UCSB_UCR_VCG TRECVID 2012

Santhoshkumar Sunderrajan*, Mahmudul Hasan**, Niloufar Pourian*, Yingying Zhu***,
B.S.Manjunath* and Amit Roy Chowdhury***
*Dept. of Electrical and Computer Engineering, University of California, Santa Barbara, CA-93106
**Dept. of Computer Science and Engineering, University of California, Riverside, CA-92521
***Dept. of Electrical Engineering, University of California, Riverside, CA-92521

I. Instance Search Task

We submitted three runs of the Instance Search task. They are listed as follows:

- UCSB_UCR_VCG_1 : This automatic run uses Bag of visual word with MSER+SIFT and Dense sift using svm classifier on chi-sq kernel.
- UCSB_UCR_VCG2: This automatic run uses Bag of visual word with MSER+SIFT and dense Color-sift based based SVM classifier on chi-sq kernel
- UCSB_UCR_VCG3: This automatic run uses Bag of visual word with MSER+SIFT and based svm classifier on histogram intersection kernel.

Discriminative reranking using a SVM classifier significantly improved the results. Compared to Histogram Intersection, Chi-square distance metric performed well on some of the queries.

Bag of visual word model was effective in retrieving location based topics. However, with discriminative reranking(an offline query expansion) approach significantly improved the results.

The discriminative classifiers learned from the internet images did not scale well on the test dataset. This poses a question on how to transfer model from one domain to another domain.

II. Surveillance Event Detection Task

We have submitted only one run named SED12_UcsbUcrVcg_interactiveED_EVAL12_2 of the Surveillance Event Detection Task.

We employ two different strategies to detect these activities based on their characteristics. Activities like CellToEar, Embrace, ObjectPut, and Pointing are the results of articulated motion of human parts. Therefore, we employ local spatio-temporal interest point (STIP) feature based bag of words strategy for these activities. Visual vocabularies are constructed from the STIP features and each activity is described by the histograms of visual words. We also construct activity probability map for each camera-activity pair that reflects the spatial distribution of an activity in a camera. We train Gaussian kernel discriminative classifier using SVM for each camera-activity pair. During evaluation we employ sliding window based technique. We slide spatio-temporal cuboids in both spatial and temporal direction to find a likely activity. The cuboid is also described by the histograms of visual words and final decision is made using the SVM classifier and the activity probability map.

For the ctivities like PeopleMeet, PeopleSplitUp, and PersonRuns, the characteristics of trajectories of persons of interest in the activities are discriminative. For instance, trajectories of PeopleMeet converge along time while those of PeopleSplitUp diverge along time. Therefore, we use track-based string of feature graph (SFG) to recognize these activities.

Discriminative Reranking based Video Object Retrieval

Santhoshkumar Sunderrajan*, Niloufar Pourian*, Mahmudul Hasan**, Yingying Zhu***,
B.S.Manjunath* and Amit Roy Chowdhury***
*Dept. of Electrical and Computer Engineering, University of California, Santa Barbara, CA-93106
**Dept. of Computer Science and Engineering, University of California, Riverside, CA-92521
***Dept. of Electrical Engineering, University of California, Riverside, CA-92521

***Abstract*—For the instance search task, we are given a set of query images with the corresponding textual meta-data and objects masks to retrieve video shots containing query objects from FLICKR video database. We extract meaningful regions in the key-frames using Maximally Stable Extremal Regions (MSER) and use SIFT descriptors for representation. We use standard Bag of visual Word (BoW) model to represent database images. Additionally, we crawled training images for each query topic using the textual meta-data from Google and FLICKR images databases to train a discriminative classifier using Support Vector Machines (SVM). We use a discriminative model to re-rank candidate images obtained by initial BoW search. The experimental results demonstrates the efficacy of the overall system. Finally, we highlight the need for domain adaptation when the source and target domains are completely different.**

I. Introduction

The aim of the instance search task is to retrieve video shots containing a particular query topic that is specified by multiple images, their associated masks marking the area of interest, and a textual meta-data. Retrieving an object in a database of images is a challenging task because an object's visual appearance may completely vary due to the changes in viewpoint, scale changes, lighting, and occlusion. For this reason, region extraction and descriptors are required to be built with some degree of invariance to viewpoint and illumination conditions.

TRECVID-2012 Instance Search task (INS) is a pilot task that concentrates on evaluating several algorithms for video object instance retrieval [6]. Videos shots are created from FLICKR video database. Participants are given with twenty-one query topics to retrieve from the given video database. The given query images appear in one or more video shots and the task is to retrieve video shots that contain the object of interest.

In previous years, because of the missing labels in the testing dataset, most of the participants used Bag of visual Words model in combination with some form of nearest neighbour search. In contrast, we used a combination of unsupervised retrieval with a discriminative re-ranking strategy. For supervised training, we crawled training examples from Google and FLICKR image databases using the textual meta-data available with query topics.

The rest of the paper is organized as follows: Section II discusses the general framework for video object retrieval. Section III discusses about discriminative re-ranking for improving the retrieval task in an efficient manner. Section IV demonstrates the results of experiments and finally we conclude in section V

II. Video Object Retrieval with Bag of Visual Words

We follow the standard BoW retrieval framework described in [9]. We retrieve key frames for the video database and extract Maximally Stable Extremal Regions (MSER) [5] and describe the regions using SIFT [4]. Finally, we represent the images with Bag of Visual words model using the dictionary trained from the TRECVID 2011 Instance Search task dataset (BBC Videos). We retrieve similar images from the testing database using chi-square distance matching and finally we re-rank the candidate list using a discriminative classifier trained from an auxiliary dataset. Figure II shows the complete framework used for the instance search task. Rest of this section explains each of the above mentioned steps in detail and discriminative re-ranking is explained in section III.

A. Key-frame Extraction

We extracted key-frames in the training dataset (BBC videos from 2011 task) using the FFMPEG utility. For the test dataset, we sampled images every 15 frames and the test database consisted of $223,141$ key-frame images.

B. Region Extraction and Feature Descriptors

For every image in the training and testing databases, we extracted Maximally Stable Extremal Regions as described in [5]. These are the regions for which the area is approximately stationary as the intensity threshold is varied. We used SIFT descriptors to represent each of these extracted regions.

C. Codebook Generation

For generating codebook from the training images, we randomly chose one million SIFT descriptors extracted from various regions in the entire training database. We used approximate k-means clustering along with the stop list criteria to obtain the final codebook.

