Fusing Appearance Features and Correlation Features for Face Video Retrieval

Chenchen Jing, Zhen Dong, Mingtao Pei, and Yunde Jia

Beijing Laboratory of Intelligent Information Technology,
School of Computer Science, Beijing Institute of Technology,
Beijing 100081, P.R. China
Email:{jingchenchen1996, dongzhen, peimt, jiayunde}@bit.edu.cn

Abstract. Face video retrieval has drawn considerable research attention recently. Most prior research mainly focused on either appearance features or correlation features, which could degrade retrieval performance. In this paper, we fuse appearance features and correlation features to exploit rich information of face videos for face video retrieval via a deep convolutional neural network. The network extracts appearance feature and correlation feature from a frame and the covariance matrix of a face video, respectively, and fuses them to obtain a comprehensive video representation. The fused feature is projected to a low-dimensional Hamming space via hash functions for the retrieval task. The network integrates feature extractions, feature fusion, and hash learning into a unified optimization framework to guarantee optimal compatibility of appearance features and correlation features. Experiments on two challenging TV-Series datasets demonstrate the effectiveness of the proposed method.

Keywords: Face video retrieval, Deep CNN, Appearance Features, Correlation Features

1 Introduction

Recent years have witnessed a tremendous explosion of multimedia data, especially videos. Millions of videos are uploaded every day to the Internet via social networking websites, mobile applications, etc. Face video retrieval aims to retrieve videos of a particular person from a video database given one face video of him/her, and has increasingly attracted more attention. It has a wide range of applications such as locating and recognizing suspects from surveillance videos, intelligent fast forward of movies, and collecting all videos of favorite character from the TV-Series.

Face video representation is critical in face video retrieval. Existing face video representation methods can be roughly divided into two categories: appearance based methods and correlation based methods. Appearance based methods focus on characterizing human faces via appearance features such as such as color and texture [3, 6, 11, 18]. They regard a face video as a set of images and fuse the well-learned representations of each frame to get the final video representation. In general, a face video comprises multiple consecutive frames, and each frame depicts appearance features of the face, which could vary greatly from frame to frame. Hence, ignoring correlation features and simply treating videos as image sets is inadvisable. Correlation based methods

treat a video as a whole and utilize the second-order statistics information such as the covariance matrix (Cov) feature of a video to capture the video data variations in a statistical manner [14–16]. Albeit covariance matrix has been proved natural and efficient for video representation, it only represents linear correlations of frames and does not capture the appearance feature of each frame. In this paper, we fuse appearance features and correlation features to obtain comprehensive video representations for face video retrieval.

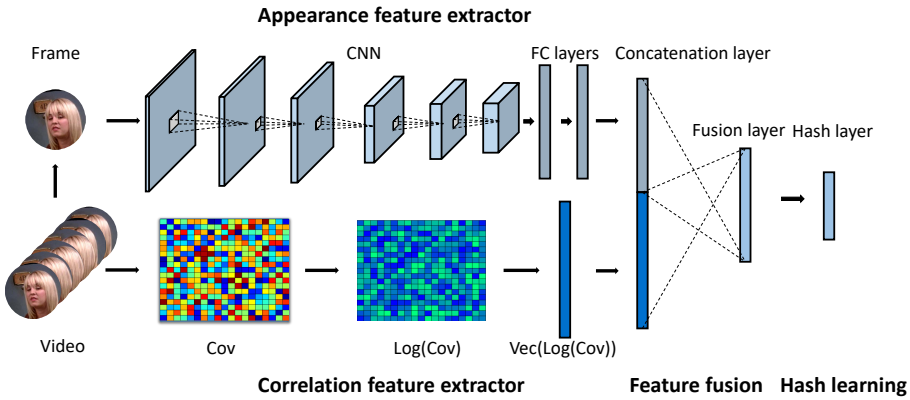

Fig. 1: The architecture of the proposed network.

