Scene Aligned Pooling for Complex Video Recognition

Liangliang Cao[†], Yadong Mu[♯], Apostol Natsev[§],
Shih-Fu Chang[♯], Gang Hua[‡], and John R. Smith[† ⋆]

[†] IBM T. J. Watson Research Center
{liangliang.cao,jsmith}@us.ibm.com
[♯] Dept. Electrical Engineering, Columbia University
{muyadong,sfchang}@ee.columbia.edu
[§] Google Research
natsev@google.com
[‡] Dept. Computer Science, Stevens Institute of Technology
ghua@stevens.edu

Abstract. Real-world videos often contain dynamic backgrounds and evolving people activities, especially for those web videos generated by users in unconstrained scenarios. This paper proposes a new visual representation, namely scene aligned pooling, for the task of event recognition in complex videos. Based on the observation that a video clip is often composed with shots of different scenes, the key idea of scene aligned pooling is to decompose any video features into concurrent scene components, and to construct classification models adaptive to different scenes. The experiments on two large scale real-world datasets including the TRECVID Multimedia Event Detection 2011 and the Human Motion Recognition Databases (HMDB) show that our new visual representation can consistently improve various kinds of visual features such as different low-level color and texture features, or middle-level histogram of local descriptors such as SIFT, or space-time interest points, and high level semantic model features, by a significant margin. For example, we improve the-state-of-the-art accuracy on HMDB dataset by 20% in terms of accuracy.

1 Introduction

The problem of video event recognition is attracting more and more attention in recent years. This is largely due to two reasons: On one hand, the popularity of video cameras makes it possible for a consumer to record or compose video clips easily. On the other hand, the emergence of social media websites including Youtube, Facebook has aggregated a large amount of online video corpus, which plays a major role in attracting Web users. For example, Youtube hosts more than 100 million videos and serves 1 billion video requests per day. Such a large amount of video data has never been available until today. How to understand the contents of these videos has become an important challenge for computer vision research.

The problem of video event recognition is the key to many applications, including personalized video recommendation, social event mining, and large scale video library

⋆ Gang Hua participated in this project while working for IBM as a visiting researcher. Apostol Natsev was with IBM Watson Research Center when most of this work was performed.

Fig. 1. Complex videos are usually composed of different scenes. (a) Key frames in a video of wedding ceremony (b) Key frames in a video of parkour activities.

indexing. The definition of "event" generalizes previous studies of simple human actions in a constrained environment [1] [2], or highly distinguishable professional activities in Olympic game or TV channels [3] [4]. In this paper, an event can be either a complicated human activity (e.g., kiss, parade, making a sandwich), or composite multimedia semantics (e.g., birthday party, wedding ceremony). Compared with simple human actions, the events in these complex videos are more attractive, since Web users enjoy those videos with rich semantics but feel bored about simple ones. A real world video often contains heterogeneous backgrounds and different viewpoints, and captures diversified contents or evolving human activities. Fig. 1 illustrates some exemplar videos of complex event (e.g., wedding ceremony and parkour), which are obviously more complicated than simple actions like running or walking.

To study the problem of event recognition in complex videos, we observe that a video clip is composed with shots of different scenes. In this paper we refer "scene" to fine-grained characteristics of video environmental semantics. Both psychological and biological evidences [5] [6] revealed that human vision can easily distinguish different scenes and the results of scene recognition can further help general image understanding. Motivated by these psychological theories, we believe scene information is a good cue to understand complex video events.

The key idea of our method is to use scene information to guide the pooling operation of video features. Traditional pooling is an operation of averaging the feature vectors within a spatial neighborhood of images [7]. In this paper, we aggregate video features into concurrent scene components, and then develop scene-dependent modules for the classification task. Since our new model is designed to capture the visual information in concurrent scenes, we name this new model as *Scene Aligned Pooling* (SAP). The advantages of our scene aligned pooling are four-folds: (1) This new visual representation naturally captures diverse video contents and dynamic semantics based

on scene structure. (2) SAP can be applied to various visual features and improve their performance. (3) SAP represents videos as feature vectors of fixed dimensions, and the model is free from the concerns of video length or assumptions on temporal evolvement. (4) We employ a soft weighting strategy named concurrent vector quantization for pooling different scene components, which is not only robust to noise but also able to handle the scenario with overlapped scenes.