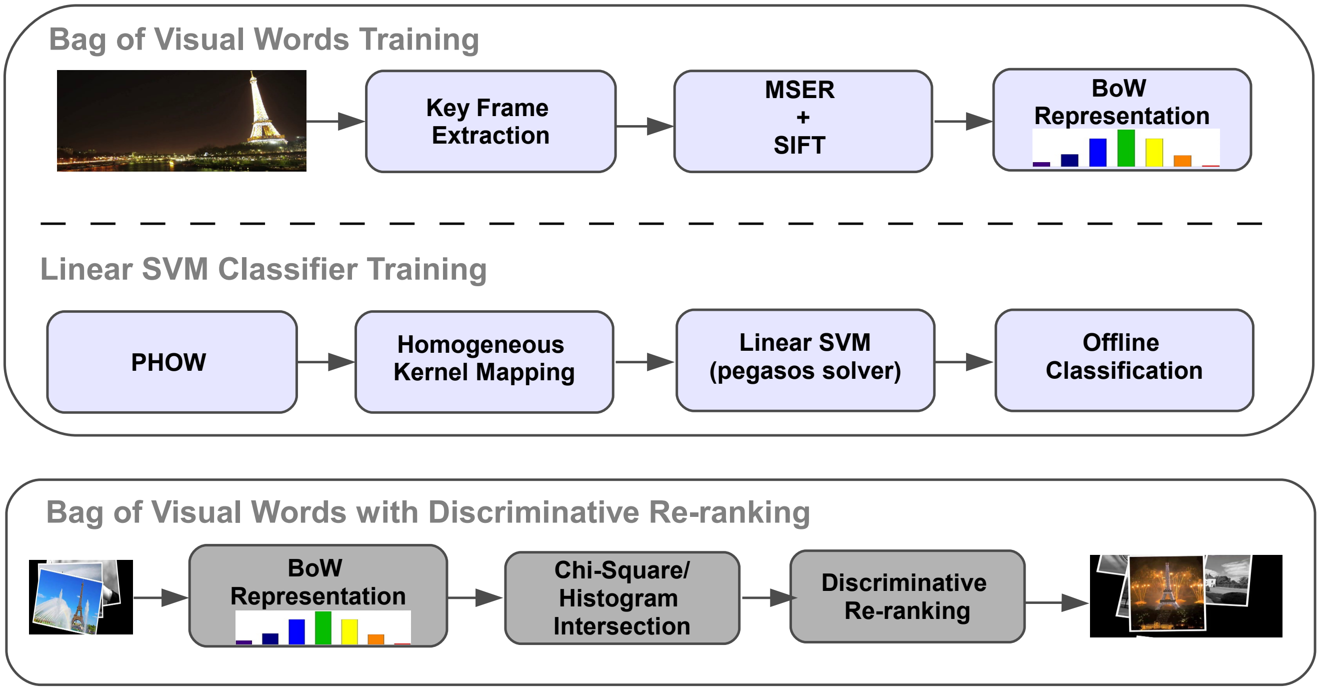

Bag of Visual Words Training
Key Frame Extraction
MSER + SIFT
BoW Representation
Linear SVM Classifier Training
PHOW
Homogeneous Kernel Mapping
Linear SVM (pegasos solver)
Offline Classification
Bag of Visual Words with Discriminative Re-ranking
BoW Representation
Chi-Square/ Histogram Intersection
Discriminative Re-ranking

Fig. 1. Video object retrieval with discriminative re-ranking framework.

1) Approximate k-means Clustering: In typical k-means clustering algorithm, a great amount of computation time is spent in finding distances between the points and cluster centres. In approximate k-means clustering, an approximate neighbour method using k-d tree data structure is used [7]. The algorithm complexity of a single k-means iteration is reduced considerably. For our computation, we chose a larger dictionary size (K = $10,000$) for a better performance.

2) Stop List Criteria: Using a stop list strategy the most frequent words that occur in most of the images are discarded. These are the noisy features that do not provide any useful information. In addition to this, we also removed least frequently occurring features from the codebook list. For our experiments, we discarded top 5% and bottom 10% of the codebooks entries. We finally ended up with a codebook size of K = $8,500$.

D. Retrieval

We represented each image by a normalized histogram using the codebook obtained from the training dataset (BBC Videos). For a given query image we computed the BoW model using a similar strategy and compared with the database images using Chi-square and Histogram intersection metrics [3]. We ordered images based on the matching score. Since the query image and database images vary to a great extent, we performed a discriminative re-ranking to enhance the performance of the retrieval as discussed in section III.

III. Discriminative Re-ranking

In the given test dataset, the object might appear in arbitrary location and undergo arbitrary distortion and transformation when compared to the query images. Hence in order to fully capture the query object characteristics, we need some form of query expansion or re-ranking mechanism [1]. We adopt a discriminative re-ranking mechanism by modelling the query object characteristics explicitly using the images obtained from the internet in an offline manner.

A. Crawling Training Images

Using the textual meta-data available with the query images, we auto-crawled 200-300 images per topic from the Google and FLICKR image databases.

B. Discriminative Learning

With the images obtained from the web, we first extracted Dense SIFT for every image and then formed PHOW descriptors [2]. We encoded the PHOW descriptor using homogeneous kernel mapping. Finally, we trained a linear classifier "1 v/s all" classifier using SVM with the Pegasos solver [8]. We classified all the key frames using the model learned for query type and associated a likelihood score. We used this score to re-rank the candidate list obtained from BoW model based retrieval. Rest of this section explains each of the above mentioned steps in detail.

1) Feature Extraction: We extracted dense SIFT feature with a step size $= 4$ pixels i.e. the grid at which features are extracted. We used k-means clustering to generate a

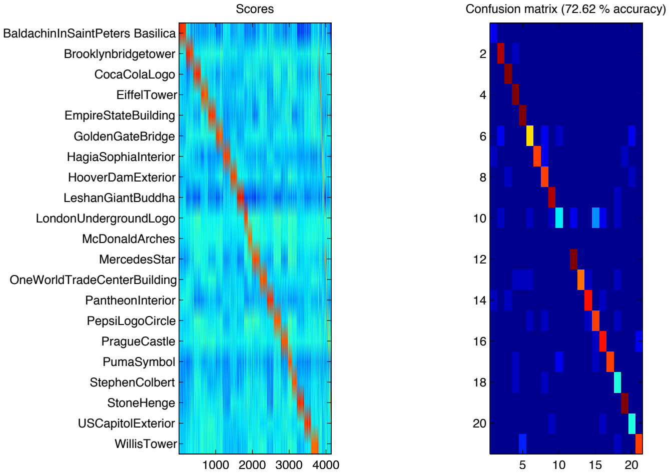


Fig. 2. Classification scores and the corresponding confusion matrix for different query topics obtained using SVM classifier on PHOW representation.

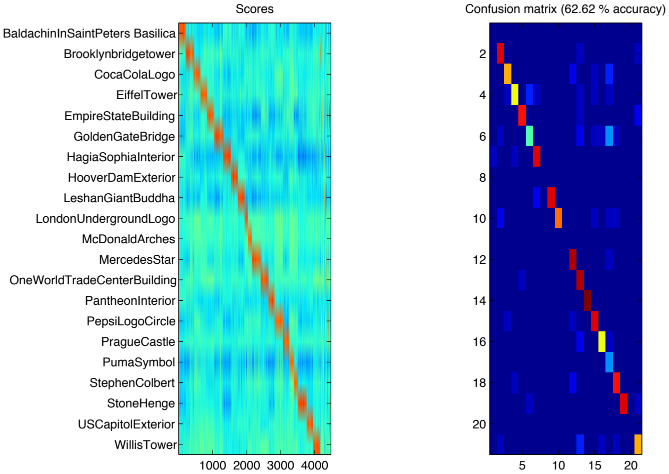


Fig. 3. Classification scores and the corresponding confusion matrix for different query topics obtained using SVM classifier on color based PHOW representation.

codebook of size C= 300. The PHOW features are a variant of dense SIFT descriptors, extracted at multiple scales [2]. Additionally, for another set of experiments, a color version, named PHOW-color, we extracted descriptors on the three HSV image channels and stacked them together. For the color version, we used a step size of 7 pixels to extract dense SIFT feature.

