A Storage Method of Online Educational Resources for College Courses Based on Artificial Intelligence Technology

Mingjie Zheng[1(✉)] and Xu Wang[2]

[1] College of Humanities and Information, Changchun University of Technology, Changchun 130122, China
Zhengmingj541@163.com
[2] Tianjin Maritime College, Tianjin 300150, China

Abstract. In the Internet era, the scale of online education resources small files of college courses in online learning is becoming larger and larger, and the workload is large. The traditional storage method meets the storage requirements of massive education resource small files, and designs a storage method of online education resource of college courses based on artificial intelligence technology. Through artificial intelligence technology to recognize the characters and voice in online education resources of college courses, complete the recognition of online education resources of college courses in the platform. Gzip compression algorithm is used to compress recognition results. Design the distributed cloud storage architecture of online education resources to realize the distributed storage of compressed data resources. The test results showed that this method took less memory and took less time to read small files randomly, and has high reliability.

Keywords: Artificial Intelligence Technology · YOLOv3 · Online Education Resources · Gzip Compression Algorithm · Resource Storage

1 Introduction

With the rapid development of information technology, people's production, life and working methods are changing. The world we live in is becoming increasingly connected, interconnected and intelligent. The educational objectives and teaching models in the context of the traditional industrial era are no longer suitable for the needs of today's information and big data era. The concept and model of "smart education" came into being [1]. Various institutions and researchers have put forward many new innovations for the realization of intelligent education, such as the application practice form of online classroom and flipped classroom, changing the previous teaching mode, so that learners can obtain more independent learning time and corresponding learning resources, and make the learning process more efficient.

In recent years, with the introduction and gradual popularization of new education models, e-Learning and MOOC (Massive Open Online Courses), more and more learners

L. Yun et al. (Eds.): ADHIP 2023, LNICST 547, pp. 122–138, 2024.
https://doi.org/10.1007/978-3-031-50543-0_9

have the opportunity to learn on online platform. Accordingly, the platform has accumulated a large amount of teaching behavior data and knowledge resources, providing a good foundation for the update and improvement of the platform itself. However, the continuous growth of the data volume of online education resources for college courses has brought difficulties with its storage. At present, there are the following problems in the storage of online education resources for college courses:

(1) Capacity demand: the platform has established a file server to store digital data uniformly, but it still cannot meet the storage capacity demand of rapid resource growth;
(2) Access performance problem: the file server read and write operations are limited to the IO speed of a single server. With the increase of file data, the performance of file location and reading also increases and decreases significantly;
(3) The security of a single file server is relatively poor. It is very likely that the server storage device will be damaged and cannot be recovered. There is a lack of good backup mechanism. There are hidden dangers in data security, and it does not have the performance requirements of high reliability and high security;
(4) Sharing problem: the traditional silo storage mode and application development mode have caused the lack of unified planning and management of the current education system, and the scattered information resources have caused information islands, which cannot be effectively shared;
(5) Application requirements: The file server can only simply provide storage space, which cannot meet the actual needs of users. Its functions do not match the actual needs, and it is inconvenient for teachers and students to use. It is facing difficulties in resource retrieval, grouping, sharing, and so on.

Based on this background, this paper studies the storage of online educational resources for college courses. For the research on the storage of educational resources, many research results have been achieved. Reference [2] proposed an online learning resource compression storage method based on edge computing. In order to improve the compressed storage efficiency of online learning resources and reduce the occupation of cloud storage space, this method uses distributed compression sensing as the edge computing method. It collects online learning resources by constructing the first joint sparse model JSM-1 compression, reconstructs resources by combining the joint reconstruction algorithm composed of synchronous orthogonal matching pursuit algorithm and dictionary learning algorithm, obtains sparse dictionary atoms and measured values, and transmits them to the cloud server to establish a complete cloud sparse dictionary, realize compressed storage of online learning resources. Experiments show that the larger the sparse dictionary and the more training and learning samples are, the higher the SNR of resource compressed storage is, and the better the resource processing performance is. Using this method to compress online learning resources, the picture has a higher definition, and the compression time is shorter, which can save the compression time and reduce the storage space. Reference [3] building a digital resource library for higher education English through cloud platforms. The cloud platform enables the integration, efficiency, scalability, and interactivity of English resource libraries, while vocational colleges achieve the co construction and sharing of multi platform resources.

