



# A Video Parallel Retrieval Method Based on Deep Hash

Jiayi Li<sup>1</sup>, Lulu Bei<sup>2</sup>(✉), Dan Li<sup>2</sup>, Ping Cui<sup>2</sup>, and Kai Huang<sup>3</sup>

- <sup>1</sup> Transportation, Shenyang Aerospace University, Shenyang 110136, China  
<sup>2</sup> School of Information and Electrical Engineering, Xuzhou Institute of Technology, Xuzhou, China  
<sup>3</sup> JiangSu XCMG Information Technology Co., LTD., Xuzhou 221008, China

**Abstract.** This paper designs a parallel video retrieval based on Spark and deep hash. The method comprises deep feature extraction using a convolution neural network based on partial semantic weighted aggregation; filtering features of image information in deep networks; the extraction and distributed storage of video summary keys; the establishment of distributed product quantitative hash coding model of image, realizing the distributed coding compression of high-dimensional features. The video parallel retrieval method proposed in this design has the advantages of high retrieval accuracy and good retrieval efficiency.

**Keywords:** Deep hash · Convolution neural network · High precision

## 1 Introduction

In terms of video image feature extraction, content-based video retrieval initially mainly manually extracted features through subjective judgment for specific scenes, such as texture, edge, shape, and color features, which is easy to lead to incomplete image feature extraction [1–5]. The emergence of convolution neural networks (Convolution Neural Networks, CNN) provides an implicit automatic image feature extraction scheme for learning. The CNN-based depth feature can not only retain more image detail information, but also perform traditional features in terms of image feature representation. In recent CNN-based video retrieval studies [6–9]. The VGG model has deep network layers and good versatility, but the number of training features is very large, and more noise features are independent of the image information. Therefore, it is necessary to suppress the background noise in the deep feature extraction part and highlight the effective discrimination part, so as to extract the more effective depth feature information [10–14].

To improve the accuracy of video retrieval, the extracted video key-frame image features need to retain more image information. Thus its features tend to include higher dimensions such as the CNN model extracted by VGG-16 based deep features of 4096 dimensions and large feature dimensions also reduce the efficiency of feature storage and retrieval [15–19]. Therefore, the coding compression of the high-dimensional characteristic vector is achieved by using the hash coding method, thus saving the storage

space and improving the computational efficiency [20–24]. However, these studies are generally in a single-machine environment, and for distributed storage scenarios, we still need to first concentrate all the data on a single node for coding model training, bringing higher data calculation and storage pressure. Therefore, it is urgently necessary to adapt to the video image distributed storage environment for distributed hash coding model training, so as to optimize the process of hash model training and computational storage [25–29].

In order to solve the above existing technical problems and improve the accuracy and efficiency of video retrieval. A deep hash-based parallel video retrieval method is proposed, combined with the characteristics of large-scale video retrieval.

## 2 System Design

The method of video parallel retrieval based on deep hash, the overall architecture schematic diagram is shown in Fig. 1, mainly includes 2 parts:

- (1) Video depth feature extraction of frame image based on CNN: extract the original video data set to obtain video summary data set; conduct weighted aggregation based on VGG network and PWA model, screen and form deep feature extraction model; and generate image depth feature set based on the trained model.
- (2) Video key-image feature quantification based on distributed hash: the training set is divided according to the video summary image depth feature set, and the distributed hash coding model is trained; the image is generated according to the coding model; the coding search according to the coding model to improve the efficiency of searching.

## 3 Specific Implementation Plan

### 3.1 Deep Feature Extraction of Video Key-Images

First, the video summary image is extracted by the key-frame, and then the deep convolution feature set through the convolution neural network VGG16 network model, and then select the partial semantic based weighted aggregation method PWA to obtain the final feature vector. The feature screening process utilizes an unsupervised strategy to select a partial convolution layer filter to generate a probabilistic weight scheme, and obtains the final feature representation by aggregating the weighted region representation corresponding to each semantic content.