2) Homogeneous Kernel Mapping: The homogeneous kernel map is a finite dimensional linear approximation of homogeneous kernels, including the intersection, chi-square, and Jensen-Shannon kernels [10]. These kernels are particularly useful for descriptors represented using histogram. For our experiments, we used a chi-square kernel.

3) Linear Support Vector Machine Classifier: We trained a linear SVM classifier using the Pegasos solver described in [8]. Pegasos is a simple and efficient iterative algorithm for the solving the optimization problem for SVM. The run-time of the solver does not depend on the size of the training dataset and hence it can be scaled to large datasets easily. We used chi-square kernel with homogeneity of kernel set to 0.5. In our experiments, we used two different classifiers trained on gray scale PHOW and a color version of it (HSV). Figures 2 and 3 show the classification scores and confusion matrix for different query topics for the two discriminative models trained from the internet images.

4) Off-line Classification: Since the model is trained by querying images from the internet using the textual meta-data, we reduced the runtime for retrieval by classifying each of the key frames using the model learned for all 21 query types in an off-line manner.

IV. Experiments

For TRECVID evaluation, we submitted three runs and each of the runs is discussed in detail in the following sub sections. Figure 4 shows number of hits per 1000 candidates retrieved for various runs compared to ground truth (gt) used for evaluation. As seen in figure 4, all three runs perform equally well for global location based queries such as $9051, 9052$ etc.

A. ***Run-1***: *BoW with Chi-square distance*

For the first set of experiments, we used the Bag of visual words model with Chi-square distance for matching the given query images with key-frames in the test database and re-ranked using the classification scores obtained by SVM classifier trained on internet images. We combined the unique results from different sub-queries and ordered them based on the matching scores. Figure 5 shows the number of hits per 1000 candidates retrieved for run-1 versus the ground truth (gt) and the best result (best). Since the BoW is a global model, it is well suited for location based queries. We attribute the difference in performance compared to the best performance to the way in which the key-frame extraction is done. Since we extracted only 3 key-frames per video shot, we might have missed some object instances that appear for a small number of frames in a given video shot.

B. ***Run 2***: *BoW with Chi-square distance + SVM on Color PHOW*

For the second set of experiments, we used the Bag of visual words model with Chi-square distance for matching the given query images with the key-frames in the database and re-ranked using the classification scores obtained by SVM classifier trained on internet images. We combined the unique results from different sub-queries for each topic and ordered them based on the matching scores. Figure 6 shows Average Precision for different queries (topics). Interestingly, one would expect the discriminative model trained on an exemplary dataset would perform well on the test dataset, however, due to inherent difference in feature distribution, learned model does not fair so well in the test dataset. Compared to run-1, for query 9048 (Mercedes star logo), run-2 performs well due to the context information used while

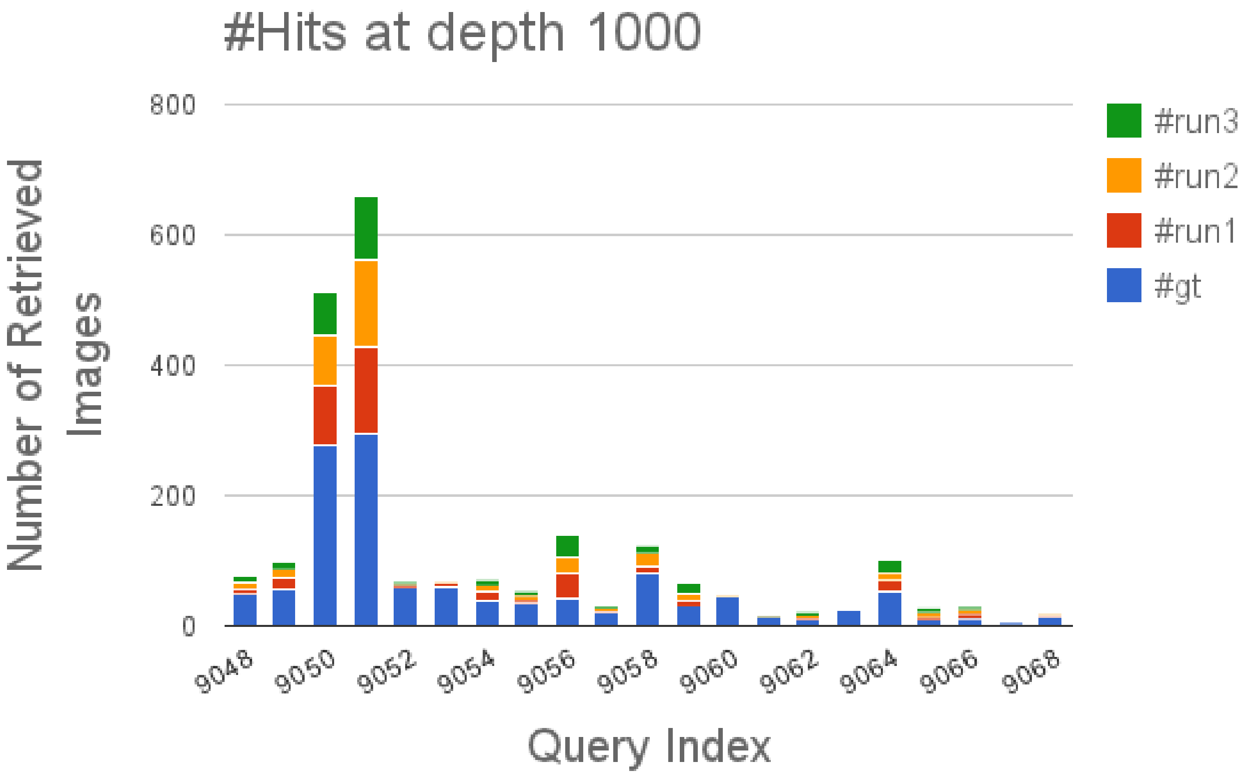


Fig. 4. Number of hits at the depth of 1000 images for Run-1, Run-2 and Run-3 compared to the ground truth (gt). Best in color.

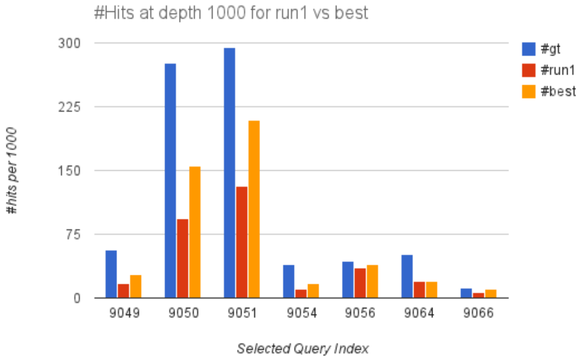


Fig. 5. Number of hits at the depth of 1000 images for Run-1 compared to the ground truth (gt) and the best for the selected query topic

learning the discriminative model. Figure 7 illustrates how the context information is helpful in improving the retrieval using discriminative re-ranking.

C. ***Run 3***: *BoW with Histogram Intersection + SVM trained on PHOW*

For the third set of experiments, we used the Bag of visual words model with Chi-square distance for matching the given query images with the key-frames in the database and re-ranked using the classification scores obtained by SVM classifier trained on internet images with color PHOW features. Results obtained from run-3 are similar to that of run-2 since the discriminative model learned from the internet images did not adapt well in the test domain.