The construction of the English resource database in vocational colleges cannot be separated from the collaboration between institutions and enterprises. The requirements for various aspects of the vocational English digital resource database system are provided from three aspects: functional requirements, performance requirements, and operational requirements. The overall technical architecture and network topology of the education cloud platform where the vocational English digital resource database system is located are established to ensure the security, scalability, and operability of the platform, to ensure the collection and storage of resources. Cloud computing technology can integrate and manage various resources through technologies such as networking, virtualization, and distributed storage. Based on the proposed optimization strategy, the core module of the vocational English digital resource library system was designed, and some important interfaces and source code were displayed. However, the storage efficiency of this method's digital resource library is relatively low.

The problem is to design a storage method of online educational resources for college courses based on artificial intelligence technology.

2 Design of Online Educational Resources Storage Method for College Courses

The storage method for online education resources of university courses is mainly divided into three parts: identifying characters and speech in the online education resources of university courses, compressing the recognized characters and speech using Gzip compression algorithm, and distributed storage of compressed data resources. The storage structure diagram of online education resources for university courses is shown in Fig. 1.

2.1 Character Recognition and Speech Recognition

Through artificial intelligence technology to recognize the characters and voices in online education resources of college courses, complete the recognition of online education resources of college courses on the platform. Character recognition is divided into two steps: first, recognize the text part in the picture content and implement it into horizontal and vertical coordinates, which is called target detection; Then character recognition and extraction are carried out for the image content within the coordinate range. These two parts need different neural network models to complete.

The target detection algorithm for character recognition uses open-source YOLO. In the prediction process of YOLO, the target detection problem is first transformed into a linear regression problem in a high-dimensional space. First, the size of the graph to be detected is proportionally scaled and adjusted, and then the adjusted graph is sent into the model as the input of the convolutional neural network for recognition, finally, the output of the model is processed to get the detected target. The advantage of this is that the object center and object type will be judged only once for each grid area in the image, and the prediction analysis will not be repeated for a certain area when judging multiple possible results in the prediction process, which greatly improves the prediction efficiency.

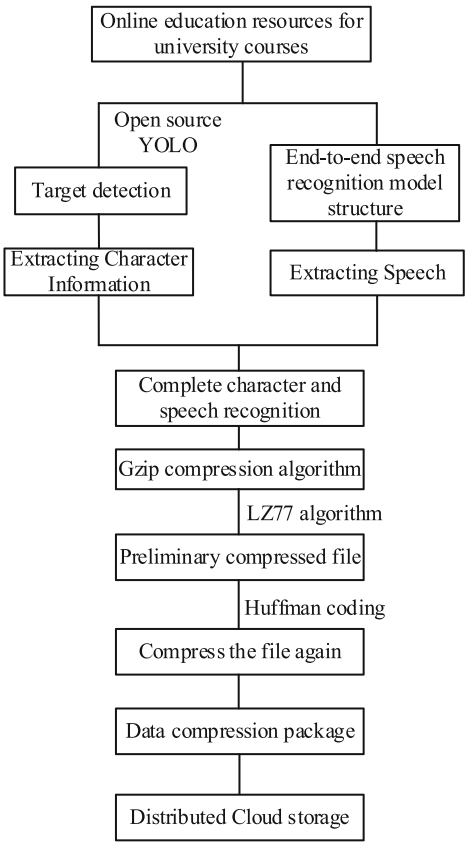

Fig. 1. Storage Structure of Online Education Resources for University Courses

YOLO's basic network model is GoogLeNet, and it has made corresponding improvements on its basis, giving up the inception. Instead, it alternately uses square convolution cores with compensation ranging from 1 to 3. A total of 24 convolutions are required, including fixed pooling steps and normalization processing, followed by two layers of fully connected neural networks. In the whole process mentioned above, the function of convolutional neural network is to extract features from the image, convert the features into one-dimensional data through normalization, input the full connection neural network, and then predict the coordinates and types of objects. This project uses an upgraded version of YOLO YOLOv3. Table 1 shows the neural network structure of YOLOv3.

