ORANGE LABS BEIJING(FTRDBJ) AT TRECVID 2013: INSTANCE SEARCH

Hongliang Bai[†], Yuan Dong[‡], Shusheng Cen[‡], Lezi Wang[‡], Lei Liu[‡], Wei Liu[†], Yunlong Bian[‡]
Chong Huang[‡], Nan Zhao[‡], Bo Liu[‡],Yanchao Feng[‡], Peng Li[†], Xiaofu Chang[†], Kun Tao[†]

[†]Orange Labs International Center Beijing, 100013, P.R.China
[‡]Beijing University of Posts and Telecommunications,100876, P.R.China
{hongliang.bai,wei.liu, peng.li, xiaofu.chang, kun.tao}@orange.com
{yuandong,censhusheng, wanglezi,liulei}@bupt.edu.cn

ABSTRACT

The framework of TRECVID [7] Instance Search (INS) 2013 task is introduced by Orange Lab Beijing. One interactive and three automatic runs have been submitted, namely:

F_X_NO_FTRDBJ_1: SIFT feature, Lucene-based search and face recognition

F_X_NO_FTRDBJ_2: SIFT feature and Lucene-based search

F_X_NO_FTRDBJ_3: Learned feature by Convolution Neural Networks (CNNs)

I_X_NO_FTRDBJ_4: SIFT feature, Lucene-based search and random walk based relative feedback

After experiments in 2013 TRECVID INS dataset, mAP performance of above four runs are 0.1934, 0.1775, 0.0155 and 0.2957 respectively. The interactive run is better than the automatic runs. It is consistent with our experience.

Keywords

TRECVID, Instance Search, Face Recognition, Convolution Neural Network

1. INTRODUCTION

The large number of TV serials are broadcast by TV stations every day. The requirements for searching some special videos are more and more strong because the videos are delighting, instructive or useful. The search topics are related with objects, places , persons and so on. The object/person detection and recognition are the basis of the video analysis.

Many state-of-art methods or algorithms have been proposed to meet with the above requirements in the recent years. They mainly include feature extraction, feature encoding, video search and reranking. In the feature extraction stage, the local feature is most frequently used. Its extraction basically has two steps; one is feature detectors, such as Harris detector, Harris Laplace detector, Hessian Laplace, Harris/Hessian Affine detector, and the other is feature descriptors, such as Scale Invariant Feature Transformation (SIFT), Shape Context, Gradient Location and Orientation Histogram, Speeded Up Robust Features(SURF), DAISY. Then Bag-of-Visual-Word and inverted table framework is widely used in feature encoding and search. Apache Lucene is a high-performance, full-featured text search engine library, and we use it in image search successfully.

The performance of video search is heavily dependent on the choice of data representation (or features) on which they are applied. Above features are hand-crafted and need expert knowledge. Learned features have been successfully used in object recognition, face recognition, NLP, especially in speech recognition. Deep convolutional neural networks achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry in ILSVRC-2012 competition.

In the TV serials, person retrieval can be implemented by face recognition. The Face verification and face identification are two sub-problems in face recognition. The former is to verify whether two given faces belong to the same person, while the latter answers "who is who" question in a probe face set. Given a face image, we have to identify who is in the image. The images may vary in pose, expression, lighting, occlusions, image quality, etc. The difficulty lies in teasing out features in the image that indicate identity and ignoring features that vary with differences in environmental conditions. Face recognition is often naturally described as part of a Detection-Alignment-Recognition (DAR) pipe-line.

Interactive instance search is attractive because it can make up the low performance of automatic instance search. The random walk based algorithm is used in our system, which is the same with last year. The more correct results are feed back in the limit time by the lucent based search. The details have been described in our 2012 TRECVID paper.

Our contributions in 2013 INS are follows, (1)face recognition is used in the search when the face region of query is detected, and it can improve the search performance, (2)

the learned features by deep CNNs are used in image search. Though its performance is not satisfying, it will be refined further, (3)random walk based algorithm achieves the best performance on the interactive INS task. The rest of paper will introduce our work to implement the INS task. Section 2 demonstrates object retrieval framework based on bag-of-word model. The face recognition and deep learning algorithms are described in Section 3 and 4. The experiments and discussion are in the Section 5. Finally, we will propose the future work to improve the current performance.

