{sgn}(x(n))|. \quad (1)$$

In the formula, (n) represents the signal, N is the sampling point, and (\bullet) is the symbolic function, and we obtained

$$\text{sgn}(x(n)) = \begin{cases} 1, & x(n) \geq 0, \\ -1, & x(n) < 0. \end{cases} \quad (2)$$

STE describes the distribution of the signal in the time domain and expressed as

$$E_n = \frac{1}{N} \sum_m x(m)w(n-m)^2, \quad (3)$$

where (n) is the window function and we obtained

$$w(n) = \begin{cases} 1, & 0 \leq n \leq N-1, \\ 0, & \text{others.} \end{cases} \quad (4)$$

The response of the triangle filter is

$$H_m(k) = \begin{cases} 0, & k < f(m-1), \\ \frac{2(k-f(m-1))}{(f(m+1)-f(m-1))(f(m)-f(m-1))}, & f(m-1) \leq k \leq f(m), \\ \frac{2(f(m+1)-k)}{(f(m+1)-f(m-1))(f(m)-f(m-1))}, & f(m) \leq k \leq f(m+1), \\ 0, & k \geq f(m+1). \end{cases} \quad (5)$$

The energy spectrum after frequency domain conversion is as follows:

$$S(m) = \sum_{k=0}^{N-1} |X(k)|^2 H_m(k). \quad (6)$$

Here, (k) is the amplitude spectrum after Fourier transform.

After applying a filter to the sound spectrum, a logarithmic transformation is usually required. The frequency spectrum of a sound signal is composed of high frequency and low frequency. The low frequency can reveal the spectrum network, and the high frequency shows the details of the spectrum. The sound signal spectrum includes important information for formants to distinguish different tones. Converting the high- and low-frequency parts of the spectrum from multiplication to addition makes it easy to perform calculations. It behaves as

$$\ln(X(k)) = \ln(H(k)E(k)) = \ln(H(k)) + \ln(E(k)). \quad (7)$$

Cepstrum is a coefficient characteristic obtained by filtering the sound signal using DCT (Discrete Cosine Transform). Among them, DCT is an important measure to make further processing on the basis of Fourier exchange to minimize the signal filter. The main feature after the change is in the lower part. The DCT transformation form is as follows:

$$C(n) = \sum_{m=0}^{N-1} \ln(S(m)) \cos\left(\frac{\pi n(m-0.5)}{M}\right), \quad 0 \leq n \leq L. \quad (8)$$

Optimize as follows:

$$\begin{aligned} \min_{\omega, b} \frac{1}{2} \omega^T \omega, \\ \text{s.t. } y_i(\omega^T x_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned} \quad (9)$$

The training model is not necessarily a new line, but the current equation improvement cannot get a solution. For this particular problem, we introduced the Soft Margin concept and built an optimized system with a small number of models. So that you can write an improvement plan accordingly:

$$\min_{\omega, b, \xi_i} \frac{1}{2} \omega^T \omega + C \sum_{i=1}^m \xi_i, \quad (10)$$

$$\text{s.t. } y_i(\omega^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \quad i = 1, 2, \dots, m.$$

The SVM method can introduce core functions under nonlinearity, drag samples into high-dimensional space, and separate them directly. Expressing the function vector of (in) in the high-dimensional space as $\phi(x)$, we obtained

$$f(x) = \omega^T \Phi(x) + b = \sum_{i=1}^m \alpha_i y_i \kappa(x, x_i) + b. \quad (11)$$

The Gaussian Hybrid Model (GMM) is a classic application model that solves the data collection problem by distributing samples. Therefore, the Gaussian Hybrid Model is also considered an extension of the K-means algorithm. This model assumes that data propagation is dominated by a large number of Gaussian components with independent parameters, which is defined as follows:

$$G(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \quad (12)$$

We use σ to denote the variance and μ to denote the mean, and we obtained

$$P(x) = \sum_i^K w_i G(x, \mu, \Sigma_i). \quad (13)$$

The neuron output y is calculated as follows:

$$y = \sigma(w_1 \cdot x_1 + w_2 \cdot x_2 + b). \quad (14)$$

Accuracy is the most common weighted index for classification problems. Accuracy refers to the relative ratio between the number of samples that are correctly classified and the total. The following is the accuracy value of the classification model f in the data set D :

$$\text{accuracy}(f, D) = \frac{1}{N} \sum_{i=1}^N I(f(x_i) = y_i). \quad (15)$$

Among them, (x_i, y_i) represents the feature and category label of the i th sample, N is the total number of samples, and I is the indicator function, defined as follows:

$$I(x) = \begin{cases} 1, & x \text{ is true,} \\ 0, & x \text{ is false.} \end{cases} \quad (16)$$