Table 1. YOLOv3 Neural Network Structure

type	filters	size	output
convolutional	32	3X3	256 X 256
convolutional	64	3X3/2	128X 128
convolutional	32	1X1	64X 64
convolutional	128	3X3	64X 64
residual	64	3X3/2	32X32
convolutional	128	1X 1	32X 32
convolutional	256	3X3	16X 16
convolutional	128	3X3/2	16X 16
residual	256	1X1	8X8
convolutional	512	3X3	8X8
convolutional	256	3X3/2	64X 64
convolutional	512	3X3	256 X 256
residual	1 024	3X3/2	128X 128
convolutional	512	1X 1	64X 64
convolutional	I 024	3X3	64X 64
convolutional	I 024	global	32X32
residual	32	T000	32X 32
convolutional	64	1X 1	16X 16
convolutional	32	3X3	16X 16
convolutional	128	3X3/2	8X8
residual	64	1X1	8X8
avgpool	128	3X3	16X 16
connected	256	3X3/2	256 X 256
softmax	128	1X 1	128X 128

Here YOLO uses a convolutional neural network with a depth of 53 layers, in which the adjacent convolutional layers will form a convolutional residual network to resist the problem of gradient disappearance that may occur with the gradual deepening of the network layer. YOLOv3 also absorbs the idea of FPN121, and considers the characteristics of different sizes when performing target detection, which improves the accuracy of target detection in images, it is more sensitive to the content with smaller size on the image.

After the target detection algorithm extracts the text box in the image, the character recognition algorithm is required to extract the text information from the image content [4]. CRNN, a neural network structure, is used to extract the features of the text content in the image and make it one-dimensional, and then CTC is used to predict the text content

sequence from the one-dimensional results output by CRNN. Since the task of character recognition is to extract the character information from the two-dimensional image, this task is suitable for both feature extraction using the CNN convolutional neural network model commonly used in computer vision and semantic association analysis using the RNN recurrent neural network model commonly used in natural language processing, so a CRNN neural network structure combining the two is designed.

The CNN structure of CRNN adopts the VGGI27 structure, and makes some fine adjustments to the VGG network. Because the final training results of the CNN part need to be imported to the RNN part, the CNN part must be output in one dimension, so the pool size of the last two largest pool layers of VGG is changed from 2 * 2 to 2 * 1 to increase the width of the feature map. The adjusted CNN structure neural network structure is shown in Table 2.

Table 2. CNN Structure of CRNN

Serial Number	Type	Configurations
1	Transcription	–
2	Bidirectional-LSTM	#hidden units:256
3	Bidirectional-LSTM	#hidden units:256
4	Map-to Sequence	–
5	Convolution	#maps:512, k:2 X 2, s:1, p:0
6	MaxPooling	Window:1 X 2, s:2
7	BatchNormalization	–
8	Convolution	#maps:512, k:3 X 3, s:1, p:1
9	BatchNormalization	–
10	Convolution	#maps:512, k:3 X 3,s:1, p:1
11	MaxPooling	Window:1 X 2, s:2
12	Convolution	#maps:256, k:3x 3, s:1, p:1
13	Convolution	#maps:256, k:3 X 3, s:1, p:1
14	MaxPooling	Window:2 X 1, s:2
15	Convolution	#maps:128, k:3 X 3, s:1,p:1
16	MaxPooling	Window:2 X 1, s:2
17	Convolution	#maps:64, k:3 X 3, s:1, p:1
18	Input	W X 32 Gy-scale image

After the CNN part is completed, the feature map is one-dimensional and transmitted to the RNN part. In order to prevent gradient dispersion in the RNN part during training, CRNN uses LSTM, a deformed RNN structure.

A training set containing 34W pieces of training data was made using all the vocabulary sets in the open source word segmentation library on Github. Each training data is

a picture, and the content on each picture is a sentence with a length of 2–14 characters, including Chinese characters, Arabic numerals and Chinese punctuation marks. Draw all the words and sentences contained in the stutter on the complex background, and change the size and angle of each character, so as to approach the artistic font recognition in the real use scene; On this basis, about 15W labeled real scene data sets have been added, forming a total of 50W training data sets. Finally, CRNN model is used to train on this data set, and the final model obtained has an accuracy rate of 99.8% on the training set and 88% on the reserved real scene test set, which meets the expected requirements [5].