2 Related Works

A real world video always conveys rich information, and it is a challenging task to capture such rich information for video event analysis. Existing studies [1] [2] [8] [3] consider human actions in specified scenarios or well-controlled environment. However, there have been not enough studies on recognizing unconstrained video or user generated video from online websites. Compared with traditional human action datasets, these real world videos are more difficult to handle since they contain longer video sequences and diversified scenes. Moreover, when those real world videos contain non-static cluttered background, the popularly used motion-dependent features [9] [10] will not work well since there are a lot of false detections due to the background motion.

Since video event classification is still in its early stage, it is desirable to learn from successful image classification techniques. As argued in [11], a lot of recent progresses in image recognition can be viewed as a combination of alternating series of coding and spatial pooling steps. Average pooling [7] tries to average feature vectors within a spatial neighborhood. Bag-of-words model can be viewed as a special case of average pooling. Max pooling is found to be useful for sparse coding features [12]. Jégou *et al.*developed a new image descriptor called VLAD by pooling local descriptors [13]. Lazebnik *et al.*proposed spatial pyramid matching (SPM) kernel [14], which can be viewed as a pooling method based on the spatial layout of images. This idea is recently generalized in [15], which enlarges the spatial bins with feature space clustering. The difference between this paper and previous image pooling methods is three-fold: (1) [15] uses either hard VQ or sparse coding, which usually requires a large size of codebook. This submission chooses only codebook size =16, using the weights of harmonic average. (2) [15] only pools the coding weights or quantized indices, while we pool the feature vectors. (3) [15] uses either max pooling or average pooling, while we compute the weighted average.

Our work is also motivated by the studies in scene recognition. A number of studies [16] [17] [14] have been carried out to classify natural scenes. Later studies explore various applications of scene recognition. Russell *et al.*[18] employed scene representation for object retrieval. Many research studies show empirical success in learning image representation within similar scenes [19] [18]. Li and Fei-Fei [20] and later Marszalek *et al.*[21] combine scene classifiers to benefit the task of object recognition or activity recognition. However, these approaches treat scene classifiers as an independent component, and require a lot of scene labels to train the scene classifiers. Unlike their method, this paper employs scenes as the pooling context, and only use scene features without the requirement of scene labels.

3 Scene Aligned Pooling

3.1 Model

A video is comprised of a sequence of frames. Following the extensive research in image recognition, we can represent frames with various feature vectors, including color histogram, LBP, texture, or SIFT histogram. Consider a video $X = [x_1, \cdots, x_{Tx}]$, where $1 \leq t \leq Tx$ stands for the index of frames or key frames, x_t corresponds to a feature vector. Note that the number of frames is not consistent across different videos.

According to the representer theorem [22], the classifier will be represented by kernel distances $K(X, Y)$ between two videos X and Y. Since X and Y may contain different number of frames, traditional methods usually compute the kernel by averaging the frame features

$$K(X, Y) = \kappa(\frac{1}{\rho_x}\sum_{t=1}^{Tx} x_t, \frac{1}{\rho_y}\sum_{t=1}^{Ty} y_t), \tag{1}$$

where κ is a kernel function between two vectors, x_t and y_t are the frame level feature vectors, ρ_x and ρ_y are the normalization factors. For example, we can choose $\rho_x = ||\sum_{t=1}^{Tx} x_t||$.

Eq (1) is widely used with different features. In recently years, many systems use bag of words model for video recognition, which fundamentally is to average the histogram features across different frames. The limitation of eq (1) is that it overlooks the diversity of video contents. The learned classification model treats all the frames using the same mechanism. If a video event contains non-uniformed backgrounds, averaging the frame features will inevitably blur the discriminant features and hence reduce the recognition performance.

In this paper, we consider the scenario where a video contains K scenes. Let s_t be the random variable for scene type of frame t, where $1 \leq s_t \leq K$. Note $p^k(x_t) = P(s_t = k|x_t)$ satisfies the constraint $\sum_k p^k(x_t) = 1$. With $p^k(x_t)$ we can redefine the kernel distance between two videos

$$K(X, Y) = \sum_{k=1}^{K} \kappa(\frac{1}{\rho_x^k}\sum_{t=1}^{Tx} p^k(x_t)x_t, \frac{1}{\rho_y^k}\sum_{t=1}^{Ty} p^k(y_t)y_t). \tag{2}$$

To make the representation clearer, we introduce a new variable named *scene component* which combines the outputs of all video frames by

$$s^k = \sum_{t=1}^{Tx} p^k(x_t)x_t, \quad r^k = \sum_{t=1}^{Ty} p^k(y_t)y_t. \tag{3}$$