Fig. 1 shows the architecture of the proposed network. Given a frame of a face video as the appearance feature and the covariance matrix as the correlation feature, the network generates a comprehensive face video representation. The network contains four components: appearance feature extractor module, correlation feature extractor module, feature fusion module, and hash learning module. The appearance feature extractor module is utilized to learn discriminative appearance features to alleviate the problem of large intra-class variations of videos in face video retrieval. The correlation feature extractor module projects nonsingular covariance matrices to a Euclidean space and then vectorizes them, since these matrices lie on a Riemannian manifold [14]. The following feature fusion module fuses appearance features and correlation features to obtain comprehensive video representations via a concatenation layer and a fusion layer. The hash learning module is exploited to project high-dimensional fused features to a low-dimensional Hamming space, considering that the fused features are designed for the retrieval task, which have an eagerly demand of time and space saving. Implemented via neural network and trained jointly, all these modules actually form an end-to-end architecture.

The appearance feature extractor module and the correlation feature extractor module appropriately process the input frame and covariance matrix, respectively, aiming to provide effective appearance features and correlation features for the feature fusion module. The feature fusion module fuses these features perfectly to utilize rich information of face videos. The hash learning module generates compact representations to

make the fusion more compact for retrieval. The four components are integrated into a unified optimization framework to ensure that appearance features and correlation features are compatibly fused for the final retrieval task. We conduct experiments on two challenging TV-Series datasets (*the Big Bang Theory* and *the Prison Break*) [16]. The excellent experimental results demonstrate the superiority of fusing appearance features and correlation features for face video retrieval.

2 Related work

In this section, we give a brief review of related works including face video retrieval and hashing methods.

2.1 Face Video Retrieval

Various methods for face video retrieval have been proposed [1, 2, 6, 14, 16, 19, 21] in recent years. Arandjelovic and Zisserman [1, 2] utilized an identity preserving and variation insensitive signature image to represent a face video and developed an retrieval system. Sivic et al. [21] represented a face video as a probability distribution to harness multiple frames of the video. They built a complete retrieval system covered every key procedure including face detection, face tracking, etc. The high-dimensional features used in above methods are not applicable to retrieval task by the current view. Li et al. [14] proposed compact video code (CVC) for face video retrieval. They computed covariance matrix from frames' DCT features to utilize the second-order statistics information. Furthermore, they proposed hierarchical hybrid statistic based video binary code [16]. This method first utilizes different parameterized fisher vectors as frame representation and then executes CVC in the Reproducing Kernel Hilbert Space. Dong et al. [6] proposed an end-to-end deep network to learn discriminative and compact frame representations and fuse them to get final video representation, which is the first attempt to employ a neural network to face video retrieval. Aiming to capture local relationships between frames, Qiao et al. [19] designed a multi-branch CNN, which learns video-level features by temporal feature pooling. Above methods mainly focuses on either appearance features or correlation features. Different from these works, our network fuses both the appearance features and correlation features to obtain comprehensive video representations for face video retrieval.

2.2 Hashing Methods

Hashing methods are efficacious solutions to nearest neighbor search problem and have been widely studied recently. Locality sensitive hashing (LSH) [7] is a representative data-independent hashing methods. Since random projections functions instead of hash functions learned from data are adopted, LSH still needs long hash codes to achieve satisfactory performance. Therefore, learning based data-dependent hashing methods have become increasingly popular because of the benefit that taking full advantage of data structure or supervision information of training data. The data-dependent methods

can be further divided into unsupervised, semi-supervised and supervised methods. Unsupervised methods learn hash functions only exploiting feature information of training data without supervision information, including spectral hashing [23], iterative quantization hashing (ITQ) [8], gaussian mixture model embedding [9]. Due to supervision information utilized in semi-supervised and supervised methods, performance of these methods superior to unsupervised methods in general. Maximizing variance and independence of hash bits over both labeled and unlabeled data, semi-supervised hashing [22] is a typical semi-supervised method. Both utilizing supervision information, discriminative binary coding [20] use point wise labels, and supervised hashing with kernels [17], robust multiple instance hashing [4] use pair wise labels. In addition, ranking label such as triplet label which aims to preserve rank order among samples are widely used. Ranking based methods include column generation hashing [13], part-based deep hashing [26]. In this work, triplet label supervision is exploited in the final binary space for the improvement of the retrieval performance.