As shown in Fig. 2, the video key-image depth feature extraction process is as follows:

- (1) For the input query image  $I$ , first pass to the restrained deep network VGG-16 model to extract deep convolution layer features  $f$  (consisting of  $C$  channel feature maps, each with height  $H$  and width  $W$ ). The image is represented by  $N$  a filtered partial detector weighted aggregate, which is a vector representation in  $N * C$  dimensions.

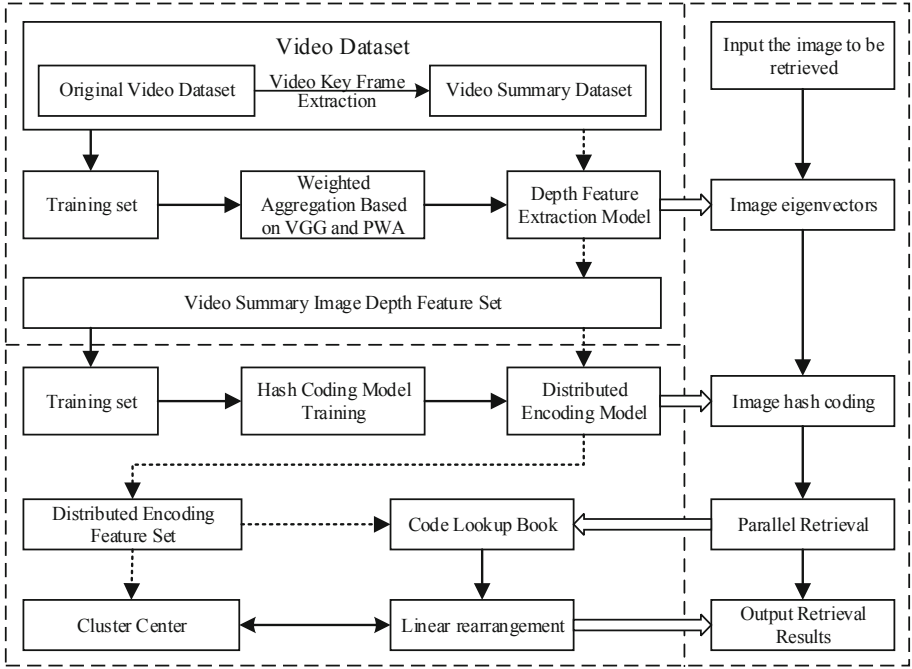


Fig. 1. Schematic drawing of the overall architecture

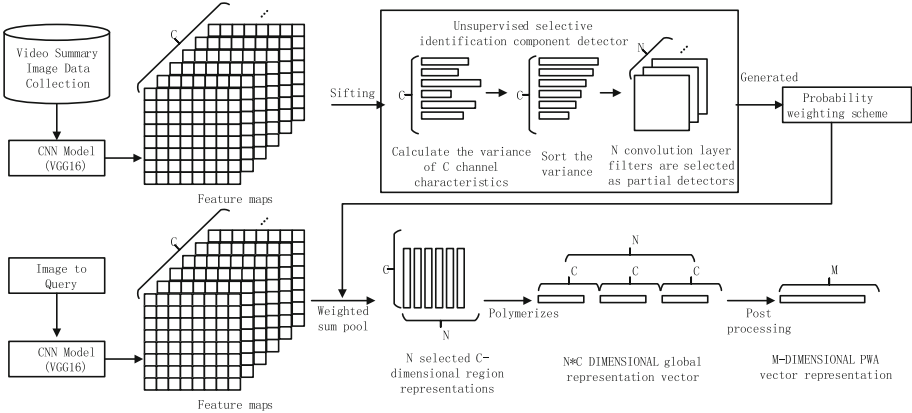


Fig. 2. Diagrammatic of depth feature extraction process of video frame image

- (2) The selection of partial discriminant detectors is trained based on the video key-image feature data-set, and the feature graph channels with larger differences are selected. Therefore, screened by calculating the variance of each channel feature, the  $C$  channel variance of the  $C$  dimension vector  $V = \{v_1, v_2, \dots, v_C, \dots, v_C\}$ :