The kernel becomes

$$K(X, Y) = \sum_{k=1}^{K} \kappa(\frac{1}{||s^k||}s^k, \frac{1}{||r^k||}r^k). \tag{4}$$

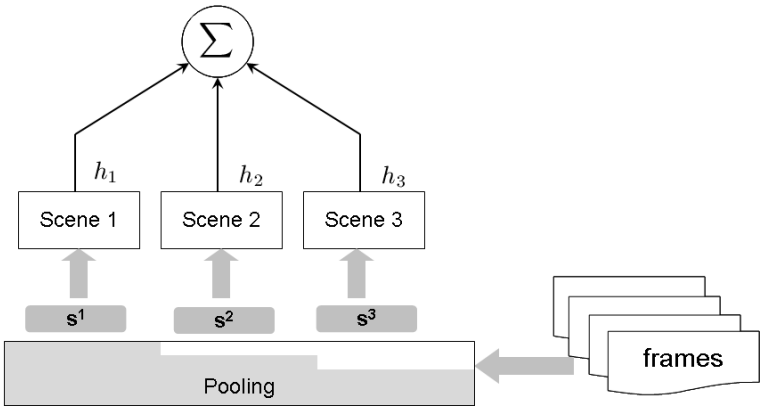

Fig. 2. An practical viewpoint for SAP model.

This new representation aligned videos of different length by K types of scenes, and we call this method *Scene Aligned Pooling* (SAP). It is easy to see that the average pooling is a special case of SAP with $K = 1$. The motivation of SAP is to employ multiple scenes to represent the diversified video contents.

It is interesting to compare our formulation with Lazebnik *et al.*'s Spatial Pyramid Matching (SPM) [14]. In summary, SAP composes kernels over the domain of semantic scenes, while Lazebnik's SPM builds kernels over spatial grids. However, there are significant differences between SPM and SAP. SAP works in a semantic scene domain with no clear boundaries. SPM works in spatial domain where we can use grids to explicitly separate one image into several parts. As a result, we do not explore the pyramid structures in SAP due to the ambiguous boundaries in video domain. Instead, we choose concurrent vectorization to perform soft weighted assignment, while SPM is based on hard assignments. Table 1 summarizes the differences between SAP and SPM.

Table 1. Comparing SAP with Spatial Pyramid Matching.

	Subject	Domain	Grid	Assignment
SAP	video	scene	$\times$	soft
SPM[14]	image	spatial	✓	hard

To provide more insights for the SAP model, we consider the linear classifier corresponding to (4). The linear classifier can be represented as

$$H(X) = \sum_{t=1}^{Tx} \sum_{k=1}^{K} p^k(x_t) w_k^T x_t = \sum_{k=1}^{K} w_k^T s^k \tag{5}$$

It is easy to see that SAP leads to different models for different scene components and combine the estimations from multiple scenes into the final model. Fig. 2 illustrates the idea for SAP classifier.

The biggest advantage of scene component representation is that it makes the training easier. We can first concatenate all the K scene components as a long vector,

$$S = [s^1, s^2, \cdots, s^K]$$

and learn a linear function $H(S) = w^T S + b$. In practice, we enforce $l_2$normalization before training by $S = [\frac{1}{||s^1||_2}s^1, \frac{1}{||s^2||_2}s^2, \cdots, \frac{1}{||s^K||_2}s^K]$. In this way, we obtain the long linear coefficients as a whole. Due to the recent advances in linear model training [23], it has become pretty efficient to learn linear SVMs from high dimensional large scale data, so that training using scene component will not be difficult.

3.2 Implementation

To implement the computation of scene component $s^k = \sum_{t=1}^{T} p^k(x_t)x_t$, in fact what we need to do is to map each frame feature x_t into different scenes

$$x_t \rightarrow [p_t^1 x_t, p_t^2 x_t, \cdots, p_t^K x_t], \tag{6}$$

with the constraint $\sum_k p_t^k = 1$. From eqs (3) and (6) we can easily see that our new model is a pooling method. The unique characteristic of our method is that our pooling weights p_t^k are based on scene semantics. That is why we call our method "scene aligned pooling". We will explain how to compute p_t^k in the following.