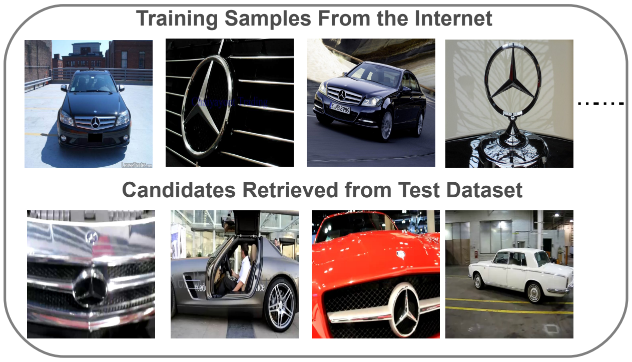

Training Samples From the Internet
Candidates Retrieved from Test Dataset

Fig. 7. Top row shows some of the training images crawled from for learning the discriminative classifier. Bottom row shows some of the retrieved images at a depth of 1000. As seen, the contextual information plays a major role in improving the accuracy.

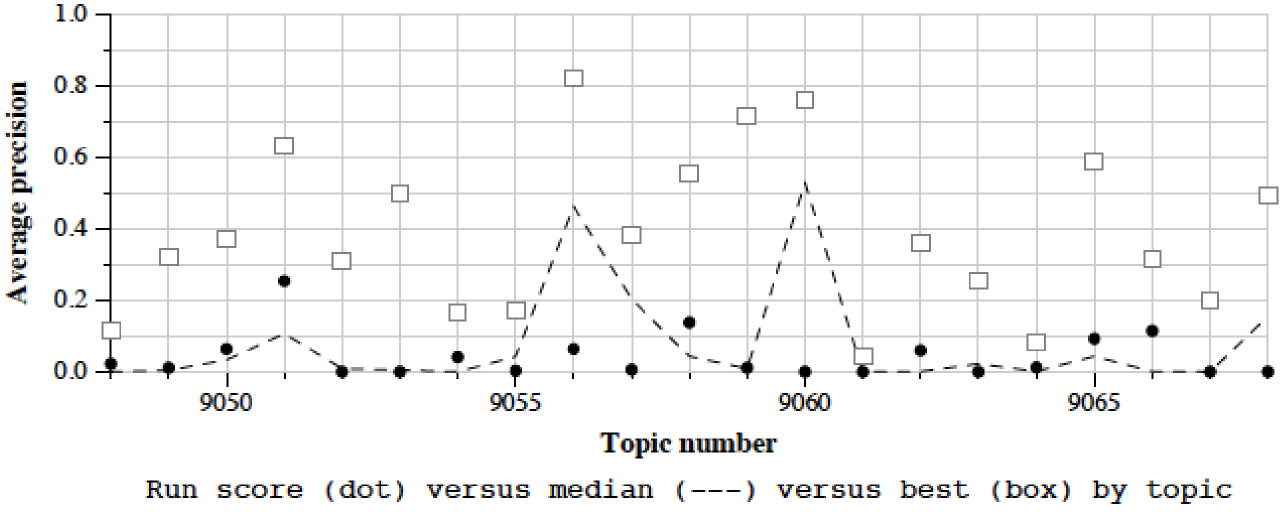


Fig. 6. Shows mean average precision obtained for different topics with Run-2.

V. Conclusion

For the instance search task, a bag of visual words model (BOW) based retrieval strategy is effectively coupled with discriminative linear classifiers for re-ranking. The training dataset from 2011 instance search task is used for learning the dictionary and the learned dictionary is used on the test dataset for obtaining the image descriptors. Because of the missing labels in the test dataset, we crawled additional training images from Google and FLICKR image databases using textual meta-data available along with query images to train a linear classifier using Support Vector Classifier (SVM). The classification margin is used for scoring the query class likelihood for every key-frame image sampled from the video shots.

Initial list of candidate images are retrieved using BOW model and chi-square distance metric, and then SVM classifier learned from the internet images is used to re-rank the candidates. In order to reduce the overall retrieval time, linear classifiers are run against the test database in an offline manner. Experimental results show better performance for global queries that occupy considerable portion of the image plane. Also, due to inherent differences in the data distributions of the training and test datasets, the discriminative model learned from web images did not perform as good as it performed on the training source domain. In the future, we plan to perform further research on Domain adaptation i.e. how to transfer models from the source domain to the target domain automatically.

Acknowledgment

The authors would like to thank Yue Cao and Michael Shabsin for the help during the course of the task.

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2911–2918. IEEE, 2012.

[2] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[3] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005.

[4] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[5] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[6] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Barbara Shaw, Wessel Kraaij, Alan F. Smeaton, and Georges Quenot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[7] 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.

[8] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning*, pages 807–814. ACM, 2007.

[9] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.

[10] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480–492, 2012.

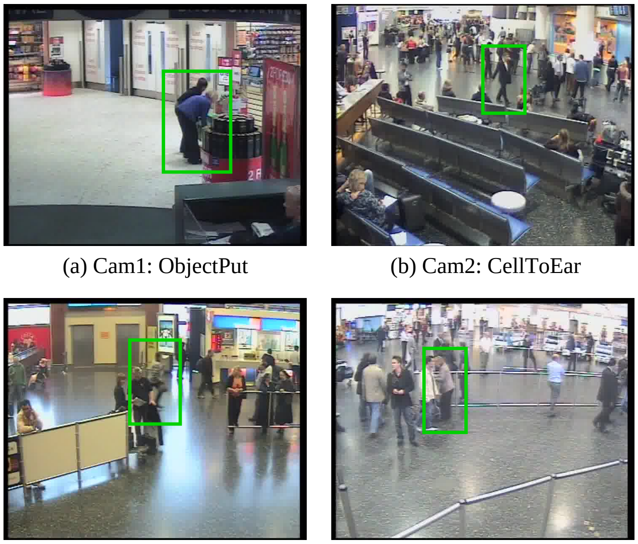
Activity Analysis in Unconstrained Surveillance Videos

Mahmudul Hasan*, Yingying Zhu**, Santhoshkumar Sunderrajan***, Niloufar Pourian***,
B.S.Manjunath*** and Amit Roy-Chowdhury**
*Dept. of Computer Science and Engineering, University of California, Riverside, CA-92521
**Dept. of Electrical Engineering, University of California, Riverside, CA-92521
***Dept. of Electrical and Computer Engineering, University of California, Santa Barbara, CA-93106

***Abstract*—We detect seven activities defined by TRECVID SED task such as CellToEar, Embrace, ObjectPut, PeopleMeet, PeopleSplitUp, PersonRuns, and Pointing. We employ two different strategies to detect these activities based on their characteristics. Activities like CellToEar, Embrace, ObjectPut, and Pointing are the results of articulated motion of human parts. Therefore, we employ local spatio-temporal interest point (STIP) feature based bag of words strategy for these activities. Visual vocabularies are constructed from the STIP features and each activity is described by the histograms of visual words. We also construct activity probability map for each camera-activity pair that reflects the spatial distribution of an activity in a camera. We train a discriminative SVM classifier using Gaussian kernel for each camera-activity pair. During evaluation we employ sliding window based technique. We slide spatio-temporal cuboids in both spatial and temporal direction to find a likely activity. The cuboid is also described by the histograms of visual words and final decision is made using the SVM classifier and the activity probability map. For the activities like PeopleMeet, PeopleSplitUp, and PersonRuns, the characteristics of trajectories of persons of interest in the activities are discriminative. For instance, trajectories of PeopleMeet converge along time while those of PeopleSplitUp diverge along time. Therefore, we use track-based string of feature graph (SFG) to recognize these activities. Results of our experimental runs on the evaluation videos are comparable with other participants. Our performances in all the activities are among the top five teams.**