3 Method

3.1 Appearance Feature Extractor Module

The appearance feature extractor module of our architecture is based on the well-known AlexNet [12]. The AlexNet has five types of layers: convolutional layer, max-pooling layer, local contrast normalization layer, fully connected layer and the non-linear ReLU activation layer. The last but one fully connected layer, "FC7" layer is followed by a fully connected layer with $1,000$ output neurons and a softmax layer to compute the probability distribution over the categories. Previous studies presented better performances of the 4096-dim features of the "FC7" layer than a large amount of the hand-crafted features [12]. The original AlexNet, which is trained on the ImageNet [5], is not specifically designed for face recognition. Hence, we fine-tune the Alexnet on CASIA-WebFace [25] to transfer the network from natural image domain to face image domain. The layers before "FC7" layer of the fine-tuned AlexNet is utilized as better initializations to learn our appearance feature extractor module.

3.2 Correlation Feature Extractor Module

The correlation feature extractor module projects covariance matrices to a Euclidean space and vectorizes them. With the fine-tuned AlexNet, CNN feature of each frame is extracted through forward propagation. Let $\boldsymbol{F} = [\boldsymbol{f_1}, \boldsymbol{f_2}, ..., \boldsymbol{f_n}] \in \mathbb{R}^{d \times n}$ be the CNN features of a video where d represent the dimension of CNN feature and n is the number of frames, $\boldsymbol{f_i}$ denotes the i^{th} frame with d-dim CNN feature. The covariance matrix of this video is defined as

$$\boldsymbol{C} = \frac{1}{n-1} \sum_{i=1}^{n} (\boldsymbol{f_i} - \overline{\boldsymbol{f}})(\boldsymbol{f_i} - \overline{\boldsymbol{f}})^T, \tag{1}$$

where $\overline{\boldsymbol{f}}$ denotes the mean of all the frame features of this video. Since nonsingular covariance matrices lie on a Riemannian manifold [14], Log-Euclidean Distance is resorted to bridge the gap between Riemannian manifold and Euclidean space. Projecting

points on the Riemannian manifold to a Euclidean space via logarithm map, then the distance of two points $\boldsymbol{C_1}$,$\boldsymbol{C_2}$ is given by

$$d_{LED} = \| \, log(\boldsymbol{C_1}) - log(\boldsymbol{C_2}) \, \|_F, \tag{2}$$

where log is the matrix logarithm operator, and $\| \cdot \|_F$ denotes the matrix Frobenius norm. Since the off-diagonal entries of the matrix $log(\boldsymbol{C})$ is counted twice during norm computation, we vectorize the covariance matrix in the form of

$$vec(log(\boldsymbol{C})) = [v_{1,1}, \sqrt{2}v_{1,2}, ..., v_{2,2}, \sqrt{2}v_{2,3}, ..., v_{d,d}], \tag{3}$$

to generate a $d(d+1)/2$-dim feature vector, where $v_{i,j}$ is the i^{th} row, j^{th} column element of $log(\boldsymbol{C})$.

3.3 Feature Fusion Module

In order to fuse the outputs of the two feature extractor modules, namely the CNN feature and the Cov feature, into an unitary feature, the feature fusion module is introduced in the proposed network. The feature fusion module contains a concatenation layer and a fusion layer. Given two vectors of d_1-dim and d_2-dim, respectively, a concatenation layer concatenates them together to get a $d_1 + d_2$-dim vector. Let $\boldsymbol{f} \in \mathbb{R}^{d_1}$ be the CNN feature, and $\boldsymbol{c} \in \mathbb{R}^{d_2}$ denotes the Cov feature. The output of the catenation layer is

$$\boldsymbol{x} = [\boldsymbol{f}, \boldsymbol{c}] \in \mathbb{R}^{(d_1+d_2)}. \tag{4}$$

As a simple concatenation of two features, $\boldsymbol{x}$ is an intermediate result provided for the fusion layer. The following fusion layer is a fully connected layer whose output is computed by

$$\boldsymbol{y} = \boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b}, \tag{5}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are weight and bias of this layer. Followed by the hash learning module, the feature fusion module guarantees the two features are combined directly for the retrieval task, *i.e.*, the features are fused for hashing, and the performance of hash codes is able to guide the fusion of two features. Moreover, the feature fusion module ensures CNN features are constrained by Cov features through back propagation, *i.e.*, parameters of the appearance feature extractor module are influenced by vectorized Cov features [24]. Hence, by introducing the feature fusion module, the whole network are designed to fuse CNN features and Cov features optimally for the final retrieval task.

3.4 Hash Learning Module

The output of the fusion layer are comprehensive but high-dimensional video representations which are not fit for the retrieval task. Hence the hash learning module is utilized to map fused features to a low-dimensional Hamming space. Specially, we enforce triplet ranking loss to hash functions to preserve the data similarity.