Our method of modeling p_t^k is very easy to implement. Following the previous studies [16] [18] [24], we use GIST features to represent the scene. Some may argue that we might use the same feature as frame feature vector x to compute the scene, however, as we shown later, the pooling weights are computed based on Euclidean distance, which is not reliable for sparse histogram features like SIFT. As our scene modeling feature, GIST is easy to compute and especially good at describing scenes [16]. Given all the training data, we compute the GIST features and cluster them into K centers using unsupervised K-means. The K center is represented as $g_c^1, g_c^2, \cdots, g_c^K$. To compute the pooling weight $p^k(x_t)$, we extract the GIST feature vector g_t for every frame t. The pooling weight is based on the comparison of $\{g_c^k\}$ and g_t.

The naive way to compute the pooling weight is by vector quantization (VQ), which forces all but one $p^k(x_t)$ to be zero. However, VQ is well-known to be sensitive to noises. Moreover, the hard assignment of VQ cannot capture the overlapping nature of evolving video scenes and hence works poorly. VQ is designed to minimize the quantization error. Suppose G is the set of GIST features, this paper considers a different criteria by

$$C = \sum_{g \in G} HA(g) = \sum_{g \in G} \left(\frac{1}{K} \sum_{k=1}^{K} \frac{1}{||g - g_c^k||} \right)^{-1},$$

where HA denotes the harmonic average proposed by Zhang *et al.*[25]. To study the minimization condition, we let $\frac{\partial}{\partial \mathbf{m}_k} C_2 = \mathbf{0}$, which leads to

$$g_c^k = \sum_{g \in G} w_g^k g, \quad w.t. \qquad w_g^k \propto \frac{1}{||g - g_c^k||^3}.$$

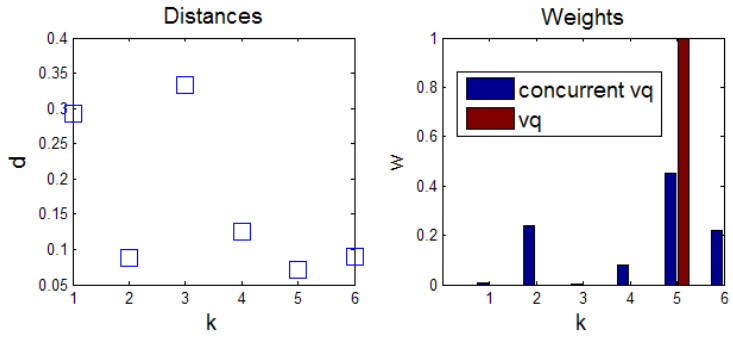

Fig. 3. Differences between concurrent VQ and VQ. Left: an exemplar of $d_t^1, \cdot, d_t^K$ ($K = 6$ in this toy example). Right: the weights from vector quantization and concurrent vector quantization.

To be consistent to the contribution of g_t for harmonic average, we choose our pooling weights as

$$d_k^t = ||g_t - g_c^k|| \\ p_t^k = \frac{1/(d_t^k)^3}{\sum_{l=1}^K 1/(d_t^l)^3}, \tag{7}$$

It is not difficult to see that our pooling weights satisfy the constraint $\sum_k p_t^k = 1$. Also our pooling weights relies on only GIST feature, with no relations with the event label y. This means we will use the same pooling strategy independent with events. We call the method in eq. (7) as concurrent VQ. Compared with traditional VQ, our new pooling method penalizes the impact of large outliers, and assigns more weights to the center with small distance. Fig. 3 employs a toy example to show the difference between eq. (7) and traditional VQ. In this toy example, the sample is close to three centers ($k = 2, 5, 6$), which can be viewed as a picture with three overlapping scenes. Vector quantization will only pick up $k = 5$, while forcing all the weights to the other centers as zero. In contrast, our new method selects all these three centers with big weights, and hence can handle the scenario of overlapping scenes easily. Moreover, the computation of eq (7) shows that the pooling weights in our method are based on the Euclidian distance d_t^k. For a lot of sparse histogram feature (such as SIFT histogram), their Euclidean distance is not reliable so that they are not as good choices as GIST.

In this paper, we choose an unsupervised way to model scenes. The reasons why unsupervised learning is preferred are as follows: First, unsupervised learning can save the extra labeling efforts. In our model, the goal is not to recognize the exact scene category but to do pooling according to scene context, so supervised learning is not necessary. Moreover, we will explain later that some video frames are associated with overlapping scenes.