I. Introduction

Rapid advancement in technologies and low cost of cameras and communication devices make it real easy to deploy complex surveillance system in various scenarios. The goal of these types of intelligent systems is to analyze huge amount of video data and to find useful information, which are valuable for safety and security. Automating these processes without manual intervention imposes a great challenge to the computer vision community. Even though researchers commit a lot of efforts in video content analysis and human activity recognition, it is still far beyond to reach performance close to human level. To expedite this process, each year National Institute of Standardization and Technology (NIST) organizes several computer vision related challenges under the banner of TRECVID that includes Surveillance Event Detection (SED). In this task, seven human activities such as CellToEar, Embrace, ObjectPut, PeopleMeet, PeopleSplitUp, PersonRuns, and Pointing are needed to be recognized in large video corpus [5]. TRECVID ([16], [17]) provides development video corpus with ground truth annotations to train the system along with test video corpus for the final evaluation of the system. These videos were captured by five surveillance cameras installed in London Gatwick Airport. Four of them are shown in Figure 1 with peoples taking part in different activities.



(a) Cam1: ObjectPut

(b) Cam2: CellToEar

(c) Cam3: PersonRuns

(d) Cam5: Pointing

Fig. 1: Four different human activities in four cameras. Activities are obscured by clutter, low resolution, background noises, etc.

Human activity recognition in unconstrained real world videos captured by the surveillance cameras is challenging due to several difficulties such as background noise, clutter, difference of viewpoints, large crowd, illumination variation, occlusion, etc. as illustrated in Figure 1. Many approaches were adopted to mitigate these difficulties and to efficiently recognize human activities. Among these, spatio-temporal approaches are particularly popular because of their effectiveness [1]. In these approaches, human activities are modeled as 3-D volume in spatio-temporal dimension and spatio-temporal features are extracted from these volumes. Video frames are concatenated along the time axis to construct 3-D volumes. In a typical spatio-temporal volume based approach, a 3-D spatio-temporal model for each activity based on the training videos is constructed. During testing, similar 3-D spatio-temporal volumes are constructed from the unlabeled video. Sliding window based technique is used to construct 3-D volumes for large video corpora. Then, different similarity measures such as template matching and discriminative classifiers are used to find the best match to this unlabeled video with modeled activities ([9], [10]). In addition to pure 3-D volume based approaches, spatio-temporal trajectory based approaches are also widely used to recognize human activities ([11], [12]). In these approaches, humans are represented by points in the spatio-temporal volume. These points usually correspond to special joint positions of a person at each video frame or they can be interest points delineating high variations in both spatial and temporal directions. Interest points are detected using Harris operator, HoG [6], SIFT algorithm [2], etc. These points are tracked between subsequent frames to obtain a trajectory of points. Features are extracted from the trajectories and template matching or classifiers are used to label the unknown videos. This method is superior over the pure 3-D volume based approach because it can cope with the dynamic properties of realistic activities.

In a spatio-temporal local feature based approach e.g. [7], spatio-temporal local features or interest points are extracted from 3-D volumes to represent and recognize activities. Spatio-temporal interest points can be found by extending 2-D interest point detection algorithm. Laptev et al. [8] proposed Spatio-Temporal Interest Point (STIP) feature which is an extension to the 2D Harris corner detector. It detects points with high intensity variations in both spatial and temporal dimensions. In [4], the MoSIFT feature is proposed, which is a 3-D extension of SIFT [2]. It detects spatially distinctive interest points, where substantial motion exists between consecutive frames. In spatio-temporal local feature based approaches, spatial and temporal relationships among detected interest points are ignored, typically called as bag-of-words. Each of the features is vector quantized to a visual word and the video is represented as the histograms of visual words. Since bag of word loses important temporal and spatial information, it is not efficient in recognizing complex human activities [13]. In [13], Gaussian Mixture Model (GMM) is used to describe the distribution of computed interest points. Generally, discriminative classifiers trained with Support Vector Machine (SVM) are used to label unknown activities. To recognize activities in complex scenarios, cascaded and hierarchical SVMs are also used.

In this work, we employ two different methods for detecting seven TRECVID defined activities based on their characteristics. Activities like CellToEar, Embrace, ObjectPut, and Pointing are the results of articulated motion of human parts belonging to one or more persons taking part that activity. For this type of activities, bag of visual word (BoW) based strategy is used to represent video clips containing the activity, where spatio-temporal interest points (STIP) are used as the low level feature. After generating STIP features from the segmented video clips, STIP features only inside the activity regions are considered in order to reduce noise during training. We train binary Gaussian kernel discriminative classifier using SVM for each camera-activity pair separately. During evaluation, overlapping spatio-temporal cuboids slid through the video frames. Sizes of these cuboids are determined from the training data and they are different for each camera-activity pair. Motion regions and event bounding boxes computed from the training videos are used to calculate activity probability maps for each camera-activity pair. Sliding of spatio-temporal cuboids is performed in both spatial and temporal directions. Each spatio-temporal cuboid is described by the histogram of visual words and a trained discriminative classifier is used for each label to compute the likelihood of each histogram for each activity label. Activity probability map is used to make the final decision by re-weighting the probability, which is effective to reduce false alarms.

For activities like PeopleMeet, PeopleSplitUp and PersonRuns, the characteristics of trajectories of the persons of interest in the activities are discriminative. For instance, trajectories of PeopleMeet converge along time while those of PeopleSplitUp diverge along time. Therefore, we use track-based SFG to recognize these events. For PeopleMeet and PeopleSplitUp, the current system uses training instances from VIRAT Dataset release 1. We use background subtraction and mean-shift to track the moving objects. Trajectories with length less than 20 frames are omitted for the detection of PeopleSplitUp and PersonRuns and with less than 5 frames are omitted for the detection of PersonRuns. In the experiments, we compare the characteristics of each pair of trajectories with the training instances. The confidence score of a testing instance belonging to a certain activity class is the average similarity scores between the testing instance and the training instances of that activity class generated by the SFG matching algorithm. Activity maps are used to re-weight the confidence scores.

The rest of the paper is organized as follows. In Section II we present our spatio-temporal cuboid based approach. Track bases SFG is discussed in Section III. Experiment results and methods are described in Section IV, while we conclude the paper in Section V.

II. Spatio-temporal Cuboid Based Approach

The general framework of spatio-temporal cuboid based approach for detecting activities like CellToEar, Embrace, ObjectPut, and Pointing is shown in Figure 2. It includes extracting STIP features from the activity video clips; clustering STIP features to obtain a visual vocabulary for each camera; encoding each video clip containing a possible activity as the bag of STIP features; and training discriminative SVM classifier with Gaussian kernel for each camera-activity pair. In this Section, we describe these steps. We also describe the construction of activity probability map and spatio-temporal cuboid sliding through the video frames in the subsequent subsections.