2. OBJECT RETRIEVAL BASED ON BAG-OF WORD MODEL

2.1 Feature Extraction

Local feature based on scale-invariant keypoint, has already been shown to be effective in multiple computer vision tasks, one of which is object retrieval. We use hessian-affine detector [6] to detect keypoints and extract C-SIFT [9] descriptor to represent the local geometry of these keypoints. The dimension of resulting feature is 192.

Dataset of this year's instant search task comprises high definition H264 videos. The raw resolution of frame extracted from video is 768x576. Though more details of object can be drawn from these high resolution frames, the large amount of features also puts a heavy burden on computational complexity. For the aforementioned size, there are as many as 4,000-5,000 local features extracted from a single image. For another thing, at raw resolution, there would be interlaced artifact on moving objects and this would introduce a lot of noise. Due to these reasons, a lower resolution is preferred. We downsize the raw frame to a resolution of 384x288. At this resolution, an average of 1000 local features is extracted per frame.

2.2 Codebook Training

Visual codebook plays an important role in large-scale object retrieval. As shown in many early research works, the retrieval precision can benefit from larger size of visual codebook. Approximate k-means (AKM) [8] is adopted in our system. Instead of brute-force Nearest Neighbor (NN) search in k-means, AKM uses randomized k-d trees to perform approximate nearest neighbor search, which makes it possible to train very large codebook in reasonable time. Total number of features extracted from frames is about 1.5 billion. A subset consisting of 75 million features is selected randomly from the whole features as training data for a 1M codebook. In our implementation, the open-source library FLANN [1] is used for fast approximate nearest neighbors search.

Although efficiency and effectiveness of AKM is well recognized, there is still some disadvantages for AKM. The most important one is non-robustness caused by inaccuracy of approximate NN search. Under typical settings (128 checks in 4 randomized k-d trees for each NN search), the precision of NN search is around 0.5 . A simple but effective technique is proposed in [5] to improve the robustness of AKM. The main idea of this technique is that the assignments in the previous iterations can be used to verify the NN search result in current iteration, which can make the NN search result more robust. This technique is employed in our AKM implementation.

For training our 1M codebook on a machine with a 8-core Xeon E5-2643 CPU and 16GB RAM, time cost is around 8 hours. the speed-up is not only contributed by fast NN search, but also benefit from the fast convergence brought by robust checking.

2.3 Indexing Using Lucene

With the trained 1M codebook, we can encode each 192-dimensional CSIFT descriptor into one of the codes ranging from 1 to 1000000. As in standard bag-of-word implementation, Vector Space Model (VSM) is used to compare the similarity between two images. The cosine distance of vectors after tf-idf weighting is computed as similar score.

Similar with text-based document retrieval, the sparsity of the feature vector enables speeding up search by using inverted index. Apache Lucene [2] is an outstanding text search software. It is open-source and has been deployed in many large web-site as search engine. Without much effort, Lucene can be adapt to perform efficient indexing and fast searching on visual words.

For simplicity, descriptors are encoded into codes by hard quantization. Thus, an image can be translated to a text document of codes. The number of codes is equal to the sum of CSIFT descriptors. We use Lucene's default setting to index the documents, tokenizing and putting words into index structure. The data structure of inverted table in Lucene is well-designed for building and retrieving large amounts of documents. Building index for our 1.5 million documents, which contains as many as 1.5 billion visual words, takes less than one hours. And the disk consumption is 6 GB, indicating an average of 4 bits per visual word.

The searching using Lucene is also very convenient. First, visual words of query image is extracted. Then, a boolean query can be created by combining visual word as boolean clauses. VSM and tf-idf weight are implemented by Lucene and we just need to invoke the corresponding function. The similar scoring function used by Lucene is

$$Score(q,d) = coord(q,d) * queryNorm(q) * \sum_{t\ in\ q} (tf(t) * idf(t) * norm(d)) \quad (1)$$

where $queryNorm(q)$ is a normalizing factor used to make scores between queries comparable, $coord(q,d)$ is a score factor based on how many of the query terms are found in the specified document, and $norm(d)$ is a normalizing factor to balance the inequality caused by non-uniform length of the indexed documents.

