UCF-CRCV at TRECVID 2013: Semantic Indexing

Afshin Dehghan[1], Guang Shu[1], Nasim Souli[1], Wenhui Li[1], Selen Pehlivan[1], Mubarak Shah[1], Jingen Liu[2], and Hui Cheng[2]

[1] [1] Center for Research in Computer Vision, University of Central Florida
[2] [2] SRI International

Abstract. This paper aims to report the system we used in semantic indexing (SIN) at TRECVID 2013. We participated in all three defined tasks this year, including main semantic indexing, localization and paired task. For the main task our approach uses a 4-stages processing pipeline. Feature extraction, fusion, classification and re-ranking are main stages of our system. For localization tasks we employed Deformable Part-based model followed by a saliency detection method to refine the initial detection candidates and finally for the paired task we use a simple fusion technique to select the right score for the concepts among the keyframes of a video-shot.

1 Introduction

Semantic Indexing is used as an approach for content-based video retrieval. The overview of tasks are described in [1, 2] but we can summarize the task into three groups:

- Main:'Given the test collection, master shot reference, and single concept definitions, return for each target concept a list of at most 2000 shot IDs from the test collection ranked according to their likelihood of containing the target'.
- Localization:'For each concept from the list of 10 designated for localization, for each shot of the top-ranked 1000 returned in a main task, localize the target in keyframe'.
- Paired: 'Given the test collection, return for each target concept pair a list of at most 2000 shot IDs from the test collection ranked according to their likelihood of containing the targe'.

The rest of the paper is organized as follows: Section 2 reviews the pipeline which was used for main task. Section 3 describes the process of localizing objects in keyframes. And in Section 4, we describe how we found the paired concepts in a video shot.

2 Semantic indexing

For the main task we submitted different runs both for *annotation* (training type A) and *no annotation* (training type E) tasks. For *annotation*, training samples for each concept are available [3]. However, for *no annotation* task we had to collect our own training set automatically.Therefore, we used web search engines including Google and Bing. The images collected in this way are noisy. In order to address this issue we used two constraints, entropy and size. The images with entropy below a threshold are discarded. Moreover, the images with size larger than a threshold are also removed in order to avoid keeping images with high resolution in our training set. This is important since the images used in testing are all of low-resolution. For *no annotation* task, we ended up keeping 500 images per concept in our training set.

There is significant difference between training sets of two tasks, figure 1 shows some of the images collected for concept *hand* and *airplane*. As can be seen the images collected online have less background noise while the images in the *annotation* task are more noisy and at the same time more similar to the images in test set. This shows that, even though we are able to collect enough training samples automatically, we expect lower performance because of the different between the type of images used in training and testing.

Fig. 1. The figure on the left shows some of the training samples used for concepts hand and airplane and the one the right shows samples for the same concepts but collected from web search engines for no-annotation task.

For semantic indexing task our approach follows a four-stage processing pipeline. We use state of the art low-level feature for representing our training images. Later these features are fused using standard early fusion technique and SVM is used as classifier to train the detector. The last step in our pipeline is re-ranking which is based on semantic similarity of video tags and concept names.

2.1 Feature Extraction

In our SIN 2013 system, we extracted three low-level static features from all key-frames.

- Dense SIFT and Color SIFT: SIFT [4] is a local feature which is widely used in object detection and is invariant to changes in scale and illumination. In our system, we extracted 128-dimension SIFT and 384-dimension color SIFT [5] dense feature descriptors. We employed the bag of visual words approach for both features and the visual vocabulary size was set to 500.
- ISA feature: ISA features have recently shown promising results in action recognition [6]. We extracted 500 dimension static ISA feature in a dense sampling manner and employed the bag of visual words approach on it. The codebook size is also 500. Our ISA features are extracted from all three RGB channels so they are encoded with the color cue. The ISA filters are trained with Trecvid IACC dataset to maximize the performance.
- GIST feature: In addition to local features like SIFT and ISA, we extracted 512-dimension GIST[7], since it is another important cue for scene understanding. GIST features help us in detecting some scene based concepts such as beach and nighttime and we expect it to be complementary to the local features.

2.2 Fusion and Classification

After extraction of the these low-level features, we normalized them and concatenate them to form a long features vector. We use Support Vector Machine (SVM) to build concept classifiers. Chi-square is selected as our kernel since it has shown better performance compared to other kernels in our experiments.