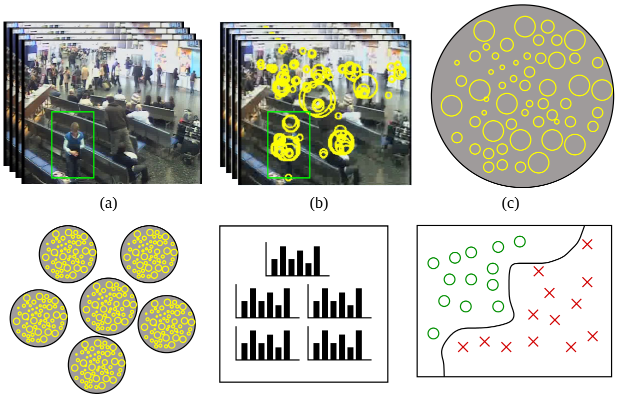


Fig. 2: Feature extraction and classifier design. (a) Event video clip with spatial extent of the activity region, (b) Generating STIP features and collecting those belonging to activity region, (c) Bag of STIP features contains all the STIPs of all events, (d) Clustering the STIP features into visual words using k-mean clustering algorithm, (e) Representation of video clips using histograms of visual words, and (f) training discriminative classifier using SVM.

A. *Feature extraction and classifier design*

In this work, we use space-time interest point (STIP) to describe each video segment. STIP detector uses spatio-temporal extension of 2D Harris corner detector to find the center locations of local spatio-temporal patches. These center locations are called as interest points and capture large variations along both the spatial and the temporal directions. For each interest point, associated spatio-temporal patches are described by local appearance features. In our implementation, we compute two patch descriptors of local appearance features such as histogram of oriented gradients (HOG) and histogram of optical flow (HOF) and concatenate them. These features are local and based on the appearance at particular interest points, and are invariant to image scale and rotation.

TRECVID provides ground truth annotations of the activities in the development video corpus. These annotations contain only the temporal extent of the activities. We use these annotations to segment video clips containing activities from the video corpus. Each video clip contains a particular activity and STIP features are computed for each video clip. However, the spatial extent of an activity occupies only a smaller portion in the video frame. For this reason, STIP features collected from the whole video frame do not accurately represent an activity. It contains a lot of noises that correspond to STIP features outside of the activity region. To alleviate this problem, we manually draw bounding boxes around the activity regions in each frame of the video clip using TRECVID recommended software named Viper. We represent an activity by the STIP features collected from the inside of these bounding boxes. We put all STIP features collected from all the activities in a bag and build a visual vocabulary of STIP features using k-mean clustering algorithm. The size of the vocabulary is empirically set to 400. Thereupon, each STIP is assigned a visual word label and each activity is represented by histograms of visual words. Above procedure is performed for each of the five cameras separately. The number of activities we segmented from development video corpus for the purpose of training is shown in Table I.

TABLE I: Number of video activities used for each type of event for each camera during training.

	CellToEar	Embrace	ObjectPut	Pointing	Total
CAM1	25	19	199	277	520
CAM2	94	82	280	304	760
CAM3	107	240	185	291	823
CAM4	2	2	9	18	31
CAM5	51	50	59	230	390
Total	279	393	732	1120	2524

For activity classification, we use SVM with Gaussian kernel. We train a binary classifier for each camera-activity pair. For a particular camera and an activity, we use all other activities in this camera as the negative examples. We use LIBSVM software available online to train the classifiers [3]. Five-fold cross validation procedure is employed to avoid over fitting. Grid search strategy is used to find the optimal parameters of the Gaussian kernel.

B. *Evaluation*

1) Activity probability map: Development and evaluation videos for TRECVID SED task were obtained from five static camera installed in London Gatwick Airport. Each of the five camera view represents public scenes, where people take part in different activities including the seven activities of interest. These activities tend to occur more in some regions of the video frame, which are generally different for different cameras and activities. This prior information from the training videos is utilized in the evaluation phase to reduce the number of false alarms. Hence, we construct activity probability map for each camera-activity pair as shown in Figure 3. Each pixel of this map is a probability that signifies the chance of occurring an activity in the surrounding regions. In order to construct this activity probability map we employ two different methods: (i) motion map and (ii) activity map and integrate them. Gaussian mixture model (GMM) based background subtraction algorithm is used to find the motion regions in the video frame. A new pixel is considered as a background pixel if it can be described by the model density. Morphological dilation operation on the foreground gives us a set of blobs. We surrounded these blobs by the bounding boxes. Furthermore, we track blobs based on overlapping regions in the consecutive frames to get a set of trajectories, $\{T_p\}$. We construct a motion map M_m^c for each camera $\{c = 1 \dots 5\}$. Pixels of this map is defined as follows:

$$M_m^c(i,j) = \sum_{\{T_p\}} \sum_{\{B_q \in T_p\}} \mathbf{1}_{(T_p,B_q)}(i,j)$$

Where $\{T_p\}$ is the set of trajectories and $\{B_q \in T_p\}$ is the set of bounding boxes correspond to a trajectory. Indicator function $\mathbf{1}_{(T_p,B_q)}(i,j)$ is defined as follows:

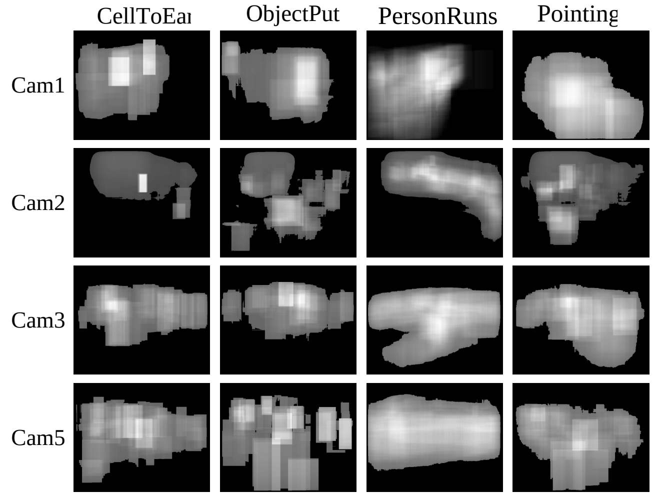


Fig. 3: Activity probability map for each camera-activity pair.

$$\mathbf{1}_{(T_p,B_q)}(i,j) = \begin{cases} 1 & \text{if } (i,j) \in (T_p,B_q) \\ 0 & \text{if } (i,j) \notin (T_p,B_q) \end{cases}$$

Similarly, we construct an activity map $M_o^{c,a}$ for each camera and activity pair $\{c = 1,\ldots,5, \quad a = 1,\ldots,7\}$. TRECVID provides only temporal extent of an activity in the ground truth annotation files. In addition to this, we manually draw bounding boxes around each activity. These bounding boxes are used to construct activity map. Final activity probability maps are derived by integrating motion and activity map as follows:

$$M^{c,a} = \alpha M_m^c + (1-\alpha) M_o^{c,a} \quad \text{for}\{c = 1\ldots5, \quad a = 1\ldots7\}.$$

2) Sliding cuboid: In order to find a likely activity in the evaluation videos, we search the whole video using overlapping spatio-temporal cuboids as shown in Figure 4. We slide the cuboids in both temporal and spatial directions. We use fixed size cuboid for a particular camera-activity pair. Size of these cuboids are shown in Table II, which are computed from the ground truth annotations of the development video corpus. For each cuboid, we collect all the STIP features that are located inside the cuboid boundary. We assign a visual word label to each STIP feature. These visual word vocabularies were pre-computed during training. Thus, each cuboid is described by histograms of visual words. Then, trained binary classifier is used to label this cuboid of being a particular activity with a probability. We re-weight this probability using the activity probability map. We know the spatial location of this cuboids and the name of the camera-activity pair. We obtain the probability of occurring an activity in this spatial location from the activity probability map of corresponding camera-activity pair. We use this probability as the prior information to re-weight the original probability. Overall probability of a cuboid will be less that corresponds to the region in the frame where the probability of occurring an activity is less. It helps to reduce the number of false alarms.

