CHALLENGE HUAWEI CHALLENGE: FUSING MULTIMODAL FEATURES WITH DEEP NEURAL NETWORKS FOR MOBILE VIDEO ANNOTATION

Jian Tu, Zuxuan Wu, Qi Dai, Yu-Gang Jiang, Xiangyang Xue

School of Computer Science, Fudan University, Shanghai, China
ygj@fudan.edu.cn

ABSTRACT

We participated in the Huawei Accurate and Fast Mobile Video Annotation Challenge (MoVAC) at IEEE ICME 2014. Three result runs were submitted by combining different features and classification techniques, with emphasis on both accuracy and efficiency. In this paper, we briefly summarize the techniques used in our system, and the components used for generating each of the three submitted results. One novel component in our system is a specially tailored deep neural network (DNN) that can explore the relationships of multiple features for improved annotation performance, which is very efficient based on an implementation with the GPU. Only 18.8 seconds were needed by one of our DNN-based submissions to process a test video. By combining the DNN with the traditional SVM learning, we achieved the best accuracy across all the worldwide submissions to this challenge.

***Index Terms*—** Video annotation, deep neural network, multimodal features.

1. INTRODUCTION

Recent years have witnessed an explosive growth of user-generated videos, mostly captured by hand-held mobile devices. There is an urgent need to developing automatic video annotation techniques, which can be deployed in many applications, such as personal video collection management and large scale video search.

In addition to the long-standing semantic gap that has posed a big challenge for automatically recognizing video contents, the extremely large scale of video databases in most applications also creates difficulties for many existing techniques. Extensive studies have been conducted to improve the recognition performance. Most existing works emphasized on designing effective features [1], multi-feature fusion methods [2], or novel recognition algorithms [3], while very few of them focused on improving the speed efficiency [4].

The MoVAC Challenge proposed by Huawei Technologies emphasizes both accuracy and efficiency of a recognition system, to ensure that the proposed techniques can be deployed in realistic applications. In this paper, we describe a comprehensive system to tackle this interesting and important challenge. Several features are extracted from both the visual and audio channels of the video data, which are then fed into learning algorithms for content annotation. For the learning methods, we consider both the traditional SVM classifier, and the recently emerging deep learning techniques. In particular, we developed a deep neural network (DNN) with special design to better explore feature relationships for improved annotation performance.

Fig. 1 gives the framework of our system. First, audio/visual frames are sampled from an input video. After that, several popular features are computed and used as inputs of both the SVM and the DNN models. Finally, the outputs of the classifiers are used in a post-processing step to generate frame-level video annotations.

By using different subsets of the system components, we submitted three result runs, which are summarized as follows:

1. Run-1 (DNN Fast) is based on a regularized deep neural network, as illustrated in Fig. 2, which has very fast testing speed. Because this run emphasizes more on speed, only two very efficient features are adopted, following [4].

2. Run-2 (SVM Strong) is purely based on the SVM classifiers, which have been popular for years. All the extracted features are employed, in order to optimize recognition accuracy.

3. Run-3 (DNN Strong + SVM Strong) is a combination of both the DNN and the SVM models. By harnessing both kinds of learning methods, we expect to generate strong accuracies. The Fisher Vectors are not used for the DNN due to the high dimensions of the features, which incur too many parameters to be well learned.

Among the three runs, Run-1 focuses more on the speed but also selects features carefully to maintain a good recognition accuracy. Run-2 and Run-3 emphasize on accuracy by using more features and classifiers.

The rest of this paper is organized as follows. Section 2 elaborates all the technical components adopted in our system. Section 3 presents the experimental results on both the validation set and the official test set. Finally, Section 4 concludes this paper.