A TCN Transformer CTC end-to-end speech recognition model is designed, and the model structure is shown in Fig. 2. It consists of pre-processing module (acoustic pre-processing module, text pre-processing module), decoder, and mixed CTC/attention loss. It regards ASR as a sequence to sequence task. The encoder maps the input frame level acoustic characteristics (Formula (1)) to a sequence high-level representation (Formula (2)). The decoder decodes the generated text (Formula (3)) and the attention adjusted hidden state (Formula (2))β_l and finally generate the target transcriptional sequence (Formula (4)).

$$\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_T) \tag{1}$$

In formula (1)α_T refers to the no T frame level acoustic characteristics for input.

$$\chi=(\chi_1, \chi_2, \ldots, \chi_N) \tag{2}$$

In formula (2)χ_N refers to the no N advanced representation of series.

$$\beta=(\beta_1, \beta_2, \ldots, \beta_{l-1}) \tag{3}$$

In formula (3)β_{l-1} refers to the no $l-1$ generated text.

$$\beta=(\beta_1, \beta_2, \ldots, \beta_L) \tag{4}$$

In formula (4)β_L refers to the no L target transcripts.

The pre-processing module is divided into acoustic pre-processing module and text pre-processing module. K 2-D convolution modules are used in the acoustic front-end module. Each convolution module contains a 2-D convolution layer and a ReLU activation layer [6]. Finally, we use Positional Encoding to obtain the absolute position information of acoustic features. In the text pre module, J TCN modules are used to learn the implicit positional relationship.

Speech recognition can be regarded as a timing problem. TCN is a new model structure based on convolution, which can deal with timing problems. Compared with using RNN structure, TCN has reached the level of RNN model in a variety of scene tasks. TCN can be divided into causal convolution and dilation convolution, of which causal convolution has strong temporal constraints, and dilation convolution can obtain a large receptive field through the setting of dilation factors. Using the TCN structure in the text front model can maintain the original good parallelism of the Transformer. In addition, it has the advantages of making the gradient more stable and occupying less memory than other convolution or RNN based structures.

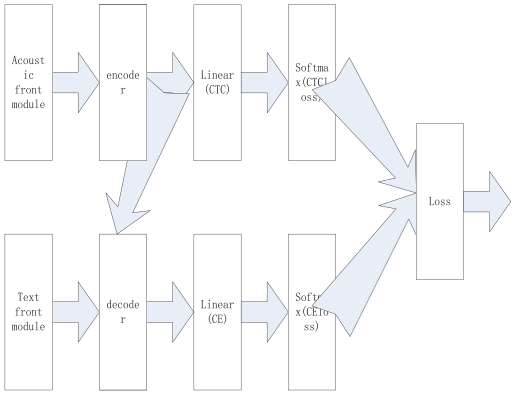

Fig. 2. End to end speech recognition model structure

The encoder and decoder are composed of several identical module stacks. Each module has two main sub layer structures, namely, the Multi Head Attention layer and the Feed Forward layer. After each sub layer, residual connection and layer normalization are used. The difference between the decoder and the encoder is that it uses a multi head attention mechanism to mask future information, so that future tag information cannot be seen during decoding, and the second Multi Head Attention layer uses cross attention. Different from the Transformer model in Linhao Dong et al., firstly, this paper adjusts the structure of the decoder. In the encoder part, the parallel TCN structure is used, which is used to fuse with the features processed by the Multi Head Attention layer, extract more features and slow down the disappearance of position information; Secondly, the encoder output part will also be input into the CTC structure to speed up the model training convergence speed and improve robustness.

In many mainstream model work, CTC target function is used as an auxiliary task, which can further improve the model effect. Because different from the attention model, CTC's forward backward algorithm can force monotonic alignment between voice and tag sequences, make up for the lack of attention alignment mechanism, and make the model more robust in noisy external environments. TCN Transformer CTC model combines the advantages of CTC and attention, so the total loss function is defined as the weighted sum of CTC loss and attention loss:

$$H_{\text{loss}}=\sigma CTC_{\text{loss}}+(1-\sigma)ATT_{\text{loss}} \tag{5}$$

Parameters in formula (5)σ is used to measure the importance of CTC loss and attention loss.

2.2 Compression of Recognition Results

Gzip compression algorithm is used to compress the recognition results. Gzip takes the Deflate algorithm as its core work. The Deflate algorithm mainly combines two algorithms to work together to complete compression and decompression. One is LZ77 algorithm, and the other is Hoffman algorithm. When Gzip uses the Deflate algorithm

to work, it first reads the external incoming file stream. LZ77 algorithm first performs a preliminary compression of the file, and then transmits the compression result to the Hoffman algorithm. The result is compressed again through Hoffman coding to obtain the final data compression package.