With Lucene's highly efficient data structure, fast search can be perform in very large index containing millions of documents. For a typical query consisting of 1000 visual words, searching time on our index is around 1 second and RAM consumption is 3 GB. This performance is better than most system mentioned in previous works.

[1] http://www.cs.ubc.ca/research/flann/

[2] http://lucene.apache.org/

2.4 Object Saliency

The target object of instant search is indicated by mask given with query image. Traditionally, only features raised from region-of-interest (ROI) of query image are reserved for searching and the features raised from background area considered as noise. However, in some topics, the target always appears in a certain scene and the features raised from background is helpful to identify the scene. Especially when the target is a tiny object, which means few local feature is extracted in ROI, background becomes crucial cue to find the image containing the target. As illustrated in Fig. 2, the search results are terrible by only utilizing the feature in ROI. However, using the whole image for query is not preferable. When the target is not large enough and the background dominates the image, most of the extracted features are raised from background and these features may put a preference on background.

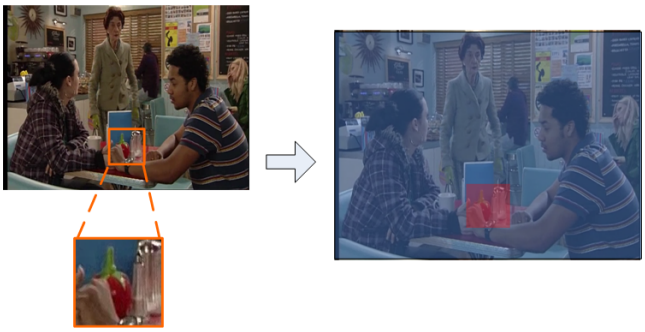


Figure 1: Weighting scheme for integrating target object with the background

These cases motivated us to integrate background with object features. Together object features, we introduce background features for query through a saliency weighting scheme. Obviously, object features is much important than background features. So, heavier weight should be added on object features to emphasize the target region.

In our practice, we over-sample the object features by duplicating them for 3 times in query image, which has the same effect of putting large weight on them, as illustrated in Fig. 1. From the results shown in Fig. 2, we can found that, over-sampling features in ROI combines information from target object and background and yields better result than the other two schemes.

3. FACE RECOGNITION

In our system, Viola-Jones classifier is used to detect faces in the given images [10]. Then, five facial landmarks (eye centers, nose tip and mouth corners) are located with simplified Deformable Part Model(DPM) algorithm [3]. At last, twelve different components are aligned separately based on the five key-points. In our approach, instead of comparing the visual features extracted from the face regions directly, we use simile classifiers trained based on reference datasets to build the high-level face descriptor.

3.1 Simile Classifier based Face Descriptor

The basic idea of our face description method is to use attribute classifiers trained to recognize the presence or absence of describable aspects of visual appearance (e.g., gender, race, and age). And to removes the manual labeling required for attribute classification and instead learns the similarity of faces, or regions of faces, to specific reference person. The similarities should be insensitive to environment variations. To train simile classifiers, we first have created a dataset of 128 reference persons. The dataset consists of 28984 images. The largest person has 720 images while the smallest one has 41 images. Some images of the dataset are from PubFig dataset (49 persons), while the remainders are crawled from Baidu Images (79 persons).

Figure 2: Examples of results generated by different weighting scheme. (a) query with object features only; (b) weights ratio of object and background is 3:1 ; (c) object share a same weight with background.

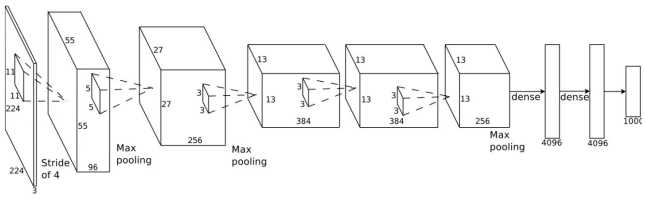
3.2 Face Classifier

In each component image, three representative low-level descriptors (uniform LBP[1], global LBP and gradient LBP) can be obtained. These descriptors are fed into 1000 best simile classifiers which are selected in the training process. The output value of each classifier is concatenated to construct 1000D high-level descriptor.

