PKU_ICST at TRECVID 2018: Instance Search Task

Yuxin Peng, Xin Huang, Yunkan Zhuo,

Zhaoda Ye, Zhang Wen, Jingze Chi, and Junchao Zhang

Institute of Computer Science and Technology,

Peking University, Beijing 100871, China.

pengyuxin@pku.edu.cn

Abstract

We have participated in both two types of Instance Search (INS) task in TRECVID 2018: automatic search and interactive search. For the **automatic search**, we first searched the specific location and person separately, and then fused the above two kinds of search results by instance score fusion. Finally, we proposed a semi-supervised re-ranking method to refine the final instance search results. In the *location-specific search*, we extracted two kinds of features to compute the location similarity score, namely (1) Bag-of-Word (BoW) feature based on Approximate K-means, and (2) deep feature based on Convolutional Neural Networks. In the *person-specific search*, our pipeline consisted of (1) query preprocessing by super-resolution, (2) face recognition on deep models, and (3) text-based score refinement. *Instance score fusion* was performed with a bi-directional strategy on the location-specific and person-specific search, which was to obtain the instance search results by mining the common information in the above two specific search. Moreover, the *semi-supervised re-ranking method* was proposed to filter noisy shots in the final instance search results. For the **interactive search**, we first adopted the same similarity computing approach in automatic search. Then, interactive query expansion strategy was applied for expanding the queries. Finally, we integrated the scores of expanded and original queries to obtain the final search results. The official evaluations showed that our team ranked 1st in both automatic and interactive search.

1 Overview

In TRECVID 2018[1], we have participated in all two types of Instance Search (INS) [2] tasks:

automatic search and interactive search. We have submitted totally 7 runs: 6 automatic runs and 1 interactive run, whose official evaluation results are shown in Table 1. In both automatic search and interactive search, our team ranked 1st among all teams. Table 2 gives the detailed explanation of brief descriptions in Table 1. The overall processes of our approach are shown in Figure 1.

In the 6 automatic runs, the notations "A" and "E" specify whether the video examples were used or not. Notation "A" means video examples are not used, while "E" is the opposite. The methods of two runs are the same if there is only a difference of "A" or "E". Run1_A/E contains basic components of our approach, including ***location-specific search*** based on both BoW feature and deep feature, ***person-specific search*** based on super-resolution preprocessing and deep face recognition, and ***instance score fusion*** of the above two processes. The difference between Run1_A/E and Run2_A/E is that Run2_A/E incorporates transcripts to further exploit person-specific information in video shots. Compared to Run2_A/E, Run3_A/E adopts semi-supervised learning based re-ranking strategy to filter noisy search results. Run4 is an interactive search run with human feedback based on automatic search Run2_E.