LZ77 algorithm is an efficient lossless compression algorithm based on a sliding window mechanism similar to TCP network to solve network congestion and ensure reliable transmission. Its core principle is to use this sliding window mechanism to buffer encoded data in a sliding window of a set size. Then, when processing unencoded data, new characters will be used to match the encoded characters in the buffer, and the longest string will be found and recorded. The method of recording the matched maximum string is a triple, and its format is:

$$\omega=(\xi\ \psi\ \zeta) \tag{6}$$

In formula (6)ξ indicates the string offset; ψ indicates the length of repeated characters; ζ represents a new character that is not encoded.

In a triple, the string offset is the distance between the previous repeated character and the character being processed, and its size cannot exceed the size of the window. The length of the repeating character is the maximum matching length between the character in the window and the character being processed. An uncoded new character is the first character that is not repeated. If new characters cannot be matched from the window, output (0,0, new characters).

For example, if the string abcdbcddadcde is compressed, the compression process of LZ77 is as follows: when encoding the first character in the string, the character a is used to match the string in the sliding window. Since there is no character in the window at this time, the first character is encoded as (0,0, a). Then the second character is encoded. We find that the first four characters are all like this, so the offset and length are both 0. When the fifth character is encoded, the repeated string in the window is bcd, so the encoding offset starts from b, is 1, the repeated character is 3, and the next character is a, so the encoding triplet is (1,3, a). And so on until the last character. The detailed coding of each step is shown in Table 3.

Table 3. Detailed coding of each step

Compressed characters	Character to be processed	code
-	abcdbcdadcde	(0,0,a)
a	bcdbcdadcde	(0,0,b)
ab	cdbcdadcde	(0,0,c)
abc	dbcdadcde	(0,0,d)
abcd	bcdadcde	(1,3,a)
abcdbcda	dcde	(3,3,e)

It can be seen from the compression process in Table 3 that when the position of the same string in the whole string to be compressed is far away, that is, when the

same string is sparse, each character must be encoded with a triplet. At this time, the compression efficiency of the algorithm will be relatively low, and even in extreme cases, the compressed data may be larger than before compression [7].

The decompression of each step is shown in Table 4.

Table 4. Decompression of each step

Read in code	Decoding processing	Decoded Data
(0,0,a)	From position 0, insert 0 decoded characters, and insert a new character	a
(0,0,b)	Starting from position 0, insert 0 decoded characters, and insert new characters as b	ab
(0,0,c)	From position 0, insert 0 decoded characters, and insert new characters as c	abc
(0,0,d)	Starting from position 0, insert 0 decoded characters, and insert new characters as d	abcd
(1,3,a)	From position 1, insert 3 decoded characters, and insert a new character	abcdbcda
(3,3,e)	Starting from position 3, insert 3 decoded characters, and insert new characters as e	abcdbcdadbce

There is a stack of data that has not yet been compressed. The data items in the data have different repetition frequencies. To distinguish different data items, a single code is allocated for each data item. The principle of allocation is that the code corresponding to the higher frequency is shorter, and on the contrary, it is longer. The above overview is the key implementation logic of Huffman coding. It can achieve that the average value of all data items is always in the shortest state. It is worth noting that the Huffman code can be created by scanning the data that has not yet been encoded twice. The two functions are respectively: sorting out the reproduction frequency of each data that has not yet been encoded and the actual coding work.

The implementation process of Hoffman algorithm is divided into two steps: building Hoffman tree and generating Hoffman code. First, we need to count the frequency of each character in the data worm, and use them as leaf nodes to build a binary tree. Then, by combining the two nodes with the lowest frequency, a complete Hoffman tree is continuously constructed. When generating Hoffman code, start from the root node to traverse the Hoffman tree. When encountering the left subtree, add 0 to the code, and when encountering the right subtree, add 1 to the code, until the leaf node is traversed, the Hoffman code of the character can be obtained.

2.3 Compressed Data Resource Distributed Storage

Design a distributed cloud storage architecture for online education resources to realize distributed storage of compressed data resources. The architecture design includes logical architecture design and physical architecture design. Logical architecture focuses on

describing the relationship between different levels, while physical architecture focuses on describing the layout and composition of hardware devices. In the design of this architecture, we mainly focus on the design of distributed storage, that is, the design and use of distributed file architecture. This architecture is implemented in the resource layer of the logical architecture, which is mainly to organize and uniformly manage different storage resources (servers) under the physical architecture.