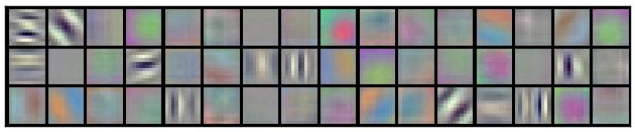
For the targeted person, we train the specialized classifier using five targeted faces as positive samples and fifty non-targeted faces as negative samples. And the classifier is RBF support vector machine trained using the LIBSVM package. All test face images are fed into the specialized classifier and the output values server as the similarity between test faces and the targeted person's face.

4. LEARNED FEATURE BY CNNS

The deep CNN and large number of 2D convolution operations on GPU framework achieved best top-5 test error rate of 15.3%, compared with 26.2% of second-base [4]. The similar deep structure is implemented as Hinton's Lab, shown in Fig.3. The final convolution layer output is used as the image feature. The dimension feature is $6 \times 6 \times 256 = 9216$. Then the original 9216D features of querying and reference dataset are generated.



(a) CNN deep structure, from Hinton's Lab PPT in ILSVRC-2012



(b) learned 48 filters on the first convoution layer

Figure 3: Learn feature by CNNs

5. EXPERIMENTS

5.1 Database Description

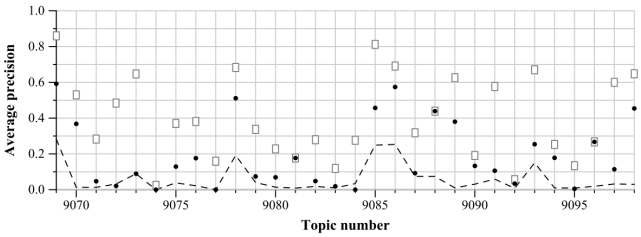
TRECVID is a laboratory-style evaluation that attempts to model real world situations or significant component tasks involved in such situations. The INS task is to find more video segments of a certain specific person, object, or place, given a visual example. In 2013, 30 topics are included, in Fig.4. four persons and 26 objects/logos are queried in the 244 BBC EastEnders videos in MPEG-4 format.

Figure 4: Query Image Samples

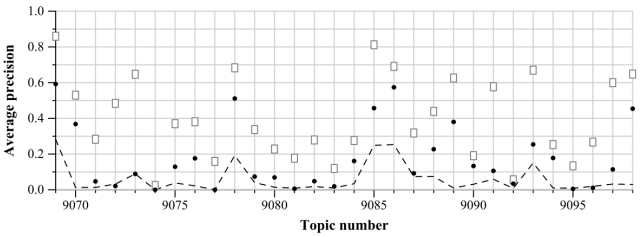
5.2 Automatic INS Performance

The automatic run results are shown in Fig. 5. The face recognition algorithm improved the search performance in some of person-related query. Take topic 9088 as examples, the mAP is improved from 0.2265 to 0.4387. But the time performance is very bad with the exhaustive search. Totally, mAP of **F_X_NO_FTRDBJ_1** is 0.1934 , which is best in all three automatic runs. but run **F_X_NO_FTRDBJ_2** is more attractive because the balance beteen the precision and time preformance. In run **F_X_NO_FTRDBJ_1**, mAP of topic 9074 (a cigarette), 9077(this dog) and 9095(a green public trash can) are closed to zero because of less features. Our face recognition can not recognize the profile or blurring faces, such as topic 9084(this man).

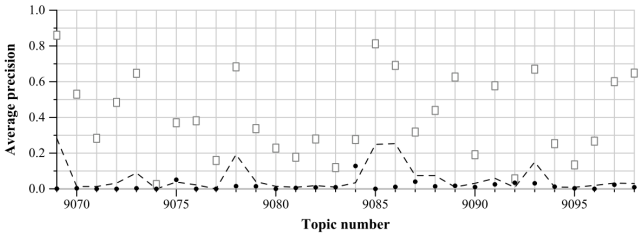
For run **F_X_NO_FTRDBJ_3**, the deep structure of CNNs is learned in the imagenet 2011 dataset and used in the task. The features focus on the global case. But the INS query topic is more related with local feature. The Learned features are not suitable for INS cases. The mAP of this run is only 0.0155.