Suppose that the deep CNN have mapped face videos to a binary space: $\{+1, -1\}^s$, where s is the length of hash code. The triplet ranking loss reflects the relative similarities in the form as "video $\mathbf{q}$ is more similar to $\widetilde{\mathbf{q}}$ than $\widehat{\mathbf{q}}$". Let $(\mathbf{q}, \widetilde{\mathbf{q}})$ be positive pair

whose samples are from the same person, and $(\mathbf{q}, \widehat{\mathbf{q}})$ be the negative pair whose samples are from different individuals, the loss of one triplet is thus formulated as

$$l\big(\mathbf{q}, \widetilde{\mathbf{q}}, \widehat{\mathbf{q}}\big) = \max\Big(d\big(\mathbf{q}, \widetilde{\mathbf{q}}\big) - d\big(\mathbf{q}, \widehat{\mathbf{q}}\big) + \delta, 0\Big), \tag{6}$$

where $d(\boldsymbol{\theta_1}, \boldsymbol{\theta_1}) = (s - \boldsymbol{\theta_1^T \theta_2})/2$ is the Hamming distance in the binary space, and $\delta \geq 0$ denotes the margin of the distance differences between positive and negative pairs. Define the training video set as $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2, ..., \mathbf{Q}_C]$ of C classes, the objective of the our deep CNN is

$$\min_{\mathbf{W}^\star, \mathbf{W}} \sum_{i=1}^{C} \sum_{\substack{\mathbf{q}, \widetilde{\mathbf{q}} \in \mathbf{Q}_i \\ \mathbf{q} \neq \widetilde{\mathbf{q}}}} \sum_{j \neq i, \widehat{\mathbf{q}} \in \mathbf{Q}_j} l\big(\mathbf{q}, \widetilde{\mathbf{q}}, \widehat{\mathbf{q}}\big), \tag{7}$$

where $\mathbf{W}$ is the parameters of the last layer (hash functions), and $\mathbf{W}^\star$ represents the parameters of the front layers.

To solve Eq.(7), the gradients of Eq.(6) is needed. Since the hash function contains the sign function $sgn(\cdot)$ which is non-smooth and non-differentiable, we use $\tanh(\cdot)$ instead of the sign function during the fine-tuning procedure. Therefore, the gradients of Eq.(6) w.r.t. hash codes are given by

$$\frac{\partial l}{\partial \mathbf{q}} = \frac{1}{2}(\widehat{\mathbf{q}} - \widetilde{\mathbf{q}}) \times I, \frac{\partial l}{\partial \widetilde{\mathbf{q}}} = -\frac{1}{2}\mathbf{q} \times I, \frac{\partial l}{\partial \widehat{\mathbf{q}}} = \frac{1}{2}\mathbf{q} \times I, \tag{8}$$

where I is a binary function which returns 1 when $d\big(\mathbf{q}, \widetilde{\mathbf{q}}\big) - d\big(\mathbf{q}, \widehat{\mathbf{q}}\big) + \delta > 0$ and 0 for other occasions. Obtaining these gradients, the optimization procedure can be conducted via the back-propagation algorithm.

4 Experiments

4.1 Dataset and Experimental Settings

We conduct experiments on the ICT-TV dataset [16] to evaluate the proposed method. The ICT-TV dataset contains two large scale video sets from two American shows: the Big Bang Theory (BBT) and Prison Break (PB). The two TV-Series are quite different in filming styles. The BBT is a sitcom with 5 main characters, in which most scenes are taken indoors. Each episode lasts about 20 minutes. Differently, many shots of the PB are taken outside during the episodes of about 42 minutes long. This results in a large range of different illumination conditions. All the face videos are collected from the whole first season of both TV series, *i.e.*, 17 episodes of BBT, and 22 episodes of PB, and the number of face videos of the two sets are $4,667$ and $9,435$, respectively.

We compare our method with seven state-of-the-art hashing methods: LSH [7], SH [23], ITQ [8], SITQ [8], RR [8], SSH [22], KSH [17], and three face video retrieval methods: DBHR [6], HHS-VBC [16], and SPC-CVC [15]. For each TV-Series dataset, we randomly select 10 face videos per actor or actress for training hash functions, and use the rest face videos for testing. Same to [16], the query set consists of 10 face videos

Table 1: Comparison mAPs of our methods.