$$V = \frac{1}{D} \sum_{i=1}^D (g_i - \bar{g})^2$$

- (3) The variance  $\{v_1, v_2, \dots, v_C\}$  of the channel  $C$  is sorted, and the former  $N$  discriminant convolution layer filter with the largest variance is selected as some detector. Then, the probability weight scheme is generated via unsupervised policies, each corresponding to the implied fixed semantic content. The construction represented by the weighted PWA of the selected probability weight scheme begins with a weighted sum set of dimensional depth convolution  $C \times W \times H$  features  $f$  of an image  $I$  with height  $H$  and width  $W$ :

$$\psi_n(I) = \sum_{x=1}^W \sum_{y=1}^H w_n(x, y) f(x, y)$$

The coefficients  $w_n$  are the normalized weight, depending on the active value  $v_n(x, y)$  in the position  $(x, y)$  of the selected probability weight scheme generated by a partial detector  $n$ :

$$w_n(x, y) = \left( \frac{v_n(x, y)}{\left( \sum_{x=1}^W \sum_{y=1}^H v_n(x, y)^\alpha \right)^{1/\alpha}} \right)^{1/\beta}$$

The sum of and are the parameters of power normalization  $\alpha$  and  $\beta$  power scaling, respectively.

- (4) Obtain  $N$  selected dimension  $C$  region representation  $\psi_n(I)$  from the weighted and pooling process, and then get a global dimension  $N \times C$  representation vector  $\psi(I)$  by connecting the selected region representation:

$$\psi(I) = [\psi_1, \psi_2, \dots, \psi_N]$$

Select the  $N$  part detector according to the variance value of the channel  $C$  of the characteristic set, which improves the performance and computational efficiency.

- (5) Implement  $l_2$ -normalization, master component analysis (Principal Component Analysis, PCA) compression and whitening of the global representation  $\psi(I)$  through post-processing, and obtain the final image feature  $M$  dimensional representation  $\psi_{PWA}(I)$ :

$$\psi_{PWA}(I) = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_M)^{-1} V \frac{\psi(I)}{\|\psi(I)\|_2}$$

Where  $V$  is the PCA matrix of size  $M \times N$ ,  $M$  is the number of reserved dimensions,  $\sigma_1, \sigma_2, \dots, \sigma_M$  is the associated singular value.

### 3.2 Distributed Hash Encoding Model

Distributed feature encoding model training is a hash product quantization coding for the image feature set output by a convolution neural network and achieves fast retrieval of an arbitrary image query via distributed coding storage. First, hash coding learning on

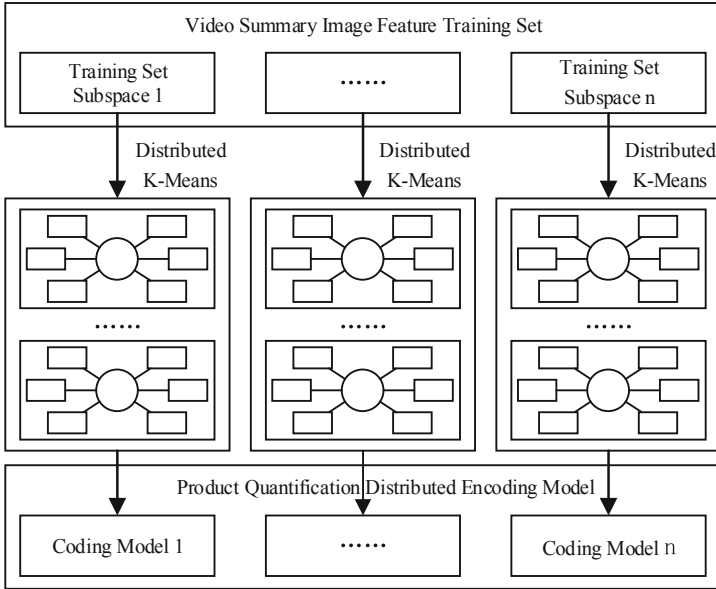


Fig. 3. Schematic of distributed hash coding model

the training data-set to obtain the feature coding model; then the trained coding model is applied to the original video key-image feature data-set to realize its compression coding representation and data distributed storage.