Windowing: the number of frames is replaced with a window function to ensure continuity and attenuation on both sides of the frame. The most common ones are Hamming window and the Hanning window. After comparison, it is found that the Hamming window is controlled by different weighting coefficients, which can better avoid the problem of spectrum leakage after the Fourier transform. The expression of Hamming window is as follows:

$$W(n, a) = (1 - a) - a \times \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1. \quad (17)$$

FFT transformation: most of the characteristics of the sound signal are hidden in the frequency domain information, so it is necessary to convert the sound signal to the frequency domain to analyze its energy distribution pattern. Fast Fourier transform is one of the most common and effective methods. The transform form of Fast Fourier transform is as follows:

$$X(k) = \sum_{n=0}^{N-1} x_w(n) e^{-j2\pi kn/N}, \quad 0 \leq k \leq N-1. \quad (18)$$

Gammatone filter: Gammatone filter is the result of multiplying the Gamma distribution function and the sine wave, and its calculation formula is defined as follows:

$$g(t) = Kt^{(n-1)} e^{-2\pi Bt} \cos(2\pi f_c t + \varphi), \quad t > 0. \quad (19)$$

The energy spectrum converted by the Gammatone filter is expressed as follows:

$$S(m) = \sum_{k=0}^{N-1} |X(k)|^2 H_m(k). \quad (20)$$

Calculate the logarithmic energy: do a log transformation of the energy spectrum obtained through the Gammatone filter to obtain the logarithmic energy spectrum, as follows:

$$LGS(m) = \ln(S(m)) = \ln\left(\sum_{k=0}^{N-1} |X(k)|^2 H_m(k)\right). \quad (21)$$

In the research conducted in this article, we need to find a function that can describe the relationship between the input set of the Ks component and the output target signal I , and listen to the normal P distribution (Ks, I). To this end, we define the missing function \mathcal{L} to measure the difference between the model predictions $f(k)$, where $(k, i) \sim P$, the mean decreases. Minimizing the loss of data distribution caused by \mathcal{L} is called expected risks:

$$R(f) = \int \mathcal{L}(f(x), y) dP(x, y). \quad (22)$$

The approximate distribution obtained by using the training data set D is called empirical distribution.

$$P_\delta(x, y) = \frac{1}{n} \sum_{i=1}^n \delta(x = x_i, y = y_i). \quad (23)$$

The corresponding optimization method is called empirical risk minimization.

$$R_\delta(f) = \int \mathcal{L}(f(x), y) dP_\delta(x, y) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(x_i), y_i). \quad (24)$$

Time Stretch refers to the scale transformation of the sound waveform in the time dimension. The goal is to control the sound signal to be accelerated or decelerated.

$$z = \text{time_stretch}(y, \text{rate}). \quad (25)$$

Pitch Shift refers to the scale transformation of the sound waveform in the frequency dimension. The goal is to increase or decrease the pitch of the sound (Pitch).

$$z = \text{pitch_shift}(y, n\text{steps}). \quad (26)$$

Adding Noise means to mix the sample with another signal containing a different acoustic scene or background noise to enhance the diversity of the sample, which is expressed as follows:

$$z = (1 - w) \cdot x + w \cdot y. \quad (27)$$

Among them, y is the original sound signal, x is the background noise signal, z is the sample after mixing, and w is a weight parameter that represents the mixing ratio of the signal.

3.4. Simulation Analysis. In addition, because the hyperparameter α in the hybrid improvement method will change the mixing ratio of random samples, we conducted a differential test on different α values to test the effect of some α hyperparameters on the realization of the model. It can be seen from Figure 2 that when the abscissa $\alpha = 0.2$, the model satisfies the true reliability of the variables ESC-10, DCASE2016 Dev, and DCASE2016 Eval. According to ESC50 data collection, the average difference in the model is $\alpha = 0.3$, which is higher than expected, but the performance difference is no different from $\alpha = 0.2$. Therefore, it can be concluded that the change of the hyperparameter α has little effect on the performance of the test model, and is not the main influencing factor.