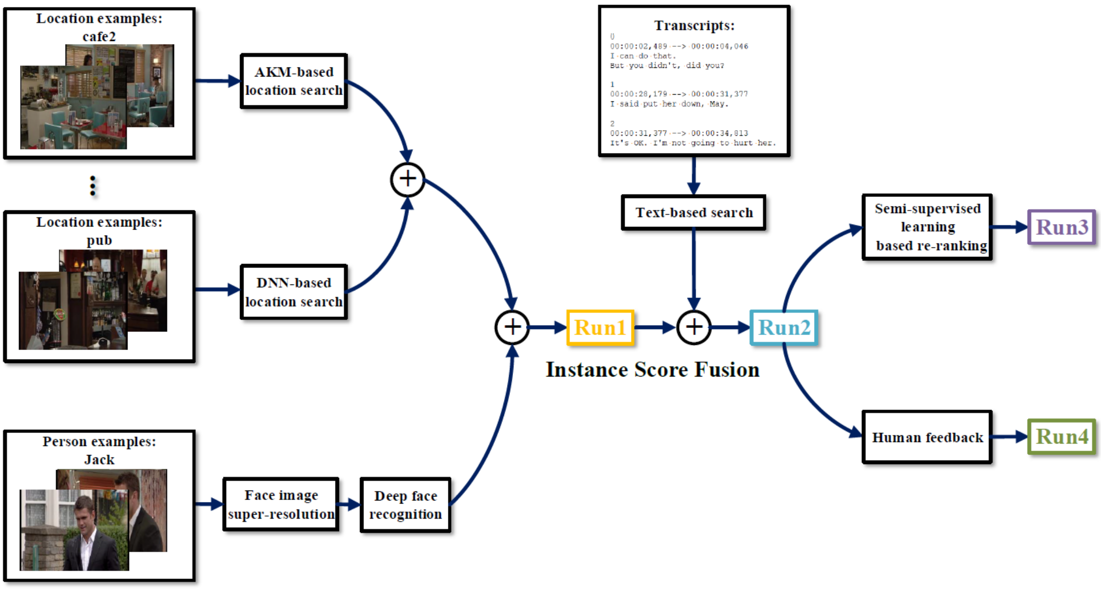

Figure 1: Framework of our approach for the 7 submitted runs.

Table 1: Results of our submitted 7 runs on Instance Search task of TRECVID 2018.

Type	ID	MAP	Brief description
Automatic	PKU_ICST_RUN1_A	0.369	A+D+R+F
	PKU_ICST_RUN1_E	0.392	A+D+R+F
	PKU_ICST_RUN2_A	0.420	A+D+R+F+T
	PKU_ICST_RUN2_E	0.459	A+D+R+F+T
	PKU_ICST_RUN3_A	0.429	A+D+R+F+T+S
	PKU_ICST_RUN3_E	0.463	A+D+R+F+T+S
Interactive	PKU_ICST_RUN4	0.524	A+D+R+F+T+H

Table 2: Description of our methods.

Abbreviation	Description
A	**A**pproximate K-means (AKM) based location search
D	**D**NN-based location search
R	Face Image Super-**R**esolution
F	Deep **F**ace recognition
T	**T**ext-based search
S	**S**emi-supervised learning based re-ranking
H	**H**uman feedback

2 Our Approach

2.1 Location-specific Search

We extracted two kinds of features for location-specific search, namely BoW feature by AKM algorithm and deep feature by convolutional neural network (CNN). Then we took the advantages of both the BoW features and deep features by the fusion strategy.

2.1.1 AKM-based Search

For location search based on BoW features, multiple detectors and descriptors were first employed to generate keypoint features, then keypoint-based BoW features were obtained by AKM algorithm, and finally the location similarity score was calculated by cosine distance measurement. We extracted the keypoint-based BoW features of video shots following the three stages:

(1) We adopted three detectors to detect keypoints from video frames, including Harris Laplace[3], Hessian Affine[4] and MSER[5] detectors. For each detector, 128-dimensional SIFT descriptor[6] and 192-dimensional ColorSIFT descriptor[7] were applied to generate features of the neighboring regions around detected keypoints. Therefore, there were 6 kinds of keypoint features generated for each keyframe.

(2) We used AKM algorithm to cluster each kind of keypoint feature into one-million cluster centroids, and built a visual vocabulary with these cluster centroids.

(3) As each shot contained several keyframes, we assigned each keypoint of all the keyframes into the nearest centroid. The word weights were determined by the keypoint-to-word similarity and region of interest (ROI), and we quantized each shot into a one-million dimensional BoW feature. As a result, totally 6 BoW features were generated for each location query and each test shot, as shown in Figure 2.

After getting 6 kinds of keypoint-based BoW features, we obtained 6 groups of similarity scores by cosine distance measurement based on each kind of feature. Finally, we integrated them by using late fusion strategy to get the AKM-based location similarity:

$$AKM_{ij} = \frac{1}{6}\sum_{k} BoW_{ij}^{(k)} \tag{1}$$

where AKM_{ij} denotes the AKM-based location similarity score between shot i and location j. $BOW_{ij}^{(k)}$ denotes the similarity score between shot *i* and location *j* based on *k*-th BoW feature.