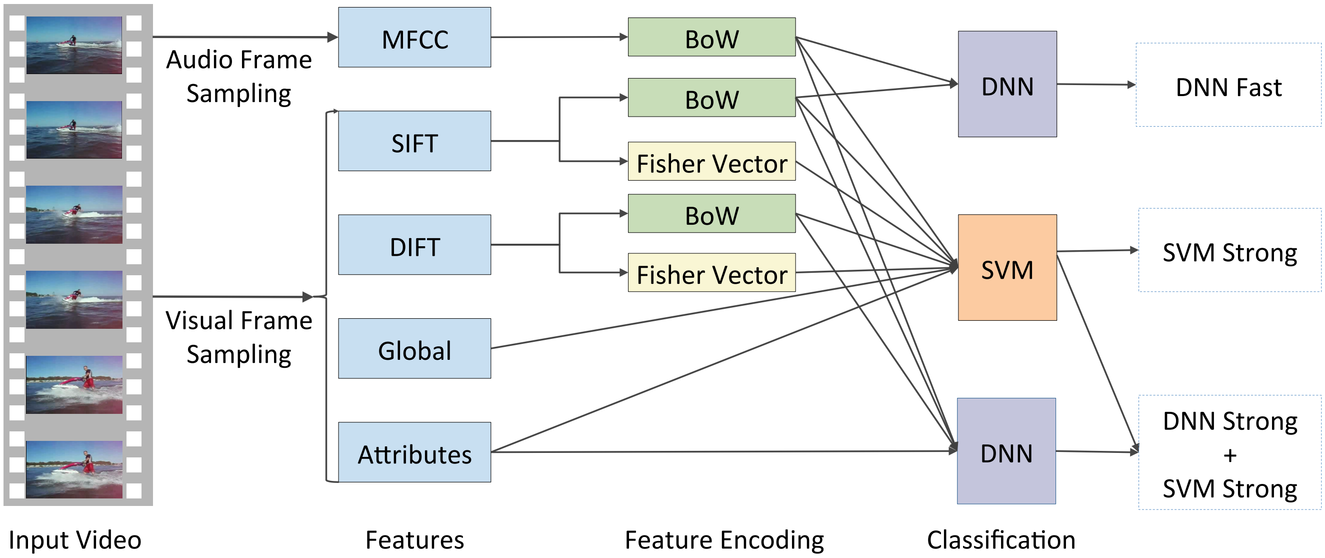

Fig. 1. The framework of our proposed video annotation system. Audio and visual frames are first extracted, on which five kinds of features are computed. For MFCC, DIFT and SIFT, both Bag-of-Words and Fisher Vector are used to produce quantized feature representations. DNN and linear SVM are adopted to classify the features. We submitted three runs using different combinations of the components, emphasizing either on speed (Fast) or on accuracy (Strong). See texts for more explanations.

2. TECHNICAL APPROACH

In this section, we elaborate the technical details of our system. As shown in Fig. 1, we followed a standard visual recognition pipeline, where features are first extracted and encoded, which are then used as the inputs of classifiers for recognition/annotation.

2.1. Frame Sampling

For visual frames, we sampled one frame from every two seconds of video sequence on the training samples. Further increasing the sampling density may improve the results but will incur significant additional training time. For the test videos, we sampled one frame per second to increase the annotation accuracy. All of the visual features discussed later were extracted from the same set of frames.

For the audio channel, we sampled audio frames densely from the soundtracks. The temporal width of a frame is 32ms and nearby audio frames have 16ms overlap. The MFCC features were computed on these audio frames.

2.2. Feature Extraction

We extracted the following audio-visual features:

SIFT: SIFT feature [5] has been widely used in many visual recognition tasks. It describes the gradient information around the keypoint. Here the standard DoG keypoints detector is used. A 128-d descriptor is extracted from each keypoint. SIFT feature was extracted on every sampled frame. Both Bag-of-Words and Fisher Vector were used to encode descriptors (cf. Section 2.3).

DIFT: We adopted Uijlings' dense SIFT implementation [6], dubbed DIFT, which is much faster than the standard SIFT. It densely extracts the SIFT feature on the whole frame. Like SIFT, both encoding methods were used.

Global Features: In addition to the SIFT local features, we also extracted a few global features, including Color Moments, GIST, LBP, and TINY. The Color Moments feature was computed by aggregating the first 3 moments of the 3 channels in Lab color space over 5×5 frame grid partitions. The GSIT, LBP, and TINY were implemented following the settings of [7]. The four features were concatenated as a single feature vector for classification.

Part-Level Attributes: We computed a part-level attributes feature, previously adopted in our work on MediaEval2013 Violent Scenes Detection Task [8]. Part filters learned from the ImageNet classes using the deformable part-based models were used to generate filter response maps over the visual frames. An attribute descriptor was formed by concatenating the max response values of the filters applied to three scales of each frame. For more details, please see [8].

MFCC: Acoustic features are complementary to the visual features in many video content recognition tasks. Here we adopted the well-known MFCC. Each soundtrack segment was first obtained by picking a 10-second window around a sampled visual frame. For each segment, we extracted MFCC features on its audio frames. Only Bag-of-Words was used to encode this feature, and the Fisher vector was not used because of its poor result found from our earlier experiments.