After the data are improved, the training samples are reprocessed, and the sample input text and the high-dimensional spatial distribution of the model research are modified to better study the impact of the previously proposed data enhancement scheme on the model. We can use key point calculation (PCA) to reduce the output size and data spread of the relevant link layer. Specifically, you can see the following figure. Figure 3(a) shows the data distribution without data improvement, and Figure 3(b) shows the data distribution after data improvement.

Figure 3 shows that there is a very significant degree of difference between the data distribution using the data

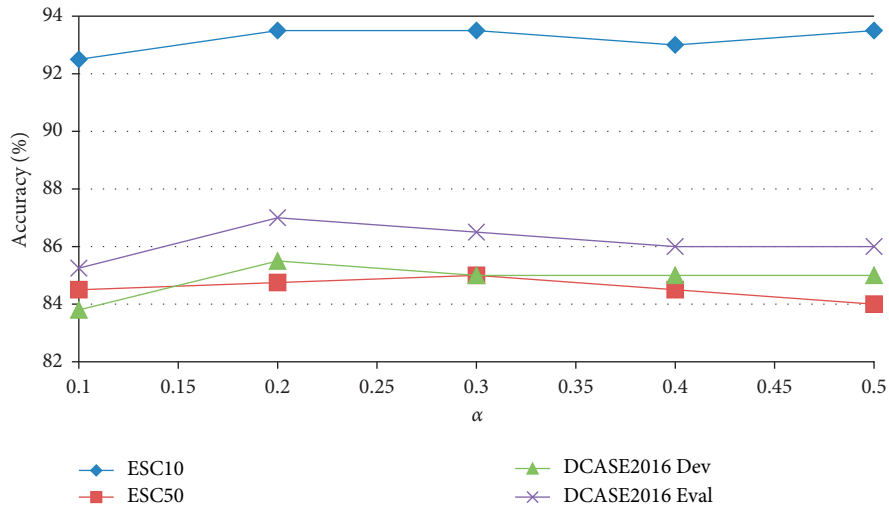


FIGURE 2: Comparison of recognition performance under different values of α in hybrid enhancement.