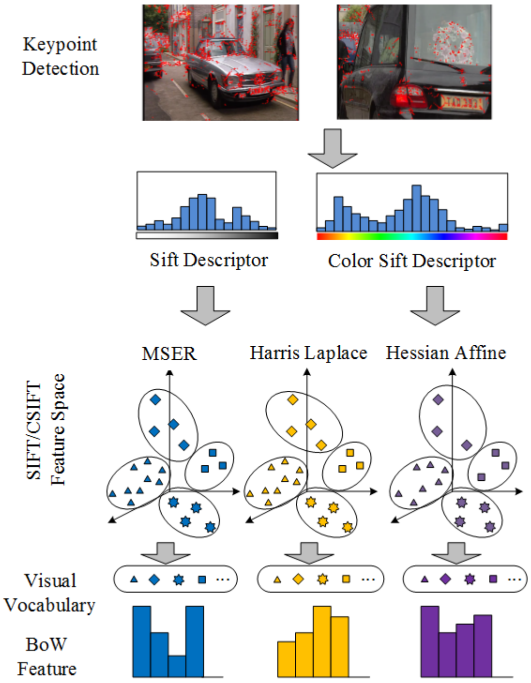

Figure 2: Extracting keypoint-based BOW features.

2.1.2 DNN-based Search

Deep neural networks (DNNs) have achieved great advances in computer vision, which have been widely used for many tasks, such as image/video recognition[8,9], object detection[10,11], and multimedia retrieval[12,13]. Compared to SIFT-based BoW features that mainly capture local textures of images, deep features focus more on semantic information. Inspired by this, DNNs were exploited to further promote local-specific search.

In DNN-based search, we extracted two kinds of location-specific deep features using two different deep models: VGGNet[14] and GoogLeNet[15]. These two kinds of deep features were then used to perform location search, where we exploited not only their discriminative representation abilities, but also the complementarity between them. The DNN-based search pipeline included two stages: model training and similarity computing.

(1) Model Training

Instead of adopting the commonly-used pre-trained DNN models on ImageNet, we pre-trained VGGNet and GoogLeNet with a large-scale scene recognition dataset, namely Places365[16]. It consists of 365 scene categories, including indoor scenes, such as cafeteria and kitchen, as well as natural and urban scenes. The pre-training on Places365 helped properly initialize the deep models for general scene understanding. Then we adopted fine-tuning strategy to transfer the representation ability of above pre-trained models to fit the location search task. When fine-tuning deep models, data augmentation was performed on training data, which was a widely-used strategy to boost the performance of deep models. Specifically, based on the provided location example images, various kinds of transformations were conducted to augment the training examples, including image blurring, noising and scaling, etc.

(2) Similarity Computing

As two deep models were well-trained using the above pre-training and fine-tuning strategies, we extracted deep features from each model respectively, and then used cosine distance to measure the similarities between queries and testing examples. Finally, we averaged the similarity scores of VGGNet and GoogLeNet to get the DNN-based location similarity score, which exploited the complementarity between the two deep models:

$$DNN_{ij} = 1/2\left(VGG_{ij} + GOOGLE_{ij}\right) \qquad (2)$$

where DNN_{ij} denotes the DNN-based location similarity score between shot i and location *j*. VGG_{ij} and $GOOGLE_{ij}$ denote the similarity scores based on VGGNet and GoogLeNet features respectively.

2.1.3 Location Similarity Fusion

After getting the scores of AKM-based and DNN-based location similarities for each location query and test video shot, we conducted late fusion of these two kinds of similarities to obtain the final location similarity $LocSim_{ij} = AKM_{ij} + 3 * DNN_{ij}$. Finally, we obtained the results of location-specific search based on $LocSim_{ij}$.

2.2 Person-specific Search

In person-specific search process, we first conducted face image super-resolution to enhance the quality of person query examples. Then we performed face recognition based on deep convolutional neural network. At last, we adopted text-based search to further improve the retrieval results.