Methods	*the Big Bang Theory*						*Prison Break*					
	8 bits	16 bits	32 bits	64 bits	128 bits	256 bits	8 bits	16 bits	32 bits	64 bits	128 bits	256 bits
Ours (EC-RS)	0.9407	0.9376	0.9362	0.9437	0.9373	0.9413	0.4370	0.4716	0.5122	0.5300	0.5382	0.5672
Ours (EC-AP)	0.9430	0.9525	0.9445	0.9628	0.9563	0.9625	0.4873	0.5320	0.5869	0.5988	0.6184	0.6438
Ours (WC-RS)	0.9604	0.9687	0.9705	0.9702	0.9742	0.9746	0.6997	0.7195	0.7493	0.7554	0.7694	0.7844
Ours (WC-AP)	**0.9665**	**0.9849**	**0.9909**	**0.9917**	**0.9853**	**0.9924**	**0.7667**	**0.7956**	**0.8056**	**0.8188**	**0.8377**	**0.8461**

Table 2: Comparison mAPs with comparison methods.

Methods	*the Big Bang Theory*						*Prison Break*					
	8 bits	16 bits	32 bits	64 bits	128 bits	256 bits	8 bits	16 bits	32 bits	64 bits	128 bits	256 bits
LSH [7]	0.4302	0.5301	0.6874	0.7486	0.8541	0.8761	0.1308	0.1299	0.1906	0.2672	0.3487	0.4264
RR [8]	0.8252	0.8738	0.8381	0.8558	0.8910	0.9131	0.2801	0.3806	0.4209	0.4637	0.4916	0.5115
ITQ [8]	0.8419	0.9019	0.8889	0.9130	0.9252	0.9345	0.3571	0.4450	0.5074	0.5337	0.5370	0.5332
SH [23]	0.6403	0.5425	0.5633	0.5332	0.4915	0.4447	0.2615	0.3135	0.3346	0.3293	0.2944	0.2675
SSH [22]	0.8113	0.8173	0.6791	0.6008	0.5571	0.5250	0.3435	0.4380	0.3293	0.2794	0.2651	0.2598
KSH [17]	0.8338	0.9116	0.9388	0.9441	0.9430	0.9435	0.5028	0.6155	0.6313	0.7041	0.7227	0.7456
SITQ [8]	0.8515	0.9439	0.9516	0.9500	0.9508	0.9483	0.4848	0.6072	0.6715	0.7008	0.6903	0.6742
DBHR [6]	0.9497	0.9696	0.9805	0.9803	0.9742	0.9814	0.7496	0.7775	0.7576	0.7857	0.8262	0.8293
HHS-VBC [16]	0.5099	0.5934	0.6718	0.6821	0.7170	0.7401	0.1388	0.1445	0.1560	0.1629	0.1784	0.1982
SPC-CVC [15]	0.5202	0.6471	0.7325	0.7543	0.7740	0.7899	0.1401	0.1525	0.1674	0.1903	0.2099	0.2287
Ours (WC-AP)	**0.9665**	**0.9849**	**0.9909**	**0.9917**	**0.9853**	**0.9924**	**0.7667**	**0.7956**	**0.8056**	**0.8188**	**0.8377**	**0.8461**

of each main character. To evaluate the quality of our method, we use four evaluation criterions: the mean Average Precision (mAP), the Precision Recall curve (PR curve), Precision curve w.r.t. different number of top returned samples (PN curve), and Recall curve w.r.t. different number of top returned samples (RN curve). For fair comparisons, all the methods use the same training and testing sets.

The network is trained using Caffe deep learning tool [10]. Stochastic gradient descent is utilized to optimize the network, with momentum of 0.9 and weight decay of 0.0005. The learning rate of the optimization is initialized as 0.001 and decreased according to the polynomial policy with power value of 0.6. The mini-batch size of the training samples is 64, and the triplets are randomly generated based on the labels. The total number of the iterations is $50,000$. In our experiments, we execute PCA to get 100-dim CNN features of faces and the dimension of the final Cov features is 5050.