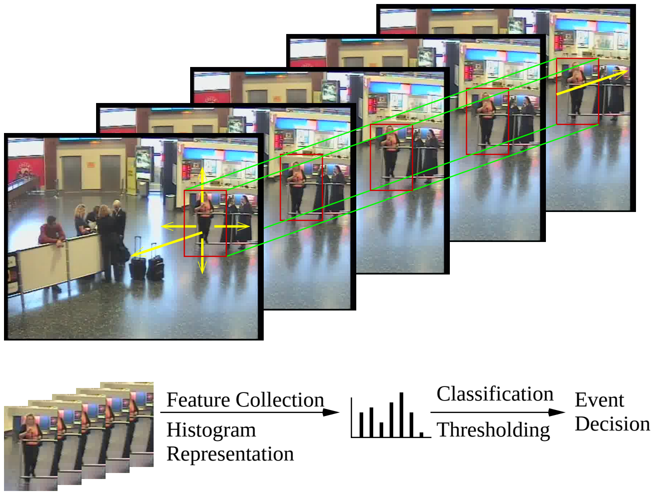


Fig. 4: Scanning videos using cuboids.

TABLE II: Average number of frames, height, and width in pixels of the cuboids for each camera-activity pair obtained from the training videos.

	CellToEar	Embrace	ObjectPut	Pointing
CAM1	49/200/97	68/246/145	29/208/135	78/215/98
CAM2	103/100/56	219/104/63	26/125/105	90/94/50
CAM3	19/167/77	120/214/124	17/145/84	68/140/75
CAM4	18/287/149	194/306/187	6/288/209	34/208/139
CAM5	22/176/81	134/168/90	20/149/85	65/160/78

III. Track-based Event Detection

In this section, we focus on events whose motion patterns can be captured from the underlying tracks (e.g., PeopleMeet, PeopleSplitUp, PersonRuns) and need to explore the relationship between two active persons. The characteristics of tracks of persons of interest in the events of PeopleMeet, PeopleSplitUp and PersonRuns are discriminative. For instance, tracks of PeopleMeet converge along time while those of PeopleSplitUp diverge along time. Therefore, we use track-based SFG method [14] to detect these events.

A. Tracking

We use background subtraction [15] and mean-shift tracker to generate tracks of moving objects. This is a simple tracker without trajectory association.

B. Track-based SFG Event Detection

In this subsection, we describe how to detect PeopleMeet and PeopleSplitUp based on the obtained tracks. For PersonRuns, a more heuristic method is used. Motion statistics such as the velocity of a trajectory and the range of the trajectory are used as feature descriptors for detection, the detection method is described in the experiment section.

1) Feature Descriptor: After obtaining the tracks of moving object, we develop background subtraction based motion features for each trajectory in order to detect PersonRuns. Pair-wise track-based features are developed for each pair of tracks, in order to detect PeopleMeet and PeopleSplitUp.

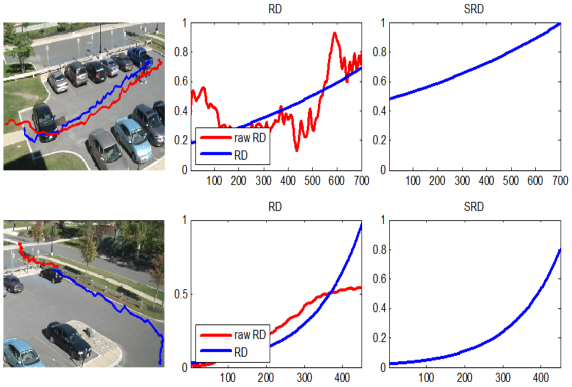


Fig. 5: Example of RD and SRD of two tracks. The images show sample frames of two people walking together (top) and person leaving a vehicle (bottom) (only regions of interest are shown). The graph on the left shows the raw relative distance between the two tracks and the exponential fitting result in each case. The graph on the right shows the derivative of smoothed relative distance (SRD) in each case.

For PersonRuns, motion statistics such as the velocity of a trajectory and the range of the trajectory are used as feature descriptors for detection. For PeopleMeet and PeopleSplitUp, given two tracks, we introduce Slope of smoothed relative distance (SRD) to describe the converge/diverge trends of the two tracks. SRD of a pair of tracks is the change of their relative distance smoothed along time, which captures the interaction trends between the two tracks.

Relative distance of two tracks is obtained first. Break-points, where the trend of interaction changes (e.g. from approaching to dispersing) are detected and used to segment the RD descriptor. Break-points are defined as those local extrema of the relative distance sequence whose distance with the immediate previous extrema is greater than a pre-determined threshold. Exponential curve fitting is utilized to smooth out the segments in the resulting the RD descriptor. Let $\tilde{t_i}$ and $\tilde{t_j}$ be the tracks of object i and j respsectively, and $p_i(t) = [x_i(t) \;\; y_i(t)]$ and $p_j(t) = [x_j(t) \;\; y_j(t)]$ for $t = 1, 2, ...$ be the positions of objects i and j at time t. The relative distance of object i and j at time t is $d(t) = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}$. The detected break points $t_1, t_2, ..., t_n$ and the beginning and end points t_0, t_{n+1} segment the sequence of relative distance of the two objects into $n+1$ segments $rd(k)$ for $k = 0, 1, ..., n$. The RD and SRD features of tracks of i and j at time t are defined as

$$RD_{(i,j)}(t) = exp\_fit(rd(k)) \quad if \quad t_k < t \leq t_{k+1}, \tag{1}$$

$$SRD_{(i,j)}(t) = \frac{RD_{(i,j)}(t)}{dt}, \tag{2}$$

where exp_fit refers to fitting an exponential function to the specific rd sequence.

2) Track-based Feature Graph Matching: In the feature graph matching, tracks are segmented into tracklets by concatenated equal-length time windows (size of 5 frame is used in the experiment). Each tracklet forms a node in the feature graph. The edge features quantize the interaction between the two underlying objects. It is natural to use the smoothed Euclidean distance between individual track features of two tracklets as the node distance measurement, and the smoothed distance between the interacting features of two pairwise tracklets as the edge distance measurement.

Assume tracklet i belongs to the query video, and tracklet i' belongs to the testing video. Let $f^{SRD}_{\overrightarrow{ij}}$ be the concatenated SRD between i and j. For a feature graph Q in the query video and a feature graph P in the testing video, the node distance, edge distance, and elements of similarity matrix defined in [14] are specified as

$$d_n(i, i') = 0 \tag{3}$$

$$d_e(\overrightarrow{ij}, \overrightarrow{i'j'}) = \frac{\| f^{SRD}_{\overrightarrow{ij}} - f^{SRD}_{\overrightarrow{i'j'}} \|}{s} \tag{4}$$

$$\tag{5}$$

where s is the length of a tracklet. We are interested in only the interaction patterns of tracks involved in activities, so, ω_n is set to be zero.