2.3 Re-ranking

Each video comes with a tag which in most cases carries sematic information about the video. We use these information for re-ranking. Given two pairs of strings (video-tag and concept name), we remove all punctuation, tokenize them, and remove redundant and stop words. Then, we build similarity matrix using Flickr text and tag search between the remaining words from two strings. We use the collective and individual counts of the two words (first word from string 1 and second word is from string 2).The similarity between individual words is computed using equation 1:

$$SemSim(w_i, w_j) = sigmoid\left[log_2\left(\frac{hits(w_i + w_j) * websize}{hits(w_i) * hits(w_j)}\right)\right] \quad (1)$$

where w_i and w_j are the two words in which we want to find the similarity between them. $hits(w_i)$ and $hits(w_j)$ show the number of times that the query word appeared in the web text and $websize$ is a constant. Once a similarity matrix between the words is constructed, then the goal is to compute a single similarity score between complete strings (from the similarity score between words). For that, we take mean of matched words after bipartite matching on the similarity matrix. The best performance out of all the submitted runs was the one which we applied our re-ranking approach to.

2.4 Evaluation

For internal evaluation purpose we created a small evaluation dataset from IACC development dataset to select features and optimize parameters. 10 concepts out of 60 concepts are selected for our evaluation dataset. We divided the subset into two parts, $\frac{2}{3}$ of the images from each concept are used for training and $\frac{1}{3}$ are used for testing. The same subset is collected for the *no annotation* task from images retrieved from web. The details of our validation dataset is shown in 2. Each row shows a single concept and each column shows the number of examples for each division.

Name	**E10_train**	**A10_train**		**10_test**	
	#examples	#positives	#negatives	#positives	#negatives
Sum up	7,503	9,952	21,492	4,866	10,145
0003_Airplane	757	749	1745	270	393
0015_Boat_Ship	849	884	2074	308	424
0017_Bridges	805	536	1163	248	481
0019_Bus	794	178	809	69	189
0025_Chair	636	1506	2979	776	1810
0059_Hand	666	2340	5200	1302	2460
0080_Motorcycle	867	384	868	140	231
0117_Telephones	805	353	617	98	382
0261_Flags	635	884	1133	709	2208
0392_Quadruped	689	2138	4904	946	1567

Fig. 2. Our evaluation dataset. E10 contains examples of 10 concepts from google images, A10 contains examples of 10 concepts from Trecvid IACC dataset.

We compared the results of different features as well as their fusion. We use Average Precision (AP) [8] as the evaluation metric. AP summarizes the characteristic of precision/recall curve, and is defined as the mean precision at a set of equally spaced recall levels. 3 shows the AP score for each concept for different feature and fusion of those features. The third row shows the mean average precision of all the 10 concepts.

SIFT has the best overall performance among all single features. However, for some concepts such as *Flags* and *Quadruped*, colorSIFT and ISA features which have color information outperform SIFT. This observation indicates that including color information could improve the performances for some concepts that have more color information. The last column shows the result of early fusion, and it outperforms all the single features for all concepts. Considering the computational cost we finally choose early fusion of SIFT, ISA and GIST as the final feature representation in our system.

Features	SIFT		CSIFT		ISA		GIST		SIFT+ISA+GIST	
Training Types*	E	A	E	A	E	A	E	A	E	A
mAP	**44.24**	**54.21**	**44.41**	**53.11**	**42.00**	**53.47**	**41.87**	**47.62**	**45.23**	**62.64**
0003_Airplane	66.26	80.28	67.53	79.13	56.68	76.5	73.85	74.49	73.03	84.84
0015_Boat_Ship	60.09	60.36	56.96	60.93	60.58	68.98	56.15	55.37	61.67	72.25
0017_Bridges	63.49	75.26	60.34	75.48	35.51	64.86	41.68	69.14	59.85	77.95
0019_Bus	43.29	41.76	39.62	37.43	44.85	32.41	53.17	40.7	54.03	54.81
0025_Chair	33.63	45.82	33.08	43.51	30.11	50.05	33.83	42.43	27.39	56.16
0059_Hand	35.66	42.41	36.64	42.88	40.97	44.02	35.47	42.1	36.71	50.12
0080_Motorcycle	40.58	63.33	40.12	57.31	52.07	67.98	31.54	47.71	39.74	76.85
0117_Telephones	23.39	31.33	25.81	27.05	25.55	34.49	27.1	24.02	24.11	37.91
0261_Flags	31.67	49.66	34.61	51.31	25.83	39.15	26.22	31.97	27.69	55.37
0392_Quadruped	44.37	51.85	49.39	56.02	47.63	56.22	39.67	48.26	48.09	60.11

* E-- google images, A—trecvid IACC data

Fig. 3. Average Precision results on our evaluation dataset. The red color show the highest scores.

3 Object Localization

Object localization (detection) is the task of localizing objects in an image, and it has many applications in computer vision and video understanding area such as monitoring and navigation. However it is generally a difficult task due to challenges, like changes in illumination, pose variation and occlusion.