4.2 Results and Discussions

In testing, we generate representations of test face videos in two ways: random selection and average pooling:

- **Random Selection (RS):** We randomly select a frame of a face video and compute the covariance matrix as the inputs of the network. A binary video representation is obtained through forward propagation.
- **Average Pooling (AP):** As for a face video with m frames, all m frames of this video are first inputted into the network with the covariance matrix to obtain m video representations. Then these representations are fused by average pooling to obtain a more robust binary video representation for retrieval.

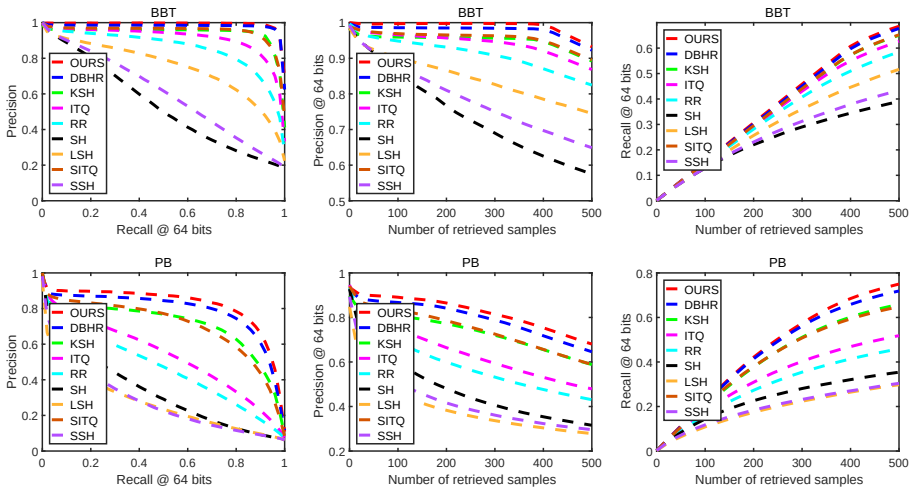

Fig. 2: Comparisons of PR, PN, and RN curves of the face video retrieval experiment on two TV-series datasets.

Moreover, in order to verify the effectiveness of the correlation features, we exclude correlation the feature extractor module and the feature fusion module of the proposed network, *i.e.*, remain the appearance feature extractor module and the hash learning module only, and get a single branch network.

We test the single branch network and the whole proposed network with different representation generation methods. The mAPs are shown on Table 1, where "EC" means "Exclude Correlation features", "WC" means "With Correlation features". It can be seen that video representations generated by average pooling are more robust because of large variations in each video of two datasets. The performance differences between "Ours (EC-RS)" and "Ours (WC-RS)", "Ours (EC-AP)" and "Ours (WC-AP)" demonstrate the effect of correlation features. Our single branch network which only contains the appearance feature extractor module and the hash learning module can achieve comparable results with other methods.

Table 2 lists the mAPs of our method and the comparison methods, and Fig. 2 depicts the comparisons of curves. For the seven hashing methods, mean vector of learned frame representations is computed as the final video hash representation. For fair comparisons, these hashing methods use the 4096-dim input features generated by the AlexNet fine-tuned on the WebFace dataset. As shown in Table 2 and Fig. 2, the proposed method significantly outperforms other comparison methods. The advantages of the proposed method mainly lie in two aspects: the utilization of correlation features and the unified optimization procedure which makes the feature extractor modules, the feature fusion module, and the hash learning module optimally compatible for the retrieval task.

The face video retrieval method DBHR [6] builds an end-to-end deep network to learn discriminative and compact frame representations and fuse them to get final video

representations. The low-rank discriminative binary hashing which is proposed to prelearn hash functions, is utilized to achieve state-of-the-art performances. The comparison results of our method and DBHR are shown on Table 2 and Fig. 2, which certify that fusing appearance features and correlation features is efficient for face video retrieval. HHS-VBC and SPC-CVC use multiple size-variant covariance matrices calculated from fisher vectors and raw intensities as video features, respectively, and learn video hash representations from these covariance matrices. We keep the experimental setting of our method same with them and report the results published in [15]. Our method quite outperforms these two face video retrieval methods, and the main reason is that our method simultaneously optimizes the feature extraction modules, the feature fusion module, and the hash learning module for optimal compatibility, rather than uses fixed features which has nothing to do with the hashing procedure as input. The HHS-VBC and SPC-CVC methods extract features from the 20×16 gray images, which may have influence on the performance. But, it takes so large time and space for running them on larger size face frames that the comparison experiment cannot be conducted under current hardware conditions.