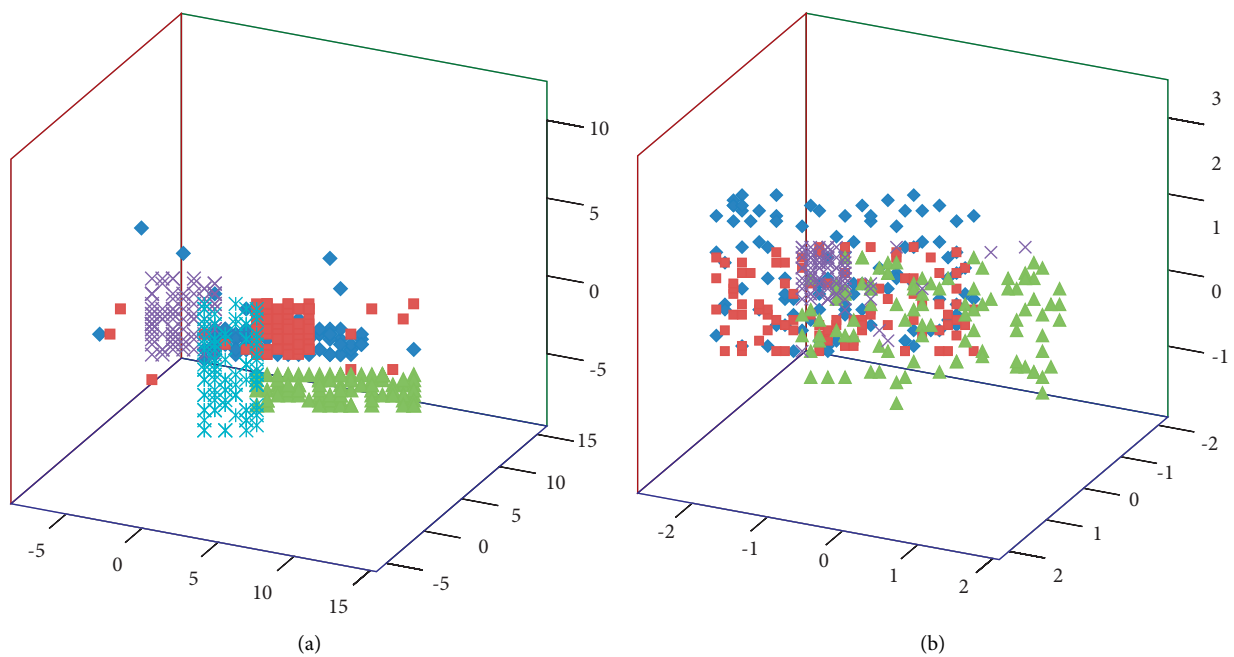


FIGURE 3: Visualization of the output data distribution of the fully connected layer after dimensionality reduction using PCA (a) unused data enhancement and (b) using data enhancement.

enhancement scheme and the data distribution without the enhancement scheme. For the data obtained without a data improvement program, some categories will have very large internal variances. Of course, there are also some internal variances with smaller types. The distance differences between them are also quite different. The model will be because of this phenomenon and improve the ability to recognize some special types of samples. The data-improved scheme has excellent robustness, and the within-class variance of each subgroup is easier to calculate.

4. Research on English Translation Platform Based on Related Technology

4.1. Translation Platform Demand Analysis. Vocabulary 16 to be used in this article is a relatively common application type. The advantage of local dictionary applications is that they can perform local translation functions without connecting to the World Wide Web. Therefore, electronic dictionaries usually implement online translation functions based on part of the functions of local dictionaries. Taking

smartphones as the representative, many smart devices have appropriate restrictions on the response speed, hardware allocation, power capacity, memory size, and other factors due to their portability and mobility requirements [19]. Therefore, users will have a better experience with online translation based on the local translation in the thesaurus. But this will also be limited by resources such as the type and capacity of the lexicon.

The other is an online dictionary based on the implementation of online translation, and the system architecture usually adopts the project C/S model. No matter what kind of client, the server will share part of the workload for the mobile client using this project mode. Usually, according to different operating conditions, such as network transmission conditions and mobile phone hardware configuration, software programs can use mobile computing models to make appropriate plans for task allocation between mobile devices and servers. However, traditional data centers often suffer from congestion during peak periods of application usage, resulting in users not getting corresponding feedback. However, if the number of servers is increased, when the utilization rate of mobile applications is low, most of the servers added by developers in the data center will be useless, which not only leads to a low average utilization rate of servers, but also damages the economic interests of developers. So there will be situations where software developers cannot predict when the next usage rate will reach the peak, nor can they switch on the server's task allocation, causing developers to often suffer losses when the number of visits increases.

The current translation function often requires the input of the target text in the translation box for translation. Many large-language users do not have a big problem with this method, but some small-language users do not have a language library that can be consulted, and do not know how to input the language at all, or their mobile phones do not directly support this language input, which causes a lot of problems and big trouble. For example, when a foreign tourist is interested in the traditional text of another country that is completely unknown to him and wants to know its true meaning, he will find that the traditional query method based on manual input cannot meet the needs of users at all.

Based on the above problems in practical applications, this article proposes a project model of an OCR-based network translation system in a mobile computing environment. The project goals of this system are as follows:

First of all, in order to avoid the inconvenience caused by large-scale manual input to users, the final input method of this system can accept images taken by mobile phone cameras. The user can take a photo of the disputed text with a mobile phone, and the data will be recorded in the form of a photo. The whole process does not require a text control basis. Of course, the user can also input the required query content in the system interface. In this way, you can not only query the photos taken on-site but also take the photos saved on your mobile phone as input content. The source language and target language can be set. The translation results and the text extracted by OCR can be saved in the form of text on the phone. For the content of interest in the translation

result, you can directly call the built-in hyperlink to perform a Google search.

Second, integrate Google framework services on the server side, and implement online translation functions by calling Google Translate API. After all, Google Translate is the fastest and most accurate machine translation available today. Just set your source language and target language, enter the language text you need to translate, and Google will immediately perform a translation search [20]. The translation module currently used by Google combines United Nations documents as the source of the contents of this multilingual library. The translation result has very good accuracy and can support the real-time conversion of hundreds of languages, surpassing all local translation dictionaries.