In this paper, we use two kinds of κ as kernel function: linear kernel and intersection kernel. When we use linear kernel, we employ liblinear [26] with its default parameter. When we use intersection kernel, we use nonlinear SVM because there is no extra

parameter to compute intersection kernel, and the intersection kernel can be computed very efficiently.

One advantage of our scene aligned pooling method is that it can be combined with various features. For example, many low level features have been designed in the image recognition research to represent a single image. To combine these features with scene aligned pooling, we can first identify the scene for each frame, and then aggregate the frame-level representations according to these scenes. For space-time interest point features [10], we can still use scene aligned pooling by aggregating the histogram for short term clips. We can also generalize this approach to multi-modal features, if both visual and audio features are synchronized.

Next we briefly discuss how to select K, the number of scenes. There is no general agreement on how many scenes exist, considering the ambiguity and great complexity contained in real world. Fei-Fei and Perona [17] employed 13 scenes in their experiment, which is later enlarged to 15 scenes by Lazebnik *et al.*[14]. The largest scene collection is recently contributed by Xiao *et al.*[24], which reported a recognition average precision of 34.5% for total 397 scene categories. [24] also conducts an interesting user study, by inviting 7 participants to write down all scene categories they experienced in more than two hundred hours. In this experiments, the participants report 52 different scene categories. In our experiments, we test the performances with different K. Fig. 4 compares results of pooling color histogram features on TRECVID 2011 dataset. We will explain the dataset later in the experiment section. We can see that the performance is similar with $K = 16, 64, 128$. In following section, we will use $K = 16$ for all the experiments, for efficiency.

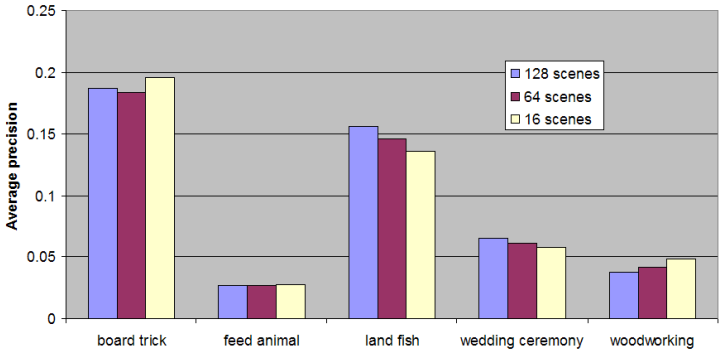

Fig. 4. Comparing the effects of different scene numbers on TRECVID 2011 Dryrun dataset.

Algorithm 1 summarizes the workflow of our SAP algorithm.

4 Experiments

In recent years, there has been a proliferation of Web-shared videos, with approximately 48 hours of video uploaded on Youtube every minute, and over 700 billion

Algorithm 1 Scene Aligned Pooling Algorithm.

Input: : A video with the set of frame feature vectors $\{x_t\}$, $1 \leq t \leq T$. A codebook with K centers of GIST feature g_c^k, $1 \leq k \leq K$.
Extract GIST descriptor g_t for every frame t.
Compute the pooling weight $p^k(x_t)$ using eq. (7).
Compute scene component s^k using eq. (3).
Normalize s^k and concatenate them into a long vector S.
Train linear SVM or intersection-kernel SVM using S.

videos watched in 2010. In our experiments, we select datasets most similar to real world online videos: TRECVID Multimedia Event Detection (MED) 2011 corpus (http://www.nist.gov/itl/iad/mig/med11.cfm) and the Brown Human Motion Recognition Database(http://serre-lab.clps.brown.edu/resources/HMDB/) (HMDB).

4.1 TRECVID MED Datasets

Trecvid MED 2011 is the largest fully annotated dataset specifically designed to model complex video events. There are about 370 hours of clips in the Event-Kit and Transparent Development (DEV-T) collections, and another 1,200 hours of video clips in the Opaque Development (DEV-O) collection. All these videos are free of editing, accompanied with non-professional recording and variety of illumination, camera motion, and cluttered background. The video duration is similar to the real Youtube video, and the average duration is about 2 to 3 minutes. The evaluation of MED is separated in two test sets: the mid size dryrun evaluation and the large final evaluation set. In the dryrun evaluation, 5 events are considered: *attempting a board trick*, *feeding an animal*, *landing a fish*, *wedding ceremony*, *working on a woodworking project*, for which labels are provided for both the training and testing sets. The final evaluation stage, on the other hand, considers 10 new events: *birthday party*, *changing a vehicle tire*, *flash mob gathering*, *getting a vehicle unstuck*, *grooming an animal*, *making a sandwich*, *parade*, *parkour*, *repairing an appliance*, and *working on a sewing project*. Each event is a combination of one or multiple people, scenes and actions. In the following we will discuss the performance of SAP for various features in both dryrun and final evaluation stage.