2.3. Feature Encoding

For the SIFT, DIFT, and MFCC features, we have a set of descriptors per frame/segment. This creates difficulties for classifiers which normally require fixed dimensional inputs. To convert the sets of features into fixed dimensional representations, we adopted two well-known methods: Bag-of-Words and Fisher Vector.

Bag-of-Words (BoW) [9]: A soft-weighting strategy [10] was used to alleviate the quantization loss of BoW. For MFCC, the vocabulary size is 4000. For SIFT and DIFT, the vocabulary size is 500. Spatial pyramid was adopted for the visual SIFT and DIFT features, using grids 1×1, 3×1 and 2×2. Finally, for each frame and each feature, a BoW representation of 4000 dimensions was formed by concatenating quantized features from the spatial grids.

Fisher Vector (FV): Recently, FV [11] has been proved to be better than BoW in many visual recognition problems, by encoding the first and second order information instead of simple count statistics. We first applied the PCA on the SIFT/DIFT features, reducing the dimension by a factor of two as in [11]. Then, a Gaussian Mixture Model (GMM) was adopted to fit the data with the number of Gaussians K=256. Each frame was then represented with a $2 \times D \times K$ FV, where D = 64 is the feature dimension after PCA reduction. Finally, L2 normalization was applied to the Fisher vectors before classification.

2.4. Classification

As aforementioned, we used linear SVM and DNN for classification. Both are efficient in the test phase.

SVM: We trained classification models with the simple linear SVM. In consideration of memory usage, a subset of the training samples were used. One-vs-all SVMs were trained for each of the target classes. When using multiple features as inputs for SVM training, we adopted linear fusion, which is widely used. Fusion weights were estimated on the validation set. The outputs of the linear SVMs were normalized to the range in $[0, 1]$ for the ease of comparison across different classes.

DNN: Deep learning architectures have exhibited strong performance in many real world applications ranging from speech recognition [12] to large scale visual recognition [13]. We recently developed a 4-layered DNN specially tailored for video classification [1]. The architecture of our DNN is shown in Fig. 2. Using this specifically structured network, we are able to perform feature fusion and classification simultaneously. Specifically, we first use one layer for each feature to perform feature abstraction separately, and then one layer for feature fusion from all the representations with a structural regularization, which enables knowledge sharing among different features as well as reserves the unique characteristics of features at the same time. Finally, the fused feature is used as inputs of the last layer for annotation. The neural network is trained in a back-propagation manner with the gradient descent method. The outputs of last layer are prediction values ranging in $[0, 1]$.

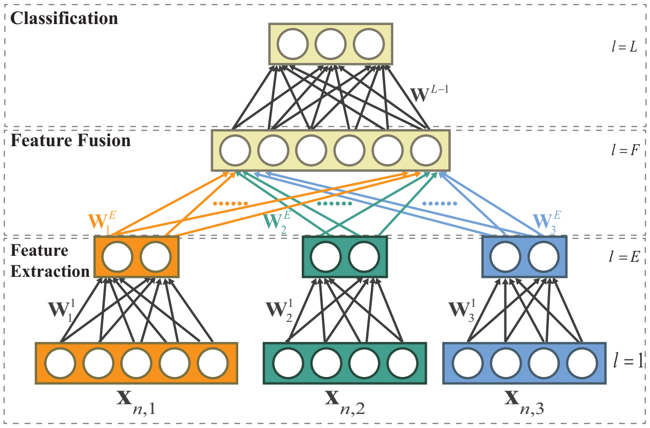

Fig. 2. Illustration of the structure of our DNN. Multiple features are used as inputs of the network, which processes the features separately first, and then enforces a regularization-based framework for feature fusion and classification.

2.5. Post Processing

With the prediction outputs of the SVM and the DNN models, a few post-processing schemes were further imposed, including fusing different kinds of classifiers and temporal smoothing of the prediction scores.

Classifier Fusion: In our submitted Run-3, we further fused the prediction values of SVM and DNN. Linear fusion was employed, with fusion weights estimated on the validation set.

Temporal Score Smoothing: As the video semantics are generally consistent in temporally nearby segments, we used average values of prediction scores over a temporal window, which is helpful for removing prediction noises. Each video contains a set of the sampled frames $\{x_1, x_2, \cdots, x_m\}$. The smoothed score $\widetilde{s(x_l)}$ of a frame x_l using a temporal window of size $2T + 1$ frames can be computed as:

$$\widetilde{s(x_l)} = \frac{\sum_{k=l-T}^{l+T} \widehat{s(x_k)}}{2T+1},$$

where

$$\widehat{s(x_k)} = \begin{cases} s(x_k) & \text{if } 1 \leq k \leq m, \\ \dfrac{\sum_{k'=1}^{m} s(x_{k'})}{m} & \text{otherwise.} \end{cases}$$

After smoothing, a frame is considered containing a class if the prediction score is larger than a threshold, estimated also based on the validation set.