From the perspective of logical architecture, the distributed cloud storage architecture can be divided into four layers. The lowest layer is the resource layer, which corresponds to the physical resources to provide the most basic underlying distributed storage architecture. Then the upper layer is the service layer, which provides the user with a file read/write function interface. Above this layer is the application layer. Developers can use the lower layer service interface to improve management functions, here are the functional modules of the distributed cloud storage architecture. The top layer is corresponding to different access systems, including support for PCs, tablets, mobile phones, etc. The design of the logical architecture is shown in Fig. 3.

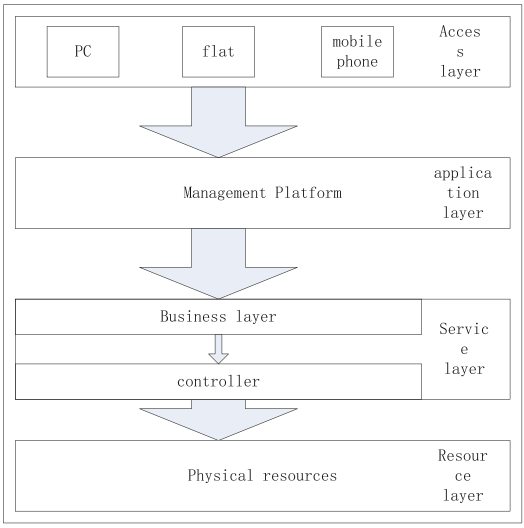

Fig. 3. Design of logical architecture

(1) Resource layer: it is the hardware foundation of the whole platform, including many servers/storage and other infrastructures. These resources are distributed in different places, including the existing data network data center room/data center hosted by a third party, and the computer room for disaster recovery. These resources are organized and managed using distributed file system and database cluster, and become the basic resource provider of distributed storage [8]. We use MFS open source architecture at this level.
(2) Service layer: carry out permission control, and provide standardized encapsulation of resources, application interfaces, etc., persistent storage of data, format conversion between different form data and background database, unified operation of

unstructured data and structured data, and users' online multi module asynchronous operation, etc., for upper layer applications, as well as third-party developers [9].

(3) Application layer: It is a cloud storage business system, mainly for teachers and students and other users to achieve resource management business operations. It is mainly responsible for the realization of business logic, modular division (business and general modules), and the realization of simultaneous online operation of the system by different users, that is, the functional requirements in the architecture requirements.

(4) Access layer: allows users to access through computer web pages, tablets, mobile phones, etc.

In terms of physical architecture, the distributed cloud storage architecture consists of multiple types of physical servers with different roles, and the composition relationship between them needs to be described. The distributed file system working with multiple server clusters is used as the underlying support, as shown in the following formula:

$$\psi = \begin{pmatrix} \text{MasterServer, Chunkserver,} \\ \text{Client, } \ldots \text{, Metalogger} \end{pmatrix} \tag{7}$$

In Formula (7)MasterServer refers to the metadata server; Chunkserver refers to data storage services; Client means the Customer; Metalogger refers to the metadata log server.

The upper layer sets up an application server to carry the business system and accept user requirements.

Each server has installed Centos 5.8bit, where:

(1) Application server: The application server uses open source Tomcat, which is the most popular free J2EE container at present; Deploy JBoss and system support services, including message queue, bus and other middleware services and JVM environment;

(2) Control node master: the cloud storage system adopts distributed storage architecture at the bottom, and uses MFS architecture for secondary development. The control node is the master node of the metadata server of the distributed file system;

(3) Data node chunkserver: MFS distributes data on different machines, called data nodes, and establishes multiple replicas to ensure reliability;

(4) Database MySQL node: MySQL database is used as the database of the business system, and the main language is the structured query language. Access the database through JDBC to realize the interaction between the database and the application. Read write separation configuration is used to improve the application access performance.

The data distribution of the distributed cloud storage architecture is implemented as follows: first, the file data is divided into 64 MB chunks, as shown in the following formula:

$$\upsilon = \left(CHUNK_1^{64}, CHUNK_2^{64}, \ldots, CHUNK_t^{64} \right) \tag{8}$$

Then they are stored on multiple chunkserver nodes, and there is a certain algorithm to select chunkserver during storage, so as to prevent uneven load distribution of data blocks between chunkservers of the system[10].