Part based models are widely used for object localization. Deformable Part Model [9] have shown decent results for object detection task; however, the performance of this model depends on training data and how good the model parts are initialized. Moreover, DPM model only considers HOG features and it does not take into account other appearance features such as color. Also, coherent specialty between pixels could be a useful cue in order to localize the object which is also ignored in DPM.

To address the mentioned issues we collected our own training data which is more similar to the ones in test set. These annotations are used to train our model. We also used a saliency detection approach to refine detection candidates. Saliency map indicates the saliency of a specific location over the entire scene. In some studies saliency detection on complex scenes have been formulated using the Bayesian method. SUN model [10] attempts to detect saliency by estimating the probability of presenting a target given visual features at every location in the scene using the maximum information approach. In [10], the first step of processing the input image is to apply different filters including Difference of Gaussians filters and Linear ICA filters on images, then by using Bayesian

framework the probability of a pixel being salient is found. In our system we use saliency map to remove the false positives as we expect them to be less salient.

Once we get the final detection output from fusing the saliency and detection map, we use superpixel segmentation [11] followed by a Conditional Random Field to further refine the detection output and get a more accurate object boundary. In our CRF model, the unary potential of each superpixel is found by summing up the normalized DPM score and saliency value for that. The pairwise edge potentials are also defined by finding the similarity of neighboring centers using the inverse of Eculidian distance in RGB features space.

The steps of our algorithm can be summarized as follows : first by using the DPM model we find our initial candidates for a specific category. Later the false positives are removed by fusing the score of saliency map and detection map. Finally we find the object boundaries using the method mentioned above.

In figure 4 some results are shown.

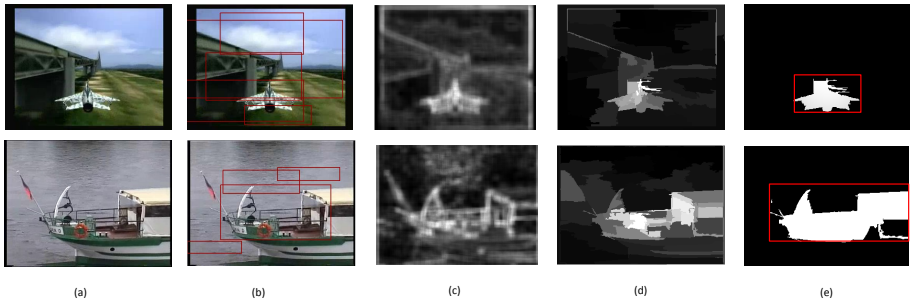

Fig. 4. Some results using our method : (a) shot images, (b) bounding boxes obtained by DPM, (c) saliency maps, (d) super-pixels with probability scores assigned to and (e) final results after CRF.

4 Semantic Indexing Pair-Concept Task

SIN pair-concept task seeks for the co-occurrences of unrelated concepts in video shots. For every concept pair, the system checks whether the shot frames includes both concepts simultaneously. Our method is a simple fusion technique using the outputs of main task detectors that are trained for individual concepts. For a given video shot, every concept detector returns the concept hypotheses with a score per key frame. This score is the probability output of the concept SVM detector. Particularly, having N individual concept detectors, we compote N scores per frame. Each score s_n^f is the probability of having the concept n in

frame f. For each concept, our fusion method finds the maximum value of all scores of that concept as the best score for a shot. Computing the maximum score of every concept in the shot, method gets the pairwise products of concept scores

$$prod(\max(s_n^f), \max(s_m^f)) \tag{2}$$

5 Acknowledgment

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract numbers D11PC20066. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation 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. : (http://www-nlpir.nist.gov/projects/tv2013/tv13.call.html)
2. Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Kraaij, W., Smeaton, A.F., Quenot, G.: Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In: Proceedings of TRECVID 2013, NIST, USA (2013)
3. Ayache, S., Quenot, G.: Video Corpus Annotation using Active Learning. In: European Conference on Information Retrieval (ECIR), Glasgow, Scotland (2008) 187–198
4. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. (2004)
5. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 1582–1596
6. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR. (2011)
7. Oliva, A., Torralba, A., Guerin-Dugue, A., Herault, J.: Global semantic classification of scenes using power spectrum templates. In: Challenge of image retrieval. (1999)
8. Benfold, B., Reid, I.: Stable multi-target tracking in real time surveillance video. In: CVPR. (2011)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32** (2010) 1627–1645
10. Zhang, L., Tong, M.H., Marks, T.K., Shan, H., Cottrell, G.W.: Sun: A bayesian framework for saliency using natural statistics. Journal of Vision **8** (2008)
11. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 2097–2104