[1] Unpublished work.

3. EXPERIMENTS

Based on the system described earlier, we performed experiments with different combinations of features and classification methods. In the following we first introduce the experimental settings and then discuss the results.

The MoVAC dataset contains 2,666 training videos and 1,455 test videos. The task is distinct from many video annotation benchmarks in several ways. First, the 10 target classes cover very diversified topics, including objects ("car", "dog", "flower", "food" and "kids"), scenes ("beach", "city view" and "Chinese antique building") and events ("football" and "party"). Second, there are many video frames containing multiple classes. Third, the task requires annotations on temporal frame level instead of entire video level. Last, speed is also considered as an important factor, which makes outstanding features like the dense trajectories [1, 14] inappropriate for this task.

We adopt *mean score* as the evaluation measure, following the official definition of MoVAC. A score of a video is defined as the *intersection over union* based on the predicted segments and the ground-truth labels:

$$\text{score} = \frac{\sum_{i=1}^{n} \text{GT}_i \cap \text{Prediction}_i}{\sum_{i=1}^{n} \text{GT}_i \cup \text{Prediction}_i},$$

where $n = 10$ is the total number of classes. The mean score is computed as the average score of all the tested videos.

Following the official evaluation guideline, we randomly divided the training set into a sub-training set (2/3 of the videos) and a validation set (1/3 of the videos). As the labels of the test set are not released, we report performances of most evaluations on the validation set. Results on the official test set will only be reported for the three submitted runs, evaluated by the organizers of the Challenge.

We also report the time needed for computed each feature and the total testing time of each submitted run. The time was measured on the following hardwares. The feature extraction and SVM classification parts were conducted on a workstation with an Intel i7 3.4GHz CPU and 32GB RAM. The DNN training/testing was conducted on a single NVIDIA Telsa K20 5GB GPU, using codes implemented with the MATLAB Parallel Computing Toolbox.

3.1. Comparing the Speed of the Features

Before reporting the annotation accuracies, we first discuss the speed of the feature extraction part. The feature extraction and encoding is the slowest step in a recognition system, and therefore it is very important to select the features that are efficient and reliable.

We report the average time needed to compute each feature on a test video in Table 1. Notice that frame extraction from the videos only costs a fraction of a second, which is not included in the feature extraction time. MFCC is the most efficient feature as the audio soundtrack is just a 1-dimensional signal, in contrast to the 3-d visual pixels. Among the visual features, DIFT is the most efficient one, using only around 1/7 of the time needed for computing the sparse SIFT features. This result suggests that in the fast system we should adopt the dense SIFT features instead of the sparse version.

Table 1. Average time needed to extract features from a test video. The features are sorted by efficiency. MFCC and DIFT are the most efficient features, even faster than the simple global features.

Feature	Extraction Time (s)
MFCC_BoW	3.3
DIFT_FV	12.5
DIFT_BoW	15.4
Global	18.3
Attribute	68.0
SIFT_FV	83.4
SIFT_BoW	87.3

In addition, we also tried to resize the visual frames to a smaller scale for fast feature extraction, which however was observed to hurt the performance greatly and was therefore discarded.

3.2. Comparing the Accuracy of the Features

Next we compare the annotation accuracy of the individual features and their combinations, using the validation set. We only report results using the linear SVM as the classifier in this experiment. The major conclusions of which features are effective will not change while using the DNN. Results are visualized in Fig. 3.

Comparing the features separately, MFCC is the worst. The visual features are all very effective with a score between 50% and 60%. This is not surprising because most of the target classes can be better captured by the visual clues, e.g., "flower", "beach", and "city view".

Further combining MFCC with the visual features can improve the results, because some classes may contain valuable auditory clues like "party". The mean score is 2.30% higher by fusing DIFT_FV with MFCC (indicated by "13" in Fig. 3) than using DIFT_FV alone (indicated by "3"). By adding the attribute feature (indicated by "134"), the mean score is significantly improved to 65.82%. Combining all the features together can further improve the results (indicated by "all"), slightly. Notice that in this fusion experiment, we do not evaluate all the feature combinations due to space limitation. Instead we incrementally add in the features, and a newly added feature is discarded in later fusion experiments if it does not improve the results. In the figure we only report the feature combinations that are observed to be effective.