In this paper, we employ average precision (AP) measure to evaluate the performance on MED datasets. The reason why we choose average precision is because it is a popular measure in computer vision field and more importantly it is easier to compare the performance of video retrieval using average precision. Suppose the scores are ranked in descending order, the AP of the ranked list is computed by $AP = \sum_{i=1}^{n} p(i) \triangle r(i)$,, where $p(i)$ is the precision at i-th position in the ranked list, and $r(i)$ is the recall at i-th position. $\triangle r(i)$ is the change in recall from item $i-1$ and i.

Since the dryrun evaluation is of a relatively small scale, we will try the effectiveness of SAP for different features. We tried the following features: edge histogram (edgehist), color histogram (colorhist), SIFT histogram (SIFT). The details of implementing these features are described in [27]. We compare three pooling methods: max pooling, average pooling, and our SAP. From Table 2 we can see that max pooling works poorly in video recognition. The average pooling works much better than max pooling for

video recognition. In the following, we will use average pooling as the default baseline, since it is the standard technique in the field. In our experiments, SAP works consistently best among all the pooling methods for all the features. The improvement over mean AP can be as significantly as $(0.92 - 0.39)/0.39 = 135\%$ for edge histogram, 131% for color histogram, 10% for local binary pattern (LBP), 18% for SIFT features.

Table 2. Comparison of different pooling methods on MED dryrun

Feature	Pooling	Average Precision					
		E1	E2	E3	E4	E5	Mean
edgehist	Max	.027	.024	.051	.024	.016	.029
	Ave	.045	.015	.079	.020	.036	.039
	SAP	.195	.028	.136	.057	.048	.092
colorhist	Max	.023	.024	.036	.078	.028	.038
	Ave	.105	.027	.062	.039	.024	.051
	SAP	.259	.030	.129	.128	.042	.118
LBP	Max	.022	.018	.038	.021	.020	.024
	Ave	.057	.030	.045	.033	.019	.037
	SAP	.067	.028	.046	.041	.020	.041
SIFT	Max	.042	.013	.191	.013	.019	.056
	Ave	.155	.039	.261	.311	.152	.184
	SAP	.166	.044	.252	.432	.190	.217

We also did a few comparison with other pooling techniques. the effects of our concurrent vector quantization and its degenerated case (hard scene assignment). As shown in Table 3, our soft scene assignment strategy is much better than the hard assignment. The only exception is in event 2 (feed an animal), for which neither VQ nor our method does a good job due to the high irregularity and diversity in animal appearance and feeding activities. We also try to use SIFT histogram to compute $p^k(x_t)$, however, the pooling results are very poor since the Euclidian distance of sparse histogram features are not reliable. Another experiment we consider is whether directly combining scene feature and SIFT can improve the result. We extract both GIST and SIFT features for each video frame, and use average pooling to train the classifier. As shown in Table 4, GIST feature is not a good representation for event recognition, so that it bears very low recognition performance. Since GIST feature performs much worse than SIFT, it makes the naive fusion result worse than that of using SIFT only. However, our SAP does not fuse the two features directly but to use scene information as a context to guide the pooling process. As a result, SAP is a much better choice than the naive fusion strategy.

After we finish the dryrun evaluation, we also apply the proposed method on the final evaluation dataset. Since TRECVID has not publicly released the labels for the DEV-O set, we use an internal test set split. The internal test consists of 40 positive video clips per event (400 videos in total) and 5,231 negative videos. The remaining 7,252 videos are used for training. From the dryrun dataset we know that SIFT outperforms many other low-level features so we are especially interested in the performance of average pooling and scene aligned pooling using SIFT feature. As shown in Fig. 5,

Table 3. Comparing average precision of concurrent vector quantization with vector quantization (soft vs. hard scene assignment) using SIFT feature.

	E1	E2	E3	E4	E5	Mean
Our Scene aligned pooling	.166	.044	.252	.432	.190	.217
Pooling using VQ on GIST	.159	.055	.221,	.263	.071	.154
Pooling using VQ on SIFT histogram	.089	.033	0.091	.084	.051	.077

Table 4. Comparison with naive fusion of scene feature with SIFT.