Finally, considering the limited memory of most mobile phones now, the network translation system designed in this article will be based on the thin client model and try to migrate complex data processing to the backend server. Because the traditional C/S architecture server is inconvenient to maintain and difficult to expand. All the systems will realize the large-scale application of electronic dictionary translation programs through the OpenNebula cloud platform. Based on the OpenNebula cloud platform server, users can significantly improve use efficiency and reduce learning costs.

4.2. Translation Platform Structure Design. With the rapid development of information and digital technology, the application universality of smartphones has surpassed that of representative electronic computing devices such as desktop computers. But in the computing power of electronic devices, the role of mobile phones is still very limited. This is why we want to use clients as little as possible when creating customer mobile services. Most of the data processing is handed over to the server side, and the client side simply stores the edited photos, sends the request, and receives the result. After each translation process, the client will automatically disconnect from the server and will not reconnect until the translation is completed. Since the client does not need to connect to the server 24/7, the mobile phone can also have better battery life. The system structure is shown in Figure 4.

It can be seen from Figure 4 that this system works a lot on the server backend of the OpenNebula platform. In addition to the OCR test program, it also integrates Google's online translation. And because the Internet image engine adopts the OCR function, it may destroy other system tools during execution, so the OCR engine is often located on the server side. In addition, OCR is the starting point of all operations, so we need to have multiple OCRs together to form a processor, and reasonably control the distribution function through the equalizer. Therefore, the server component is divided into three parts: OCR engine, load balancer, and translation processor.

The user of the mobile device establishes a connection with the server through a wireless network. The wireless networking can support 3G, 4G signals or use WIFI. On the

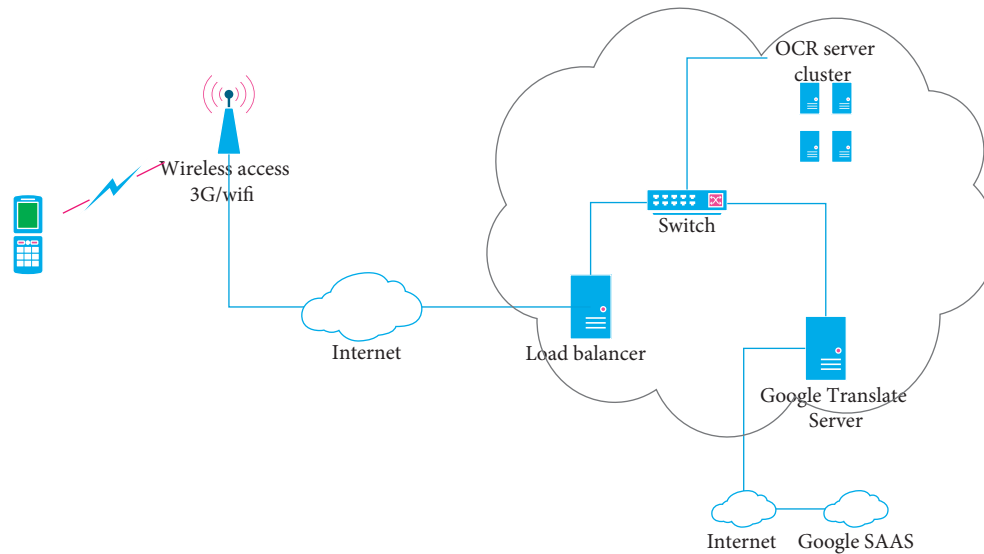


FIGURE 4: The whole system structure diagram.

server side of the OpenNebula cloud platform, there will be a web hosting server, OCR engine exchange server, and translator to form a small local area network for enabling and configuring various server resources. In addition, the server and the conversion server have an Ethernet interface that matches any Ethernet interface that intervenes in the Internet address. The shared server can receive client requests, and the World Wide Web connection can be used for Google translation services and retrieve client translation results. The number of OCR configuration servers will change according to the data fluctuations of their applications, so the configuration of translation servers and serial servers will also change according to the number of functions to be carried out.

The translation server integrates the Google application framework, that is, it can call the Google Translate API to translate the extracted words and sentences into the target language. Google Translate is not only an online translation function. Users can download the Google Translate plug-in and paste the document into the translation page, and the document to be translated will be sent to <https://translate.google.com> for processing. Google also provides a complete human translation team for the Translate API. Users can translate into the program using Google Translate API.