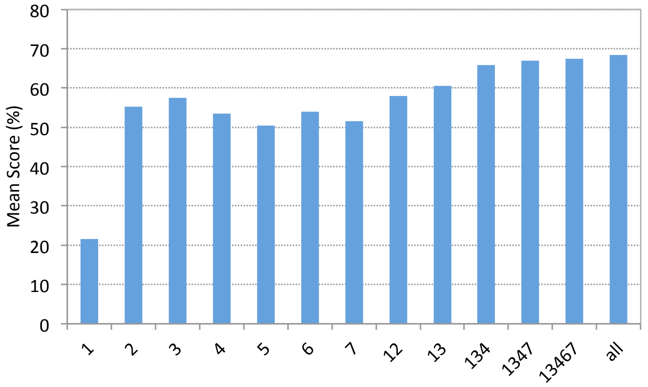

Fig. 3. Accuracies (mean scores) on the validation set using linear SVM trained with individual features and their fusion. 1. MFCC; 2. DIFT_BoW; 3. DIFT_FV; 4. Attribute; 5. SIFT_BoW; 6. SIFT_FV; 7. Global. See Section 3.2 for more discussions.

Table 2. Accuracies of our submitted approaches on the validation set. Combining the two kinds of classifiers leads to the highest result.

Approach	Mean Score
DNN Fast	63.22%
SVM Strong	68.35%
DNN Strong+SVM Strong	71.17%

Table 3. Official accuracies and speed of our submitted approaches on the test set of MoVAC. Speed is measured by the total time of processing the entire test set, including all the processing modules from frame sampling to classification.

ID	Approach	Mean Score	Time (s)
1	DNN Fast	62.53%	27,411
2	SVM Strong	63.14%	912,417
3	DNN Strong+SVM Strong	68.94%	912,417

3.3. Comparing and Fusing DNN and SVM

In this subsection, we compare and fuse the two kinds of classification methods, also on the validation set. We first use the two most efficient features MFCC and DIFT. As shown in Fig. 3, the FV encoding offers better performance than the BoW. With linear SVM as the classifier, we obtained a mean score of 58.03% (“12” in Fig. 3) with DIFT_BoW and MFCC, and 60.82% (“13” in Fig. 3) with DIFT_FV and MFCC. However, the FV representation cannot be used as inputs of the DNN because the dimension is too high. Therefore we only adopted the BoW encoding for the DNN and compare it with the FV encoding using SVM. Notice that the DNN classification with the DIFT_BoW and the MFCC features is exactly the setting used in our submitted Run-1. We achieved a mean score of 63.22% on the validation set, which is better than the linear SVM approach using DIFT_FV and MFCC (60.82%). This clearly shows that DNN is powerful while using even slightly weaker features. We attribute this to a fact that the DNN structure we used was designed with a special function of modeling feature relationships, which cannot be discussed in detail in this paper due to space limitation.

We also compare the two methods using more features. This time the DNN is trained based on MFCC, DIFT_BoW, SIFT_BoW and the Attribute features, namely “DNN Strong”, and the SVM is trained using all the features (indicated by “SVM Strong” in Fig. 1, which is our Run-2). This “DNN Strong” approach and the “SVM Strong” (Run-2) approach achieve mean scores of 69.61% and 68.35%, respectively, which again verify the effectiveness of the DNN although less features were used.

We further fuse the two kinds of classifiers to study whether the two methods are complementary. Linear fusion is used, with weights estimated on the validation set. We fuse the two methods using their “strong” settings (i.e., our submitted Run-3). This simple fusion method further improves the accuracy to 71.17% on the validation set, indicating that SVM and DNN are complementary to some extent. Table 2 summarizes the accuracies (on the validation set) of the three approaches used in our official submissions.

Fig. 4 further gives several visual examples of the annotation results. We can observe that many false alarms share some common patterns with the positive samples, e.g., the water scenes with the “beach” class and the grass fields with the “football” class. Stronger features are needed to distinguish these samples, which deserve in-depth future investigations.

3.4. Official Submissions

Finally, we report the results of our official submissions on the test set of MoVAC in Table 3. Overall, the results are consistent with that obtained on the validation set. “DNN Fast” is a practically preferable approach as it is highly efficient and is just a few percents lower. Note that the estimated processing time of Run-3 is the same as that of Run-2. This is because the time needed by DNN prediction is almost negligible compared with that of feature extraction, and Run-2 and Run-3 have exactly the same set of features.