2.2.1 Face Image Super-Resolution

In this year, we adopted the common pipeline for face representation, which first detected faces by MTCNN[17] from the video keyframes and query person examples, and then extracted a 4096-dimensional feature vector for each face by VGG-Face model[18]. However, we noticed that some of the detected faces were rather blurry due to low resolution and camera shake, which were difficult to distinguish and recognize. For addressing this issue, we employed image super-resolution (SR) to recover a high-resolution image from low-resolution image. Concretely, we followed the same setting as SRCNN[19] to pre-train an end-to-end mapping model between low-resolution and high-resolution images. Figure 3 shows an overview of SRCNN network structure.

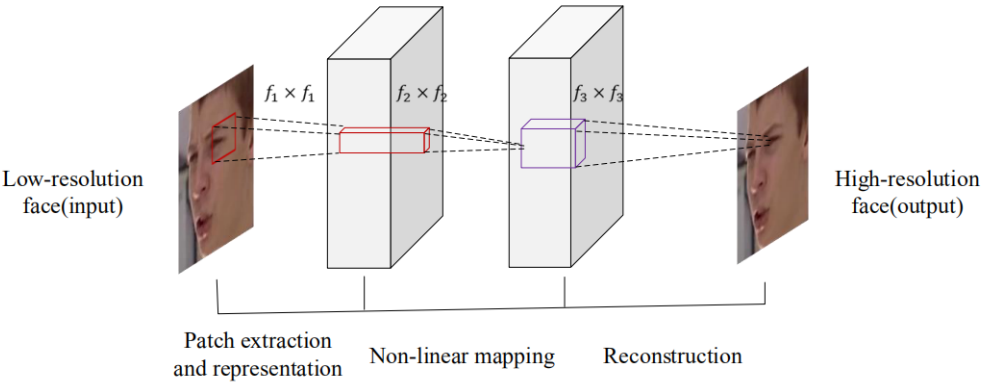
Low-resolution face(input)
$f_1 \times f_1$
$f_2 \times f_2$
$f_3 \times f_3$
High-resolution face(output)
Patch extraction and representation
Non-linear mapping
Reconstruction

Figure 3: The overview of Super-Resolution Convolutional Neural Network (SRCNN).

For a single low-resolution face image, we first upscaled it to the desired size (upscaling factor is set to be 2 here) using bicubic interpolation, and then the pre-trained SRCNN accepted the low-resolution image, and output the high-resolution one. Considering time consuming, we only performed face image super-resolution on the given query examples. When conducting the same processes on all detected faces from test video keyframes, it can further improve the retrieval performance.

2.2.2 Deep Face Recognition

As mentioned above, faces in the video keyframes were first detected by MTCNN[17], and then represented by 4096-dimensional feature vectors by VGG-Face model[18]. The above process was also performed for faces in query person examples. Basically, the similarity between a query face and a keyframe face was computed via cosine distance. However, for one person, not all query faces are actually helpful. We observed that there existed some “bad” faces in query examples, which caused significant decline of retrieval performance due to bad face orientation. Therefore, we detected and removed such “bad” faces by the following strategy: We considered the outlier of the 4 given query faces for a person as “bad” face, which was largely different from the other queries. Specifically, we calculated the similarity s_{ij} between i-th and j-th query face examples of one person, then we defined $c_i = \sum_{j \neq i} s_{ij}$ as the confidence score of the i-th query face. Finally the i-th query face would be detected as “bad” face if $c_i + \theta < \sum_{j \neq i} c_j / 3$, where θ was set to be 0.05 here.

We also adopted progressive training strategy for boosting the discrimination capability of the VGG-Face model. Concretely, we selected the faces in the top N (set to be 100 here) returned shots of each person to form the training set, which we used to fine-tune the VGG-Face model. Finally, we extracted a 4096-dimensional feature vector based on the fine-tuned model for each detected face, and calculated the cosine similarity cos_{ij} between the shot i and person j. The faces in the top N (set to be 200 here) returned were selected to train an SVM[20] model, and svm_{ij} denotes the classification score of predicting shot i as person j. We combined the similarity and classification scores to get the final person similarity as $PerSim_{ij} = cos_{ij} + svm_{ij}$.