The translation server uses the Google Translate API in conjunction with the Google service framework to identify online translation services. The system puts the user's request in different threads through the Java multi-thread counting machine, and each link request is processed by a separate thread. In this way, when another user issues a request again, the processor can continue to receive other commands without affecting the previous work. The process is shown in Figure 5.

The whole system is based on the C/S architecture and is divided into two parts: server-side and client-side.

The server side is built using the OpenNebula cloud platform, which can provide various tools used in the server

building process and ensure that at least one main network is used to connect the front end. The user end is rooted in the Android platform. Android was born from Google, and there is an Open Handset Alliance (OHA) that is stronger than ever. The design idea of the Android platform is more flexible than the previous platform, and it is more versatile and safer.

4.3. Database Design. This section will explain the names, types, concepts, and descriptions of each part of the translation system. The conversation message table stores conversations and conversation messages, and the specific structure is shown in Table 1.

Every message on the computer must be written in the original language. Although this system is currently only suitable for translation into Chinese and English, the possibility of translation into multiple languages has been considered here, and an interface has been reserved for this, and the translation template in the original language and the result in the target language are stored separately. Message Type is a message type. The receiver has the id information of the message sender, which is used to communicate with each other.

The language table stores the language types supported by the translation system, and the specific structure is shown in Table 2.

The LangName in the table is the key for storing the language name. Because the system can support multiple languages, the final text displayed on the screen layer should not be a fixed character, but can be customized according to the user's language. In iOS, the system provides related configuration files for multi-language conversion modules. By specifying the corresponding key in the code, you can read the corresponding text in the configuration file according to the computer language.

Table 3 shows a framework designed for system expansion.

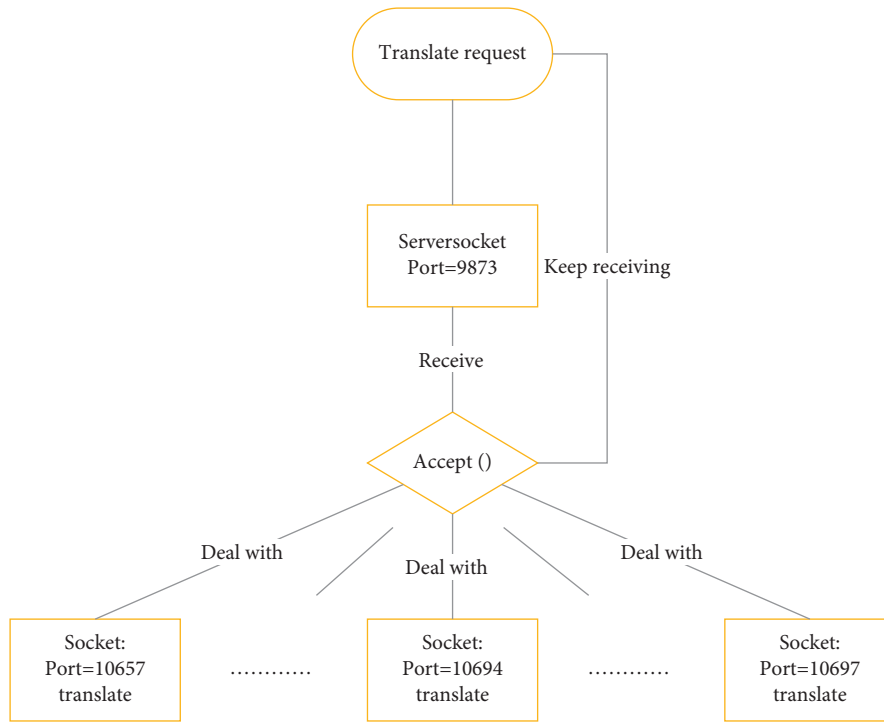


FIGURE 5: The processing flow of the translation server.

TABLE 1: Conversation message table.

Serial number	Field name	Types of	Attributes	Description
1	id	varchar(32)	PRIMARY KEY	Message id
2	sreLang	tinyint	FK, NOT NULL	Source language
3	destLang	tinyint	FK, NOT NULL	Target language
4	sreSentence	varchar(255)	NOT NULL	Source language sentence
5	destSentence	varchar(255)	—	Target language sentence
6	onversationType	tinyint	NOT NULL	Dialogue type
7	messageType	tinyint	NOT NULL	Message type
8	messageTime	varchar(32)	—	Message time
9	Sender	varchar(32)	FK	Message sender id
10	Receiver	varchar(32)	FK	Message recipient id
11	senderName	varchar(32)	-	The name of the message sender

TABLE 2: Language list.