Feature	Average Precision				
	E1	E2	E3	E4	E5
GIST	.074	.017	.030	.033	.018
SIFT	.155	.039	.261	.311	.152
GIST + SIFT	.150	.040	.211	.256	.143
SAP + SIFT	.166	.044	.252	.432	.190

SAP can significantly improve the average precision in all the ten events. By using SAP, the mean of ap scores can be improved from 0.183 to 0.212.

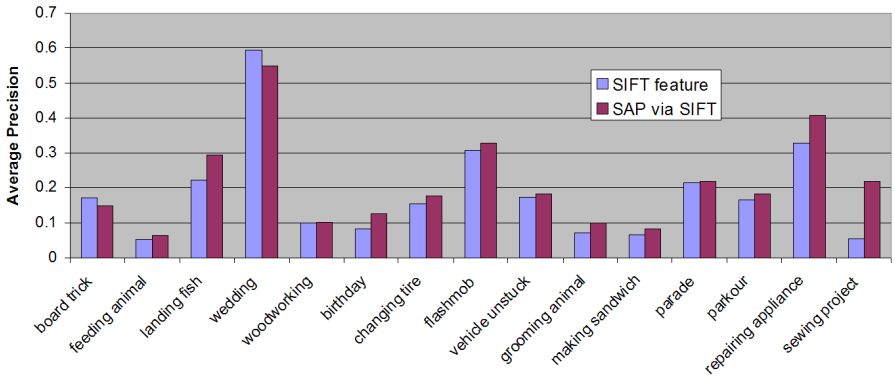

Fig. 5. Results of nonlinear SVM using SAP.

Finally, we consider a special feature called semantic model vectors. Our semantic model vector is an intermediate level semantic representation, by evaluating 780 concept classifiers for each frames. The 780 classifiers are trained separately using thousands of labeled web photos. The semantic model vector is complementary to low level features and can be useful in many retrieval and annotation tasks [28] [29]. Our SAP can also significantly improve the semantic model vector. As shown in Fig. 6, SAP improves the performance in 9 of 10 events.

After obtaining the results using different features, we can do a late fusion to get the final classification model. This paper does not focus on fusion techniques but our

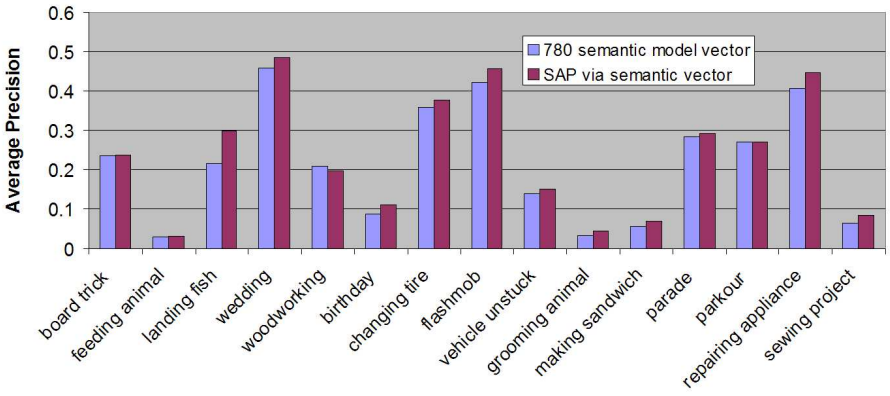

Fig. 6. Results of using SAP for semantic features

simple fusion model can arrive at 0.50 average precision in the dryrun evaluation and 0.45 in the final evaluation (internal test data split).

4.2 HMDB dataset

Very recently, Kuehne *et al.*[30] describe an effort of designing a large video database containing 51 distinct action categories, named the Human Motion DataBase (HMDB), which tries to better capture the richness and complexity of human actions. They argued that the UCF Sports dataset [3] is designed for specific titles on YouTube, in which the actions are usually unambiguous and highly distinguishable from shape cues alone (e.g., the raw positions of the joints or the silhouette extracted from single frames). They collected a new motion dataset, which contains 51 distinct action categories, with at least 101 clips for each category. The final dataset includes a total of 6,766 video clips extracted from a wide range of sources. Each clip was validated by at least two human observers to ensure consistency. Kuehne *et al.*[30] also studied the biological motion perception and recognition technique [31] based on this new dataset.