(a) run **F_X_NO_FTRDBJ_1**



(b) run **F_X_NO_FTRDBJ_2**



(c) run **F_X_NO_FTRDBJ_3**

Figure 5: Three Automatic INS Runs

5.3 Interactive INS Performance

The interactive runs results are shown in Fig. 6. The mAP of this run is 0.2957, which is the best run in all submitted interactive runs. The quick search engine makes more times feed back possible.

6. CONCLUSION

BOW and inverted table framework is still the basic framework of our INS system. But the SIFT feature is not suitable for the small or less-texture objects, such as cigarette, BMW logo and pendant. Some objects are detected by the their

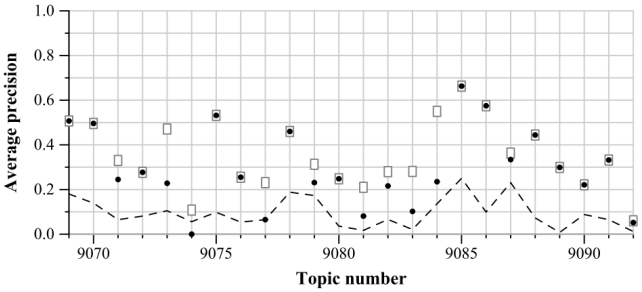


Figure 6: run I_X_NO_FTRDBJ_4 for Interactive INS

existing scene. SIFT and LBP are also hand-crafted feature. In the next year, the learned features will be developed and used to solve the above difficulties.

7. ACKNOWLEDGMENT

We want to thank programme material ©BBC[2] for supplying us with the audio, visual and metadata material from 'EastEnders' television programme.

8. REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. In *ECCV*, pages 469–481, 2011.

[2] BBC. http://www.bbc.co.uk/programmes/b006m86d.

[3] P. F. Felzenszwalb, D. A. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[4] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[5] D. Li, L. Yang, X.-S. Hua, and H.-J. Zhang. Large-scale robust visual codebook construction. In *Proceedings of the international conference on Multimedia*, pages 1183–1186. ACM, 2010.

[6] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

[7] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Queenot. Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[8] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[9] K. E. Van De Sande, T. Gevers, and C. G. Snoek. Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1582–1596, 2010.

[10] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

The Orange Labs International Center China (FTRDBJ) Video Semantic Indexing Systems – TRECVID 2013 Notebook Paper

Kun Tao[1], Yuan Dong[2], Yunlong Bian[2], Xiaofu Chang[1,2], Hongliang Bai[1], Wei Liu[1], Peng Li[1]

[1]Orange Labs International Center China, Beijing, 100013, P.R.China
[2]Beijing University of Posts and Telecommunications, Beijing, 100876, P.R.China
kun.tao@orange.com yuandong@bupt.edu.cn

ABSTRACT

In this paper, we introduce our systems and experiments in TRECVID2013 SIN task. This year, we submitted 3 main task runs with localization results and 2 concept pair runs including one pair baseline. For main task, we tried two new features: sparse coding dense-sift and OLBP. They can be used to boost the system performance by late fusion with our original system. We also trained 10 object detector based on DPM algorithm to accomplish the location subtask. The detectors showed good precision and slightly low recall in final evaluation.

1. INTRODUCTION

In 2013, the semantic indexing task reduced the concept number to 60, which will help us to make more in-depth and continuous research. Some new features were tested in our experiments. Although the single feature performance of them can't go beyond our 9-feature composite kernel classifier, they can contribute by late fusion with the 9-feature results [1]. The DPM [2] object detection was also evaluated in our experiments. Finally, we submitted 3 main task runs with corresponding localization results, one concept pair run using confidence and one baseline run. The basic information of submitted runs is shown below:

- 13_M_A_FTRDBJ-M1: Using composite-kernel SVM with 9 features. MAP = 0.189.
- 13_M_A_FTRDBJ-M2: late fusion of run1+spc+OLBP. MAP = 0.196.
- 13_M_A_FTRDBJ-M3: run2 + object detection. MAP = 0.187.
- 13_P_A_FTRDBJ-P1: Fusion by confidence. MAP = 0.105.
- 13_P_A_FTRDBJ-Pb: Pair Run Baseline. MAP = 0.084.