2.2.3 Text-based Search

The strategies of text-based search were similar to our approach in last year[21]. As the transcripts of videos provided by NIST contain explicit cues to distinguish the speaker, they are usually complementary to visual information. We utilized transcripts to perform text-based search for each topic, where the person's information for search was extended by retrieving structured data from Wikipedia webpages, such as nick name, character name, names of the specific person's family and his/her closest friends, etc. For each topic, we generated a list of shots whose transcripts included the keywords of the topic. The retrieval results of text-based search were used to modify similarity scores during instance score fusion with a bonus mechanism. See Section 2.3 for details.

2.3 Instance Score Fusion

So far, we have the location similarities from the *location-specific search*, as well as the person similarities from the *person-specific search*. As INS task of this year required to find a given person in a given location, we adopted two-directional score re-ranking and fusion strategies to comprehensively consider the location and person similarities:

(1) We searched the candidate shots via *location* search, and then ranked them by *person* search. top-*N* (*N* > 1000) ranked shots were obtained according to location similarity score $LocSim$. We considered that these shots were likely to contain the given location. Then, a bonus mechanism for text-based person search results is proposed as follows:

$$s_1 = \mu \cdot PerSim \tag{3}$$

where μ is the text-based bonus parameter. $\mu > 1$ if the shot existed in text-based person search results, otherwise $\mu = 1$. By the text-based bonus mechanism, the shots whose transcripts contained the keywords of the query topic would gain higher similarity scores. Finally, the candidate location shots were re-ranked by s_1.

(2) We searched the candidate shots via *person* search, and then ranked them by *location* search. Similarly, top-*M* (*M* > 1000) returned shots ranked by $PerSim$ were selected as the candidate person shots, which had a considerable probability to contain the specified persons. We also employed text-based bonus mechanism to modify location similarity and got the score s_2 as following:

$$s_2 = \mu \cdot LocSim \tag{4}$$

Then we used s_2 to re-rank the candidate person shots.

(3) Moreover, in order to integrate the scores of location-based ranking and person-based ranking to further improve the search performance, we proposed an intuitive instance fusion strategy on s_1 and s_2. In this strategy, the fusion score of a shot would be calculated as:

$$s_f = \omega(\alpha s_1 + \beta s_2) \tag{5}$$

where α and β are weight parameters to balance s_1 and s_2, and ω is a bonus parameter. We set $\omega > 1$ if the shot simultaneously existed in the top-*N* location-specific results and top-*M* person-specific results, otherwise $\omega = 1$. The bonus parameter ω could help highlight the common shots of both location-specific and person-specific search results, which were more likely to be the right instances. In this way, the final instance scores can preserve information of both location and person aspects, for improving the instance search accuracy.

(4) The information of video is continuous, and the adjacent shots in time sequence usually contain closely related content. Therefore, we proposed a time sequence based re-ranking algorithm to refine the fusion results. Concretely, the instance fusion score of each shot was adjusted by its adjacent shots' scores as follows:

$$s_f^{(i)} = \sum\nolimits_{-T<k<T} s_f^{(i+k)} + \theta_k \tag{6}$$

where $s_f^{(i)}$ denotes the score of i-th shot and $s_f^{(i+k)}$ denotes the score of $(i+k)$-th shot in time sequence. k is the index difference between these two shots ($-T < k < T$), and θ is a parameter to adjust the similarity score. We used the adjusted scores to re-rank shots and got the final shot ranking list.

2.4 Semi-supervised Result Re-ranking

After the instance score fusion in the last process, we observed that although most of the top-ranked video shots were correct, a few noisy video shots still existed. These noisy shots tended to be largely different from the other top-ranked shots. Aiming to filter these noisy video shots, we proposed a semi-supervised learning based re-ranking algorithm to refine the ranking results obtained from the above fusion step. The overall of our algorithm is shown as Figure 4.