As shown in Fig. 3, the video key-image feature quantification process based on distributed hash is as follows:

- (1) The distributed computing process in the Spark platform distributed stores the summary image feature data-set in the data store RDD, divided into multiple partitions on different nodes in the cluster.
- (2) For a summary image feature data-set  $X$  of one dimension  $p$ . Data distributed stored on  $m$  computing nodes and thus divided into  $m$  subsets  $X = [X_1, \dots, X_m]$ .
- (3) K-Means clustering for each subset. Each subset contains  $n$  cluster centers. Total  $m \times n$  distributed matrix is required for storage. The encoding matrix is represented by the  $B = [B_1, B_2, \dots, B_S]$ . According to the characteristics of distributed data, the target function of distributed product quantification is:

$$\begin{aligned} \ell_{PQ} &= \min \sum_{t=1}^S \left\| X_t - \begin{bmatrix} C^1 B_t^1 \\ \dots \\ C^m B_t^m \end{bmatrix} \right\|_F^2 \\ &= \min \sum_{t=1}^S \left\| \begin{array}{c} X_t^1 - C^1 B_t^1 \\ \dots \\ X_t^m - C^m B_t^m \end{array} \right\|_F^2 \end{aligned}$$

- (4) The trained coded model is compressed and coded for the distributed stored video key-image depth feature data-set. First, the set of eigenvectors distributed on the  $m$  storage nodes is divided into  $m$  parts, and then encoding the compressed K-Means prediction for the subset of each vector using a distributed coding model, so as to obtain the cluster center of each eigenvector, and then make the encoding representation of each vector subset using its cluster center.

## 4 Conclusion

Deep feature extraction of video key-frame images by convolution neural network based on partial semantic-weighted aggregation method, ensures the key information retention of the original image through product-quantified hash coding model.

**Acknowledgement.** This work was supported in part by Xuzhou Science and Technology Plan Project (Grant No. KC19003). Science and technology project of Jiangsu Provincial Department of housing and construction (Grant No. 2018ZD265). Major projects of natural science research in Colleges and universities of Jiangsu Province (Grant No. 19KJA470002) and Xuzhou Science and Technology Plan Project (KC21303).

## References

1. Xiong, S., Lei, M., Liu, C.: Batch Images parallel retrieval based on deep hashing. *J. Chongqing Univ. Technol. (Nat. Sci.)* (2018)
2. Jiang, Y., Zhuo, J., Zhang, J., et al.: The optimization of parallel convolutional RBM based on spark. *Int. J. Wavelets Multiresolut. Inf. Process.* (2018)
3. Mou, F.U., Yang, H., Tangmei, W.U., et al.: Fast video transcoding method based on spark streaming. *J. Comput. Appl.* **38**, 3500 (2018)
4. Qin, J., Chen, J., Xiang, X., et al.: A privacy-preserving image retrieval method based on deep learning and adaptive weighted fusion. *J. Real-Time Image Process.* **17**(1), 161–173 (2020)
5. Dong, Y., Li, J.: Video retrieval based on deep convolutional neural network (2017)
6. Wan, L., Zhang, G., Li, H., et al.: A novel bearing fault diagnosis method using spark-based parallel ACO-K-means clustering algorithm. *IEEE Access* **PP**(99), 1 (2021)
7. Shi, L., Liu, S., Shi, Y., et al.: Sea ice concentration products over polar regions with Chinese FY3C/MWRI Data (2021)
8. Liu, P., Ye, S., Wang, C., et al.: Spark-based parallel genetic algorithm for simulating a solution of optimal deployment of an underwater sensor network. *Sensors* **19**(12), 2717–2727 (2019)
9. Langer, M., Hall, A., He, Z., et al.: MPCA SGD—a method for distributed training of deep learning models on spark. *IEEE Trans. Parallel Distrib. Syst.* **29**, 2540–2556 (2018)
10. Hou, J., Sheng, W.U., Yingna, L.I.: Research on parallel K-means clustering model based on spark. *Comput. Digit. Eng.* (2018)
11. Qin, J., Li, H., Xiang, X., et al.: An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing. *IEEE Access* **7**, 24626–24633 (2019)
12. Cao, Y., Wang, N., Xu, Z., et al.: Network big data classification processing method based on Spark and distributed KNN classifier. *Appl. Res. Comput.* (2019)
13. Deng, Q., Yang, N.: Research of improved parallel K-means algorithm based on spark framework. *Intell. Comput. Appl.* **8**, 76–78 (2018)