3 Experimental Test

3.1 Server Configuration

For the designed storage method of online education resources for college courses based on artificial intelligence technology, in order to test its performance, configure the server as shown in Table 5.

Table 5. Configured Servers

S/N	role	system	IP
1	MFS Master、keepalived Master	CentOS6.2 64bit	192.182.10.101
2	MySQL	CentOS6.2 64bit	192.182.10.102
3	MySQL	CentOS6.2 64bit	192.182.10.103
4	MFS Metalogger、keepalived Backu	CentOS6.2 64bit	192.182.10.104
5	MFS Chunkserver	CentOS6.2 64bit	192.182.10.105
6	MFS Chunkserve	CentOS6.2 64bit	192.182.10.106
7	APP Server、MFS Client	CentOS6.2 64bit	192.182.10.107
8	Virtual IP, configured by 192.182.10.101 and 192.182.10.104 respectively to provide services	-	192.182.10.108

The master is the core of MFS and needs to be installed on a server with high stability and high configuration. Large memory, large hard disk, and high CPU requirements. A machine is used as both the metalogger and the backup master server, so that machine should use the same configuration as the master.The installation configuration keeps alive: 192.182.10.101 and 192.182.10.14. Where 101 is the master and 104 is the backup. Monitor the master server through keepalived. When the mfsmaster service on the master server 192.182.10.101 has a problem, it will automatically switch to the backup server 192.182.10.104.

3.2 Test Items

The purpose of performance testing is to check whether the performance requirements are met. The test items are as follows:

(1) Memory usage

To analyze the memory usage of NameNode by different numbers of files, upload 2000, 4000, and 10000 small files respectively, and observe and record the memory usage.

(2) File reading experiment

Upload 2000, 4000 and 10000 small files of educational resources respectively. In the test, the online learning resources compression storage method based on edge computing (Reference [2] method) and a digital resource library of english for higher education based on a cloud platform (Reference [3] method) are used as the comparison method, which is represented by Method 1 and Method 2 respectively.The random reading operation experiment and sequential reading experiment of small files were conducted respectively, and the corresponding average value was obtained as the experimental data after several experiments.

(3) Use three methods to upload five types of files: text files, image files, video files, audio files, and compressed files, and compare the situation of uploading files using different methods.

3.3 Test Results

The test results of memory usage are shown in Fig. 4.

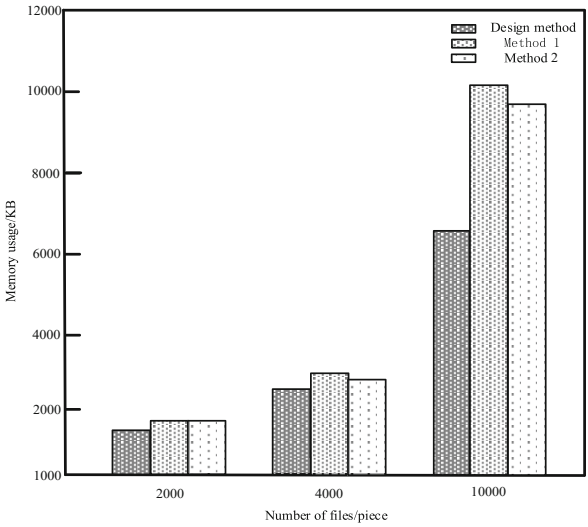

Fig. 4. Memory Usage

As can be seen from Fig. 4, the memory occupation of file name nodes of different orders of magnitude is different. When the number of files is small, the occupation of design methods is not much different from that of comparison methods. When the number of files is 10000, the memory usage of Method 1 and Method 2 is 10100KB and 9800KB, respectively. The memory usage of the design method is only 6700KB, indicating that the design method can reduce memory usage and has good stability.

In the file reading experiment, the random access experiment results of small files are shown in Fig. 5.