Serial number	Field name	Types of	Attributes	Description
1	id	tinyint	PRIMARY KEY	Language type
2	langName	varchar(32)	—	Language name
3	Description	varcahr(255)	—	Language description

TABLE 3: Language translation table.

Serial number	Field name	Types of	Attributes	Description
1	sreLang	tinyint	PRIMARY KEY	Source language
2	destLang	tinyint	PRIMARY KEY	Target language
3	nterpretable	tinyint	—	Can it be translated

TABLE 4: User information table.

Serial number	Field name	Types of	Attributes	Description
1	id	varchar(32)	PRIMARY KEY	User id
2	Name	varchar(32)	—	User name
3	lang	tinyint	FK	User language
4	Sex	tinyint	—	User gender
5	Region	varchar(32)	—	User's area
6	Phone	varchar(32)	—	User phone
7	Birthday	varchar(32)	—	User birthday
8	AvatarUrl	varchar(255)	—	User avatar url

This framework preserves the relationship between translatable languages and provides support for future multilingual translation.

The user information table stores system user information and the specific structure can be seen in Table 4.

This table is used to store and manage user information, and it records many specific user information.

4.4. Realization of Functional Modules of Translation Platform. In the Android program, the user interface will be configured with View and ViewGroup, and there are multiple types of displays. All display interfaces are subparts of the system vision class. The display object is a data system processing unit used to store specific icon layouts and content attributes on the screen. Service plug-ins are a collection of subparts used to design interactive elements. The plug-ins used in this system are: Image Display, Edittext, and Radio Button. The function of ViewGroup is to load and manage low-level views and other view groups, which can add some structure to the custom UI. When the Activity is activated, the system will instruct the Activity to use the root node to draw them based on the node distance measurement, and each ViewGroup node is responsible for drawing its direct child nodes.

HttpClient is an affiliate program of Apache Jakarta Commons. You can request updates of themes, parameters, and content structure through the HttpClient link. All background services of the system are operated and run based on the cloud virtual machine of the OpenNebula platform. The implementation process of the OCR processor has been shown above, using Tornado written in Python as an online container, through a simple and extensible barrier-free IO server, to achieve timely response communication. Compared with other web servers, Tornado can respond faster and better to the huge influx of traffic. At present, there are many real-time needs of websites that need to be built using Tornado. In the application of this article, an online translation system is established, so the real-time requirements are relatively high, so it is best to use Tornado as a web server.

Load balancing can be used to handle the process of Elastic Load Balancing on the cloud computing platform. In the OpenNebula operating system, we can use Sunstone to view the process and related data of the OCR processing server. If the existing OCR is already in a high load state, the response speed of the system will slow down. Therefore, we

will use it as a virtual machine to help OCR perform data processing, and load balancing will effectively distribute system tasks on these multiple servers, thereby improving the robust performance of existing servers to a certain extent. The elastic load balancing function can observe the status of all current virtual machines and allocate tasks according to the number of idle virtual machines and the total number of tasks, ensuring that each OCR server has the same workload. At certain points in time, when user demand is low and OCR workload is low, a certain number of OCR processing servers can be shut down to ensure server utilization efficiency.

5. Conclusion

First of all, this article introduces the current development status and industry prospects of embedded systems at home and abroad. On this basis, it analyzes the function and feasibility of embedded computers as a test platform and proposes a configuration scheme for a general test platform. And use the platform designed in this paper to introduce deep learning algorithms into the recognition of environmental sounds as research materials, and conduct in-depth research and exploration on the basic construction of deep learning networks and data optimization schemes. At the end of the article, this article takes the popular mainstream OCR processor as an example, introduces the operating system based on OpenNebula and elaborates the process of creating an OCR-server virtual machine to assist the OCR server in data optimization processing, and then respectively introduces Load balancer LVS-reserve-server and Google Translate API. And give a key explanation on their technical background, implementation methods, impact on the client, and operation process. Finally, it describes in detail the two main functional modules that the mobile client based on the Android operating system relies on, and implements the language conversion operation of the English translation platform.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by A Study on the Reconstruction of College English Teaching Model in the “Internet+” era, a General Project of the National Social Science Foundation (No.: 17BYY102).