2. SEMANTIC INDEXING SYSTEM

2.1 Key-frame Selection

For IACC.2 database used in 2013 SIN task, all I frames were provided to facilitate the experiments of localization subtask. Thus we tried to find representative key-frames of the shots using an intra-shot clustering on I frames. For all I frames in one shot, the images were resized to 100*100 gray image and 64D features were extracted by a fast DCT algorithm. Then an automated thresholding agglomerative hierarchical clustering [3] was used to find 1 to 3 RKF and NRFKs for each shot.

2.2 Main task runs

Our original system based on 9-feature composite kernel classifier has been proved to be valuable. It was used as a baseline in our experiments. We also submitted our first run 13_M_A_FTRDBJ-M1 based on it. Then we mainly focused on testing some new features including sparse coding based features and OLBP feature. These new features' dimensions are quite high. Thus they can't be added to composite kernel early fusion system. They were evaluated separately and then fused with the results of the baseline to boost the performance of final system.

Our sparse coding features are based on the super-vector coding described in [4,5]. First the descriptors of dense-sift or dense-opsift [6] are extracted, and a k-means based codebook $C=[c_1,\ldots,c_p]$ is prepared.

For each descriptor z, find the nearest index $i=\arg\min_j \| z-c_j \|^2$. Then get the vector quantization coding $r\in R^p$, whose i-th element is one and others are zero. The SV coding result $\beta=[(r_1 s, r_1(z-c_1)),\cdots,(r_p s, r_p(z-c_p))]$, where $\beta\in R^{(d+1)p}$ and s is a predefined small costant.

Let $Z=[z_i]_{i=1}^n$ be the set of descriptors of an image and $[\beta_i]_{i=1}^n$ be their SV codes. Give the p code groups the weights w_k ($\sum_{k=1}^p w_k=1$) proportional to the numbers of z_i belong to them. The pooling result for image is

$x = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{w_{\delta(i)}}} \beta_i$. $\delta(i)$ Indicate the index of group z_i belongs to.

As the final feature, the dimension of pooling result is quite large. For dense-opsift, we use 512D codebook and the feature dimension is 197120. For dense-sift, the codebook size is 256 and the features are extracted on 4 sub-regions with the total dimension 132096. Considering the training cost, the linear SVM classifier is selected to train our models.

The OLBP feature extends the original LBP by a new coding strategy. Given a central point g_c in the image, and its P circularly and evenly spaced neighbors g_p, p=0,1, ⋯ ,P-1 , we can simply calculate the difference between g_c and g_p as $d_p=g_p-g_c$. Using local difference vector $f=[d_0, \cdots, d_{P-1}]^T$ as a descriptor of local region centered on g_c and then accumulate them across the whole image into one of 2^P bins in the following way:

$$H = \sum_{all\ g_c} [bin_0(f)f, \cdots, bin(f)_{2^P-1} f]$$

$$bin_i(f) = \begin{cases} I & \text{,f belongs to i-th bin} \\ [0] & \text{,otherwise} \end{cases} \qquad bin_i \in R^{P*P}$$

So the final dimension of histogram H is 2^P*P. In practice we use uniform LBP (59D patterns) with 8 points and 4 different radiuses, and the descriptors are extracted in 3 channels of opponent color space. The dimension of the final feature is 5664. The OLBP features are also sent to a linear SVM for classification.

In our experiments, the performances of the classifiers based on one single feature out of above 3 can't go beyond the performance of 9-feature early fusion. But when we combine their results by late fusion, the system performance can be improved. Our second run 13_M_A_FTRDBJ-M2 is based on the late fusion of run1+dense-sift sparse coding+ OLBP.

Our 3rd run is based on re-ranking with the object detection results. For each concept their top 10000 shots found by run2 are selected as candidates. Then we use DPM models to detection the objects in their key-frames. If a DPM model returned one positive score at least, the highest score will be added to original score and the re-ranking results will be used for our run 13_M_A_FTRDBJ-M3.

2.3 concept pair runs

This year, our concept pair run kept using the confidence based weighted average which has been proved in 2012 evaluation. The single concept results for fusion came from our 13_M_A_FTRDBJ-M1 run. A baseline using simple average was also submitted.

3. OBJECT LOCALIZATION SUBTASK

This year, a new challenging subtask was added. Participants were asked to find the temporal-spatial location of the target objects. All I frames belong to top 1000 shots should be checked and the rectangles of the objects should be submitted. To accomplish this task, the deformable part model (DPM) is used, which has been fully proved in PASCAL VOC evaluation [7].