(1) Data matrix of 1000 top-ranked video shots *F* and *L* could be got from the introduced processes, where F_i denotes the deep feature vector for location of a keyframe, and L_i denotes the video shot ID of vector F_i, $i \in \{1, 2, \ldots, \mathrm{n}\}$ where *n* > 1000 means there were *n* keyframes from 1000 video shots.

(2) Initialized the affinity matrix W with all zeros, and updated it as follow:

$$W_{i,j} = \frac{F_i \bullet F_j}{|F_i| \bullet |F_j|}, i, j \in \{1, 2, \ldots, n\}, i \neq j \tag{7}$$

(3) Generated the k-NN graph:

$$W_{i,j} = \begin{cases} W_{i,j}, & F_i \in kNN(F_j); \\ 0, & otherwise. \end{cases} \tag{8}$$

where $kNN(F_j)$ denotes the set of k-nearest neighbors of F_j.

(4) Constructed the matrix: $S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where D is a diagonal matrix with its (*i*, *i*)-element equal to the sum of the *i*-th row of *W*.

(5) Iterated $G_{t+1} = \alpha S G_t + (1 - \alpha) Y$ until convergence, where G_t denotes the refined result in

t-th round and we set $G_0 = Y$. α is a parameter in the range (0, 1). Y is the final instance score ranking list of 1000 top-ranked video shots, and we set the score of each keyframe as same as its original video shot.

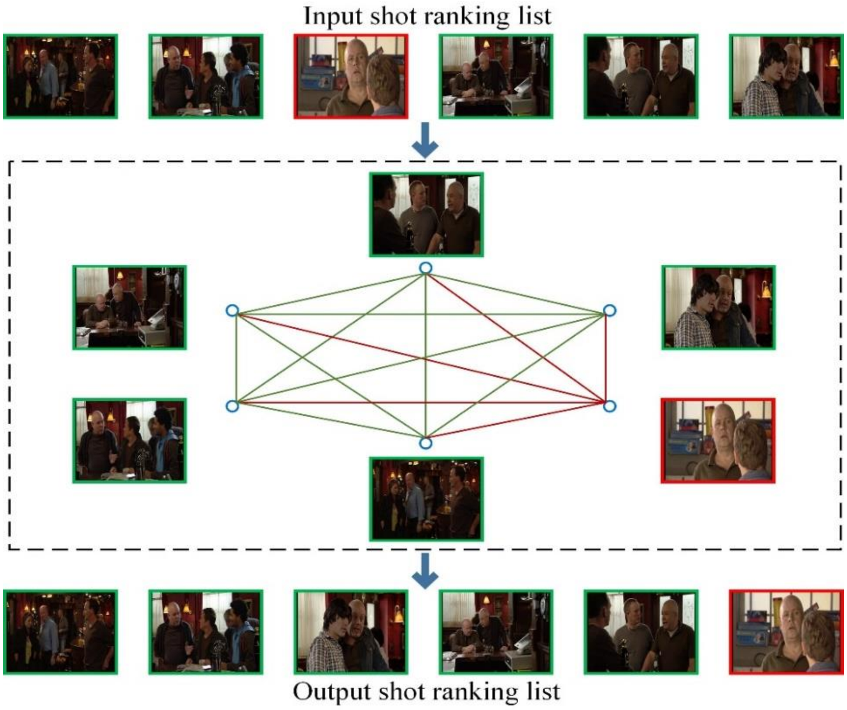

Figure 4: Our semi-supervised re-ranking algorithm. The shots with green rectangles mean right results, while red ones mean wrong results. Green edges in the graph mean high similarity between shots, while red ones mean low similarity.

3 Interactive Search

This year, we adopted a similar strategy as what we used in the interactive search task of INS 2017[21]. The interactive search was based on RUN2_E, without result re-ranking stage. First, the user labeled positive or negative samples for each topic's top-ranked results in the automatic search ranking list. Next, we regarded the positive samples as expanded queries to conduct the location and person search. For efficiency, we only selected 10 positive samples in each topic for interactive search. Finally, we merged the scores of expanded and original queries to get merged score list, where the negative samples were discarded.