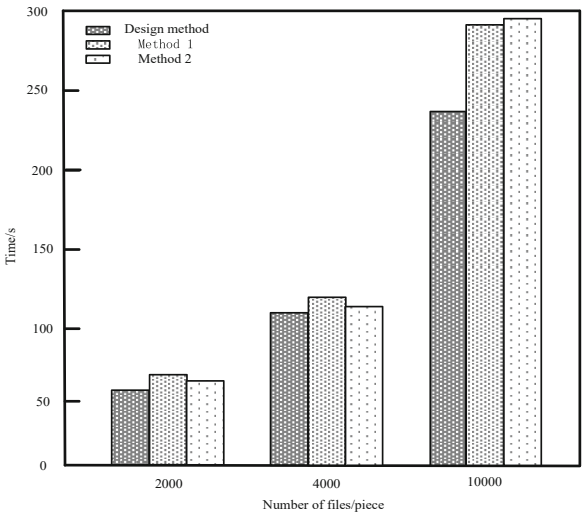

Fig. 5. Comparison of Random Reading Time of Small Files

From Fig. 5, it can be seen that as the number of small files increases, the random read time of small files using different methods also increases. When the number of small files is 10000, the random read times for Method 1 and Method 2 are 285s and 290s, respectively, while the design method only takes 240s, indicating that the design method is more efficient in randomly reading small files.

The file upload situation of different methods is shown in Table 6.

From Table 6, it can be seen that uploading image files using Method 1 failed, while uploading video files and compressed files using Method 2 failed. The design method can successfully perform upload operations for all file types, indicating its high reliability.

Table 6. File Upload Status by Different Methods

File type/type	Design method	Method 1	Method 2
Text file	Success	Success	Success
Image file	Success	Fail	Success
Video file	Success	Success	Fail
Audio files	Success	Success	Success
Compressed file	Success	Success	Fail

4 Conclusion

With the popularization and wide use of the Internet, online learning is becoming more and more popular, and online education resources of college courses in the network are becoming more and more abundant, especially the learning resources in the form of text. Design a storage method of online education resources of college courses based on artificial intelligence technology. Using artificial intelligence technology to recognize characters and speech in online educational resources, the text pre model in speech recognition uses TCN structure to maintain good parallelism of the Transformer and reduce memory usage. The Gzip compression algorithm is used to compress the identification results to design the distributed Cloud storage architecture of online education resources, realize the distributed storage of online education resources of college courses. The storage method of online educational resources for university courses based on artificial intelligence technology has great potential in the future, not only providing personalized, interactive, and efficient learning experiences for students, but also helping teachers better conduct teaching. With the development of artificial intelligence technology, the recognition process of characters and speech in online education resources will be improved in the future to achieve better storage effects.

References

1. Bowker, L.: Translating for Canada, eh?: Developing open educational resources to support localization into Canadian English and French. J. Int. Local. **8**(2), 156–164 (2021)
2. Yuan, L.: Research on compressed storage method of online learning resources based on edge computing. J. Ningxia Normal Univ. **43**(01), 76–83 (2022)
3. Wang, J., Li, W.: The construction of a digital resource library of English for higher education based on a cloud platform. Sci. Program. (Pt.10), 4591780.1–4591780.12 (2021)
4. Tomas, V., Solomon, P., Hamilton, J., et al.: Engaging clinicians and graduate students in the design and evaluation of educational resources about universal design for learning. Can. J. Speech-Lang. Pathol. Audiol. **45**(1), 59–75 (2021)
5. Barkhatova, D.A., Simonova, A.L., Lomasko, P.S., et al.: Features of "inverted" educational resources for distance learning of pupils. Open Educ. **25**(4), 4–12 (2021)
6. Hakim, A.: Study of open educational resources: a survey based approach. Acad. Int. Multidisc. Res. J. **11**(2), 1546–1553 (2021)

7. Hongmin, S., Gongjian, Z.: Research on a new algorithm of erasure correcting code for distributed storage data based on decision tree model. Comput. Integr. Manuf. Syst. **39**(6), 473–477 (2022)
8. Khan, N.A., Shafi, S.M.: Open educational resources repositories: current status and emerging trends. Int. J. Dig. Liter. Dig. Comp. **12**(1), 30–44 (2021)
9. Borzutzky, C.: Adolescent medicine and pediatric residency training: the value of collaboration and shared educational resources. The Journal of Adolescent Health: Official Publication of the Society for Adolescent Medicine **68**(5), 842–843 (2021)
10. Camel, V., Maillard, M.N., Descharles, N., et al.: Open digital educational resources for self-training chemistry lab safety rules. J. Chem. Educ. **98**(1), 208–217 (2021)