14. Jing, W., Zhang, D., Song, H.: An application of ternary hash retrieval method for remote sensing images in panoramic video. *IEEE Access* **8**, 140822–140830 (2020)
15. Wang, Y., Nie, X., Shi, Y., et al.: Attention-based video hashing for large-scale video retrieval. *IEEE Trans. Cogn. Dev. Syst.* **PP(99)**, 1 (2019)
16. Hajkacem, M., N'Cir, C., Essoussi, N.: A parallel text clustering method using Spark and hashing. *Computing* (6) (2021)
17. Gupta, A., Thakur, H., Shrivastava, R., et al.: A big data analysis framework using apache spark and deep learning. In: *IEEE International Conference on Data Mining Workshops* (2017)
18. Lunga, D., Gerrand, J., Yang, L., et al.: Apache spark accelerated deep learning inference for large scale satellite image analytics. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **13**, 271–283 (2020)
19. Khan, S., Liu, X., Alam, M.: A spark ML driven preprocessing approach for deep learning based scholarly data applications (2019)
20. Uddin, M.A., Joolee, J.B., Sohn, K.A.: Dynamic facial expression understanding using deep spatiotemporal LDSP on spark. *IEEE Access* **PP(99)**, 1 (2021)
21. Yang, F., Wang, H., Fu, J.: Improvement of recommendation algorithm based on collaborative deep learning and its parallelization on spark. *J. Parallel Distrib. Comput.* **148**(2), 58–68 (2021)
22. Zaouk, K., Song, F., Lyu, C., et al.: Neural-based modeling for performance tuning of spark data analytics (2021)
23. Takam, C.A., Samba, O., Kouanou, A.T., et al.: Spark architecture for deep learning-based dose optimization in medical imaging. *Inform. Med. Unlocked* **19**, 100335 (2020)
24. Sundareswaran, A., Lavanya, K.: Real-time vehicle traffic prediction in apache spark using ensemble learning for deep neural networks. *Int. J. Intell. Inf. Technol. (IJIT)* **16**, 19–36 (2020)
25. Khan, S., Khan, M., Iqbal, N., et al.: Spark based parallel deep neural network model for classification of large scale RNAs into piRNAs and non-piRNAs. *IEEE Access* **PP(99)**, 1 (2020)
26. Singh, P., Singh, S., Mishra, P.K., et al.: A data structure perspective to the RDD-based Apriori algorithm on spark. *SSRN Electron. J.* (2019)
27. Wang, L., Chen, C., Yun, S.U.: Design and implementation of massive ship density distribution calculation system based on spark. *Ship Electron. Eng.* (2019)
28. Lunga, D., Gerrand, J., Yang, H.L., et al.: Apache spark accelerated deep learning inference for large scale satellite image analytics (2019)
29. Shmeis, Z., Jaber, M.: A rewrite-based optimizer for spark. *Future Gener. Comput. Syst.* **98**, 586–599 (2019)