4 Conclusion

By participating in the INS task in TRECVID 2018, we have the following conclusions: (1) Enhancing query image quality is helpful to achieve better performance, as is indicated by supersolution preprocessing and “bad” faces removal. (2) Video examples are still helpful for accuracy improvement, which can provide more information to achieve higher results (see Table 1). (3) Human feedback is very useful to boost the accuracy of INS task.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant 61771025. For the using of BBC EastEnders video or images snapshots, we thank for the programme material copyrighted by BBC.

References

[1] G. Awad, A. Butt, K. Curtis, *et al.*, “TRECVID 2018: Benchmarking Video Activity Detection, Video Captioning and Matching, Video Storytelling Linking and Video Search”. *Proceedings of TRECVID 2018*, 2018.

[2] G. Awad, W. Kraaij, P. Over, *et al.*, “Instance search retrospective with focus on TRECVID”, *International Journal of Multimedia Information Retrieval*, vol. 6, no. 1, pp. 1-29, 2017.

[3] C.G.M. Snoek, K.E.A. van de Sande, O. de Rooij, *et al.*, “The MediaMill TRECVID 2008 Semantic Video Search Engine”. *TRECVID*, Maryland USA, November 17-18, 2008.

[4] K. Mikolajczyk, and C. Schmid, “Scale and Affine Invariant Interest Point Detectors”. *International Journal of Computer Vision (IJCV)*, vol. 60, no. 1, pp. 63-86, 2004.

[5] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions”. *British Machine Vision Conference (BMVC)*, pp. 384-393, 2002.

[6] D. G. Lowe, “Distinctive Image Features from Scale-invariant Keypoints”. *International Journal of Computer Vision (IJCV)*, vol. 60, no.2, pp. 91-110, 2004.

[7] K. Mikolajczyk and C. Schmid, “A Performance Evaluation of Local Descriptors”. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 27, no.10, pp. 1615-1630, 2004.

[8] Peng Y, He X, Zhao J. Object-Part Attention Model for Fine-Grained Image Classification. *IEEE Transactions on Image Processing (TIP)*, 2018, 27(3): 1487-1500.

[9] Peng Y, Zhao Y, Zhang J. Two-stream collaborative learning with spatial-temporal attention for video classification. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2018.

[10] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2014: 580-587.

[11] Girshick R. Fast r-cnn. *In Proceedings of the IEEE international conference on computer vision (CVPR)*. 2015: 1440-1448.

[12] Peng Y, Qi J, Huang X, et al. CCL: Cross-modal Correlation Learning With Multigrained Fusion by Hierarchical Network. *IEEE Transactions on Multimedia (TMM)*, 2018, 20(2): 405-420.

[13] Huang X, Peng Y. Deep Cross-media Knowledge Transfer. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018: 8837-8846.

[14] K. Simonyan, A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, *International Conference on Learning Representations (ICLR)*, 2015.

[15] C. Szegedy, W. Liu, Y. Jia, *et al.,* “Going deeper with convolutions”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9, 2015.

[16] B. Zhou, A. Khosla, A. Lapedriza, *et al.,* “Places: An image database for deep scene understanding”, arXiv preprint arXiv:1610.02055, 2016.

[17] K. Zhang and Z. Zhang and Z. Li, *et al.*, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks”. *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.

[18] O. M. Parkhi, A. Vedaldi, and A. Zisserman. “Deep Face Recognition”. *British Machine Vision Conference (BMVC)*, pp. 41.1-41.12, 2015.

[19] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang. Image Super-Resolution Using Deep Convolutional Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016, 38(2): 295-307.

[20] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines”. *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1-27, 2011.

[21] Y. Peng, X. Huang, J. Qi, *et al.*, "PKU-ICST at TRECVID 2017: Instance Search Task", *TRECVID*, Gaithersburg, MD, USA, Nov. 13-15, 2017.