4. CONCLUSION

In this paper, we have discussed a comprehensive system for video annotation. Our system consists of a large set of audio-visual features and two kinds of classification methods, SVM and DNN. In particular, the DNN method has a special structure tailored for video annotation with multiple features, which has been observed to be consistently better than the

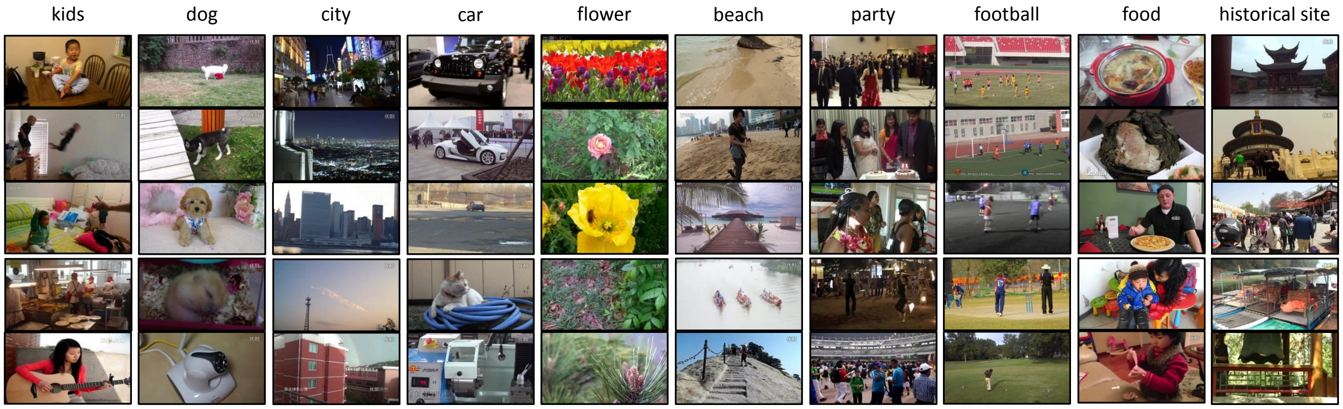

Fig. 4. Example annotation results on the validation set by the "DNN Strong+SVM Strong" approach. For each class, the first three rows are correctly annotated frames and the last two rows are false alarms.

SVM. We achieved the best accuracy across all the submitted results to this challenge. Our approach is also efficient. The DNN based Run-1 only requires 18.8 seconds to process a test video, including both feature extraction and classification. One promising direction to further improve the performance is to develop deep learning techniques that can learn the features directly from the raw video data, in addition to using the hand-crafted features in the current system.

5. ACKNOWLEDGMENTS

This work was supported in part by two grants from the National Natural Science Foundation of China (#61201387 and #61228205), a National 863 Program (#2014AA015101), two grants from the Science and Technology Commission of Shanghai Municipality (#13PJ1400400 and #13511504503), and a project from the Ministry of Education (#20120071120026), China.

6. REFERENCES

[1] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013.

[2] Guangnan Ye, Dong Liu, I-Hong Jhuo, and Shih-Fu Chang, "Robust late fusion with rank minimization," in *CVPR*, 2012.

[3] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Deep Fisher networks for large-scale image classification," in *NIPS*, 2013.

[4] Yu-Gang Jiang, "Super: Towards real-time event recognition in internet videos," in *ICMR*, 2012.

[5] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[6] Jasper RR Uijlings, Arnold WM Smeulders, and Remko JH Scha, "Real-time visual concept classification," *TMM*, vol. 12, no. 7, pp. 665–681, 2010.

[7] Jianxiong Xiao, James Hays, Krista Ehinger, Aude Oliva, and Antonio Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010.

[8] Qi Dai, Jian Tu, Ziqiang Shi, Yu-Gang Jiang, and Xiangyang Xue, "Fudan at MediaEval 2013: Violent scenes detection using motion features and part-level attributes," in *MediaEval Workshop*, 2013.

[9] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.

[10] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *CIVR*, 2007.

[11] Florent Perronnin, Jorge Snchez, and Thomas Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010.

[12] Geoffrey Hinton, Li Deng, Dong Yu, and et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *SPM*, 2012.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet classification with deep convolutional neural networks.," in *NIPS*, 2012.

[14] Robin Aly, Relja Arandjelovic, Ken Chatfield, and et al., "The AXES submissions at TrecVid 2013," in *NIST TRECVID Workshop*, 2013.