The HMDB dataset is very challenging. From the reports in [30], the the state-of-the-art's performance is about 23%. It is very interesting to apply our SAP model to this challenging dataset. We use STIP features [10] provided in the dataset webpage, and extract scene features for video frames at every 0.5 seconds. Since the STIP features are only sparsely distributed among video frames, to improve performance we condense the STIP features from nearby half-second video frames when computing on specific target keyframe. The STIP histogram are aggregated together for every 0.5-second clips, and then pooled using SAP model. We compare the performance of SAP with those of STIP histogram, and Kuehne's biological motion system C2. Table 5 compares the recognition accuracy of our SAP with STIP histogram and C2 models. Our model significantly improve the best performance from 23.18% to 27.84%, as a relative 20% increase of accuracy.

Table 5. Comparison with the results on HMDB

Model	Accuracy
STIP histogram	21.96%
C2	23.18%
SAP + STIP	27.84%

5 Conclusion and Future Work

In this paper, we have discussed a new pooling method named scene aligned pooling (SAP). We show that SAP can consistently improve different features (color histogram, SIFT, semantic model vectors) for complex video classification. SAP also significantly improves the state-of-the-art performance on HMDB datasets.

Our future work will focus on generalizing the classification model to more video recognition and annotation tasks. More results will be available on `http://researcher.ibm.com/person/us-liangliang.cao`.

Acknowledgement

This research is supported by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Busyness Center contract number D11PC20070. The U.S. government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

1. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. ICPR (2004)
2. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. ICCV (2005)
3. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos ”in the wild”. CVPR (2009)
4. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. ICCV (2003) 726–733
5. Epstein, R., Kanwisher, N.: A cortical representation of the local visual environment. Nature **392** (1998) 598–601
6. Friedman, A.: Framing pictures: The role of knowledge in automatized encoding and memory for gist. Journal of Experimental Psychology **108** (1979) 316–355
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86** (1998) 2278–2324
8. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. IJCV **79** (2008) 299–318

9. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. IEEE International Workshop on VS-PETS (2005)
10. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. CVPR (2008)
11. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR. (2010)
12. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear pyramid matching using sparse coding for image classification. In: CVPR. (2009)
13. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 3304–3311
14. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. Volume 2. (2006) 2169–2178
15. Boureau, Y., Le Roux, N., Bach, F., Ponce, J., LeCun, Y.: Ask the locals: multi-way local pooling for image recognition. In: ICCV. (2011)
16. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV **42** (2004)
17. Fei-Fei, L., Perona, P.: A bayesian hierarchy model for learning natural scene categories. In: CVPR. (2005)
18. Russell, B., Torralba, A., Liu, C., Fergus, R., Freeman, W.T.: Object recognition by scene alignment. In: NIPS. (2007)
19. Boutell, M., Luo, J., Brown, C.M.: Improved semantic region labeling based on scene context. In: ICME. (2005)
20. Li, L.J., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: ICCV. (2007)
21. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR. (2009)
22. Kimeldorf, G., Wahba, G.: Some results on tchebychefan spline functions. Journal of Mathematical Analysis and Applications **33** (1971) 82–95
23. Yu, H., Hsieh, C., Chang, K., Lin, C.: Large linear classification when data cannot fit in memory. In: Proceedings of ACM SIGKDD, ACM (2010) 833–842
24. Xiao, J., Haysy, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR. (2010)
25. Zhang, B., Hsu, M., Dayal, U.: K-harmonic means-a data clustering algorithm. Hewlett-Packard Labs Technical Report HPL-1999-124 (1999)
26. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: Liblinear: A library for large linear classification. The Journal of Machine Learning Research **9** (2008) 1871–1874
27. Cao, L., Chang, S.F., Codella, N., Cotton, C., Ellis, D., Gong, L., Hill, M., Huang, G., Kender, J., Merler, M., Mu, Y., Natseve, A., Smith, J.R.: Ibm research and columbia university trecvid-2011 multimedia event detection (med) systems. In: NIST TRECVID Workshop. (2011)
28. Natsev, A., Naphade, M.R., Smith, J.R.: Semantic representation, search and mining of multimedia content. In: ACM KDD. (2004) 641–646
29. Merler, M., Bert Huang, L.X., Hua, G., Natsev, A.: Semantic model vectors for complex video event recognition. IEEE Transactions on Multimedia (2011)
30. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: A large video database for human motion recognition. In: ICCV. (2011)
31. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action recognition. ICCV (2007)