For the 10 concepts selected for localization, we selected their positive samples in IACC.1 corpus and labeled the object rectangles for training. Discarding some redundant and invalid images, hundreds to thousands samples were labeled for each concept and the corresponding DPM models were trained. For all top 1000 shots of 10 selected concepts in 3 main task runs, their I frames were detected by DPM and the resulting bounding boxes were submitted.

4. EXPERIMENT RESULTS

4.1 Experiments for new features

This year, some new features were tested in our experiments: sparse coding dense-sift, sparse coding dense-opsift and OLBP. Using linear SVM classifier, the comparison of single feature performance is showed below:

Table 1. Single Feature MAP

Feature	Dimension	MAP
dense-sift spc (dss)	132096	0.168
dense-opsift spc (doss)	197120	0.157
OLBP	5662	0.124
9-feature composite kernel(9f)	8916	0.28

Above MAPs were calculated on 25 concepts selected from all 60 concepts. The dimensions of sparse coding features are so high that we need enough storage space to save the extracted feature. But using linear classifier, the training and testing processes are very fast, while the composite kernel models are time-costing.

Then we tried to combine above results by late fusion. The experiment results of different combinations are shown below:

Table 2. Late fusion MAP

Combination	Fusion Weights	MAP
dss+OLBP+9f	Liblinear	0.291
dss+OLBP+9f	Single MAP as weight	0.284
dss+9f	Single MAP as weight	0.28

OLBP+9f	Single MAP as weight	0.282
dss+OLBP	Single MAP as weight	0.28

Our 2nd run is based on dss+OLBP+9f and Liblinear is used to train the weights. The MAP in final evaluation is 0.196, which is beyond the result of our 1st run using 9f only (0.189).

4.2 Experiments for DPM

There are 10 concepts selected for localization subtask: Airplane, Boat_Ship, Bridges, Bus, Chair, Hand, Motorcycle, Telephones, Flags, Quadruped. For each concept, we labeled the training data on IACC.1 and trained DPM models. By evaluation on training set, the thresholds of DPM models were re-estimated.

In our 3rd run, we used DPM to detect objects in top 10000 shots found by our 2nd run. But the final result show that the precision of DPM is not as good as we expected. 10000 seems to be too big for this task, while the weights given to DPM in re-ranking is too high. So many false alarms were raised to the top of ranking list, which caused the decline of the MAP (from 0.196 to 0.187).

In localization subtask, as we used relatively high thresholds, the frame level and pixel level precision is quite good. But the recalls are too low that the final fscores are also unsatisfactory. The results of our localization systems are shown below:

Table 3. Localization System

	Run1	Run2	Run3
iframe_fscore	0.1512	0.1437	0.1212
iframe_precision	0.1863	0.1798	0.1259
iframe_recall	0.2238	0.1958	0.229
mean_pixel_fscore	0.0967	0.0918	0.0673
mean_pixel_precision	0.1194	0.115	0.0828
mean_pixel_recall	0.0953	0.0899	0.0661

5. CONCLUSION

In recent years, many new feature extraction and coding methods were published. This time we tried to add these new techniques to our SIN systems. Although not so significant, some improvement has been shown in our experiments. In the future, we wish to further explore the potential of these new techniques. We also wish to try some new frameworks such as deep learning in the near future.

6. REFERENCES

[1] K. Tao, etc. “The France Telecom Orange Labs (Beijing) Video Semantic Indexing Systems – TRECVID 2012 Notebook Paper,” http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.12.org.html, 2012.

[2] P. F. Felzenszwalb, etc. “Object Detection with Discriminatively Trained Part Based Models”, PAMI 2010.

[3] R. Sibson. "SLINK: an optimally efficient algorithm for the single-link cluster method" The Computer Journal 1973.

[4] X. Zhou, K.Yu, etc. “Image classification using super-vector coding of local image descriptors.” In ECCV 2010.

[5] Y.Q. Lin, etc. “Large-scale image classification: Fast feature extraction and SVM training”, in CVPR 2011

[6] Koen, etc. “A Comparison of Color Features for Visual Concept Classification”, CIVR 2008.

[7] http://pascallin.ecs.soton.ac.uk/challenges/VOC/