5 Conclusion

In this paper, we fused appearance features and correlation features for face video retrieval via a deep CNN. In the network, the appearance feature extractor module and the correlation feature extractor module extract discriminative appearance features and vecorized correlation features, respectively. The following feature fusion module fuses these features together to exploit rich information of face videos. The hash learning module projects the fused feature to a low-dimensional Hamming space. The network integrates these modules into a unified optimization framework to ensure that appearance features and correlation features are optimally fused for the retrieval task. Our method achieved excellent performances on two challenging TV-Series datasets.

Acknowledgments.

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant No.61472038 and No.61375044.

References

1. Arandjelovic, O., Zisserman, A.: Automatic face recognition for film character retrieval in feature-length films. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 860–867. IEEE (2005)
2. Arandjelović, O., Zisserman, A.: On film character retrieval in feature-length films. Interactive Video pp. 89–105 (2006)
3. Cevikalp, H., Triggs, B.: Face recognition based on image sets. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 2567–2573. IEEE (2010)
4. Conjeti, S., Paschali, M., Katouzian, A., Navab, N.: Learning robust hash codes for multiple instance image retrieval. arXiv preprint arXiv:1703.05724 (2017)

5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. IEEE (2009)
6. Dong, Z., Jia, S., Wu, T., Pei, M.: Face video retrieval via deep learning of binary hash representations. In: AAAI. pp. 3471–3477 (2016)
7. Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In: VLDB. vol. 99, pp. 518–529 (1999)
8. Gong, Y., Lazebnik, S.: Iterative quantization: A procrustean approach to learning binary codes. In: CVPR. pp. 817–824. IEEE (2011)
9. Hoang, T., Do, T.T., Tan, D.K.L., Cheung, N.M.: Enhance feature discrimination for unsupervised hashing. arXiv preprint arXiv:1704.01754 (2017)
10. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 675–678. ACM (2014)
11. Kim, T.K., Kittler, J., Cipolla, R.: Discriminative learning and recognition of image set classes using canonical correlations. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6) (2007)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
13. Li, X., Lin, G., Shen, C., Van Den Hengel, A., Dick, A.R.: Learning hash functions using column generation. In: ICML (1). pp. 142–150 (2013)
14. Li, Y., Wang, R., Cui, Z., Shan, S., Chen, X.: Compact video code and its application to robust face retrieval in tv-series. In: BMVC (2014)
15. Li, Y., Wang, R., Cui, Z., Shan, S., Chen, X.: Spatial pyramid covariance-based compact video code for robust face retrieval in tv-series. IEEE Transactions on Image Processing 25(12), 5905–5919 (2016)
16. Li, Y., Wang, R., Shan, S., Chen, X.: Hierarchical hybrid statistic based video binary code and its application to face retrieval in tv-series. In: FG. pp. 1–8. IEEE (2015)
17. Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: CVPR. pp. 2074–2081. IEEE (2012)
18. Parkhi, O.M., Simonyan, K., Vedaldi, A., Zisserman, A.: A compact and discriminative face track descriptor. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1693–1700 (2014)
19. Qiao, S., Wang, R., Shan, S., Chen, X.: Deep video code for efficient face video retrieval
20. Rastegari, M., Farhadi, A., Forsyth, D.: Attribute discovery via predictable discriminative binary codes. Computer Vision–ECCV 2012 pp. 876–889 (2012)
21. Sivic, J., Everingham, M., Zisserman, A.: Person spotting: video shot retrieval for face sets. In: International Conference on Image and Video Retrieval. pp. 226–236. Springer (2005)
22. Wang, J., Kumar, S., Chang, S.F.: Semi-supervised hashing for scalable image retrieval. In: CVPR. pp. 3424–3431. IEEE (2010)
23. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Advances in neural information processing systems. pp. 1753–1760 (2009)
24. Wu, S., Chen, Y.C., Li, X., Wu, A.C., You, J.J., Zheng, W.S.: An enhanced deep feature representation for person re-identification. In: Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on. pp. 1–8. IEEE (2016)
25. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
26. Zhu, F., Kong, X., Zheng, L., Fu, H., Tian, Q.: Part-based deep hashing for large-scale person re-identification. IEEE Transactions on Image Processing (2017)