References

- [1] A. Martini and J. Bosch, “Architectural technical debt in embedded systems,” *IncoSE International Symposium*, vol. 26, no. 1, pp. 1029–1043, 2016.
- [2] K. Wang, M. Du, D. Yang, C. Zhu, J. Shen, and Y. Zhang, “Game-theory-based active defense for intrusion detection in cyber-physical embedded systems,” *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 1, pp. 1–21, 2016.
- [3] A. P. Rodrigues, B. Santos, P. F. Carvalho et al., “Taking advantage of the intercommunication features of IPMCs in ATCA CDAQ systems,” *Fusion Engineering and Design*, vol. 128, pp. 138–142, 2018.
- [4] J. Yang and R. K. Das, “Long-term high-frequency features for synthetic speech detection,” *Digital Signal Processing*, vol. 97, Article ID 102622, 2020.
- [5] Z. Wu, X. Xiao, E. S. Chng, and H. Li, “Synthetic speech detection using temporal modulation feature,” in *Proceedings of the IEEE International Conference On Acoustics, Speech And Signal Processing*, pp. 7234–7238, IEEE, Canada, CA, USA, March 2013.
- [6] Z. H. Wu, R. K. Das, J. Yang, and H. Li, “Light convolutional neural network with feature genuinization for detection of synthetic speech attacks,” in *Proceedings of the Interspeech 2020*, Shanghai, China, October 2020.
- [7] J. Sanchez, I. Saratxaga, I. Hernaez, E. Navas, and D. Erro, “A cross-vocoder study of speaker-independent synthetic speech detection using phase information,” in *Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association*, Singapore, September 2014.
- [8] Y. Zhang, F. Jiang, and Z. Duan, “One-class learning towards synthetic voice spoofing detection,” *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.
- [9] N. Chatterjee and S. Gupta, “Efficient phrase table pruning for Hindi to English machine translation through syntactic and marker-based filtering and hybrid similarity measurement,” *Natural Language Engineering*, vol. 25, no. 1, pp. 171–210, 2019.
- [10] S. Feng, “The acquisition of English definite noun phrases by Mandarin Chinese speakers,” *Studies in Second Language Acquisition*, vol. 41, no. 04, pp. 881–896, 2019.
- [11] G. F. Peng, H. T. Wu, J. Y. Xu, and J. S. Chang, “Mining and clustering phrases for English for special purpose: travel writing,” in *Proceedings of the 2020 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 97–101, IEEE, Taipei, Taiwan, December, 2020.
- [12] M. Schoeberl, A. E. Dalsgaard, R. R. Hansen et al., “Safety-critical Java for embedded systems,” *Concurrency and Computation: Practice and Experience*, vol. 29, no. 22, p. 3963, 2017.
- [13] M. Adil, H. Alshahrani, A. Rajab, A. Shaikh, H. Song, and A. Farouk, “QoS review: smart sensing in wake of COVID-19, current trends and specifications with future research directions,” *IEEE Sensors Journal*, vol. 2022, Article ID 3170055, 1 page, 2022.
- [14] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: a survey,” *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.
- [15] H.-q. Gao and Y.-j. Zhai, “System design of cloud computing based on mobile learning,” in *Proceedings of the 2010 Third International Symposium on Knowledge Acquisition and Modeling*, pp. 239–242, IEEE, Wuhan, China, October 2010.
- [16] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: a survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.
- [17] J. Yang, R. K. Das, and H. Li, “Significance of subband features for synthetic speech detection,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2160–2170, 2020.
- [18] D. Paul, M. Pal, and G. Saha, “Spectral features for synthetic speech detection,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 605–617, 2017.
- [19] R. Joyce and N. Audsley, “Exploring storage bottlenecks in Linux-based embedded systems,” *ACM SIGBED Review*, vol. 13, no. 1, pp. 54–59, 2016.
- [20] Z. Wang, Z. Zhou, H. Zhang, G. Zhang, H. Ding, and A. Farouk, “AI-based cloud-edge-device collaboration in 6G space-air-ground integrated power IoT,” *IEEE Wireless Communications*, vol. 29, no. 1, pp. 16–23, 2022.