IV. Experiments and Discussion

We use sliding cuboids to detect likely activities in the evaluation videos. Size of the these cuboids are different for each camera-activity pair. We determine these sizes by analyzing training videos as shown in Table II. We slide the cuboids both in temporal and spatial direction. We keep five frame temporal and twenty pixel spatial distance between two overlapping cuboids. As discussed above, each cuboid is described by histograms of visual words and SVM classifier is used to get the probability of occurring an activity corresponds to this cuboid. We re-weight this probability by multiplying it with the probability we get from the activity probability map. This final probability is thresholded to get the final decision. We select different thresholds for each camera-activity pair empirically.

For PeopleMeet and PeopleSplitUp, the current system uses training instances from VIRAT Dataset release 1. Tracks with length less than 20 frames are omitted for the detection of PeopleSplitUp and PersonRuns, less than 5 frames are omitted for the detection of PersonRuns. In the experiments, we compare the characteristics of each pair of tracks with the training instances. The confidence score of a testing instance belonging to a certain activity class is the average similarity scores between the testing instance and the training instances of that activity class generated by the SFG matching algorithm. Activity maps are used to reweight the confidence scores. We use a threshold of 0.5 to separate positive and negative instances. For PersonRuns, tracks with a length less than 5 frames, or the XY ranges of tracks are less than the average size of the bounding boxes, are tripped. we calculate the average velocity of each trajectory. Tracks with 5% highest velocity are classified as PersonRuns.

TABLE III: Final evaluation results.

	Inputs			Actual Decision DCR Analysis								Minimum DCR Analysis			
Title	#Targ	#NTarg	#Sys	#CorDet	#Cor!Det	#FA	#Miss	RFA	PMiss	DCR	Dec. Tresh	RFA	PMiss	DCR	Dec. Thresh
CellToEar	194	260	263	3	0	260	191	17.05229	0.985	1.0698	0.3002	0.06559	1.000	1.0003	0.682
Embrace	175	338	358	20	0	338	155	22.16797	0.886	0.9966	0.6002	7.67353	0.891	0.9298	0.801
ObjectPut	621	112	116	4	0	112	617	7.34560	0.994	1.0303	0.6502	0.06559	1.000	1.0003	0.805
PeopleMeet	449	1007	1068	56	48	959	393	62.89670	0.875	1.1898	0.5031	0.06559	0.998	0.9981	0.998
PeopleSplitUp	187	335	360	24	56	279	163	18.29842	0.872	0.9631	0.5011	13.96976	0.888	0.9575	0.785
PersonRuns	107	1827	1851	24	0	1827	83	119.82510	0.776	1.3748	0.5061	3.73839	0.963	0.9813	0.982
Pointing	1063	221	230	9	0	221	1054	14.49444	0.992	1.0640	0.5700	0.19676	0.999	1.0000	0.816

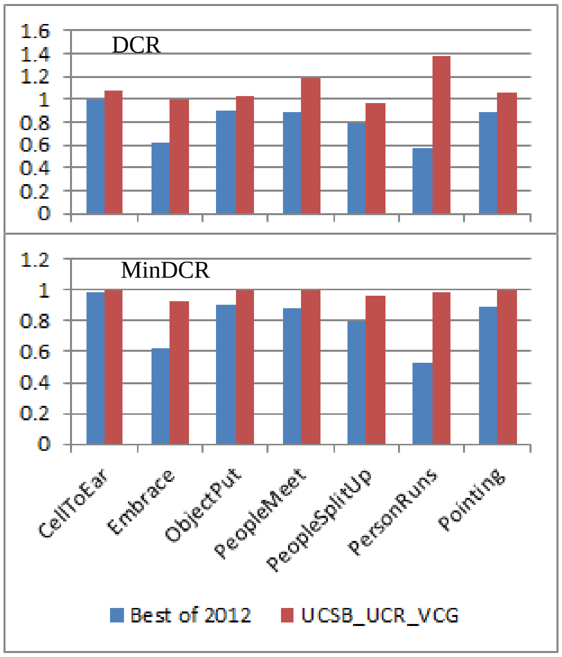


Fig. 6: Comparison between our results and the best results of 2012.

The final evaluation results of our system we got from TRECVID are shown in Table III. Comparison between our results and the best results of 2012 are shown in Figure 6.

V. Conclusion

In this paper, we described our approach for detecting seven activities such as CellToEar, Embrace, ObjectPut, PeopleMeet, PeopleSplitUp, PersonRuns, and Pointing defined by TRECVID SED task. We employed two different approaches based on the characteristics of the activities. In the first approach, we used STIP feature based bag of words to represent an activity. Gaussian kernel based discriminative classifier trained using SVM was used to label the unknown activities with the help of activity probability map. We used sliding cuboid to find the probable activities in the large videos. In the second approach, we used track-based string of feature graph (SFG) to recognize the activities like PeopleMeet, PeopleSplitUp, and PersonRuns. Results of our experimental runs on the evaluation videos are very comparable with other participants. Our performances in all the activities are among the top five teams.

Acknowledgment

The authors would like to express their best regard to Abir Das and Robert Tingle of UC, Riverside for the help during the course of the task.

References

[1] J. K. Aggarwal, and M.S. Ryoo, Human Activity Analysis: A Review, To appear in ACM Computing Survey.

[2] D. G. Lowe, Distinctive image features from scale-invariant keypoints, Int. Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001. Software http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[4] M.-y. Chen and A.Hauptmann. MoSIFT: Reocgnizing Human Actions in Surveillance Videos Carnegie Mellon University, 2009.

[5] National Institute of Standards and Technology (NIST): trecvidDetection.http://www.nist.gov/speech/tests/trecvid/2009/

[6] N. Dalal and B. Triggs. Histogram of oriented gradient for human detection. In CVPR, 2005.

[7] Schuldt, C. Laptev, and B. I. Caputo. Recognizing human actions: a local SVM approach. ICPR(17), pp 32-36, 2004.

[8] I. Laptev and T. Lindeberg. Space-time interest points. ICCV, pages 432439, 2003.

[9] Bobick, A. and Davis, J. 2001. The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 3 (Mar), 257-267.

[10] Rodriguez, M. D., Ahmed, J., and Shah, M. 2008. Action MACH: A spatio-temporal maximum average correlation height filter for action recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[11] Sheikh, Y., Sheikh, M., and Shah, M. 2005. Exploring the space of a human action. In IEEE International Conference on Computer Vision (ICCV). Vol. 1. 144-149.

[12] Yilmaz, A. and Shah, M. 2005a. Actions sketch: a novel action representation. In IEEE Con-ference on Computer Vision and Pattern Recognition (CVPR). Vol. 1. 984-989.

[13] Xi Zhou, et. al. SIFT-Bag Kernel for Video Event Analysis, In ACM Multimedia 2008.

[14] U. Guar, Y. Zhu, B. Song, and A. K. Roy-Chowdhury. A string of feature graphs model for recognition of complex activities in natural videos. In IEEE Intl. Conf. on Computer Vision, 2011.

[15] Z. Zivkovic. Improved adaptive gausian mixture model for background subtraction. In International Conference of Pattern Recognition, 2004.

[16] S. Ayache and G. Quenot, Video Corpus Annotation using Active Learning, In 30th European Conference on Information Retrieval (ECIR'08), Glasgow, Scotland, 2008.

[17] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2012 an overview of the goals, tasks, data, evaluation mechanisms and metrics. In Proceedings of TRECVID 2012. NIST, USA, 2012