UCF-CRCV at TRECVID 2014: Semantic Indexing

Afshin Dehghan[1], Mahdi M. Kalayeh[1], Yang Zhang[1], Haroon Idrees[1], Yicong Tian[1], Amir Mazaheri[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 the main task of semantic indexing (SIN) at TRECVID 2014. Our system uses a five-stage processing pipeline including feature-extraction, feature-pooling, feature-encoding, classification and fusion. We employed CNN-based representation as well as other widely used hand crafted features at feature extraction level. We report the results of all four submitted runs as well as the improved version which we evaluated after the judgment file was released. Our improved system achieves the infAP of 25.41% for the 30 evaluated concepts.

1 Introduction

Semantic Indexing is used as an approach for content-based video retrieval. The main task in Semantic Indexing is defined as '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' [1]. Based on the training data used in the system, each method can be divided into one of the following types:

- Type A:'used only IACC training data'.
- Type B:'used only non-IACC training data'.
- Type C:'used both IACC and non-IACC TRECVID (S and V and/or Broadcast news) training data '.
- Type D:'used both IACC and non-IACC non-TRECVID training data '.

In our training we used both IACC and non-TRECVID data. Thus all of our runs are of type D. The rest of the paper is organized as follows: Section 2 reviews the pipeline which was used for the main task. Section 3 describes the features and descriptors used in our pipeline and finally in Section 4 we show quantitative performance of all four submissions as well as the improved version.

2 System Overview

We used training images available to all the participants [2] as well as images collected from web search engines including Google and Bing. We collected additional training samples ($100-200$ images) for 5 concepts, each with less than 200

training images. The overview of our system is shown in Figure 1. Each image is divided into different spatial regions. Features are extracted and encoded for each region and a classifier is learned for each feature individually. The decision values are fused at the end to obtain the final detection score.

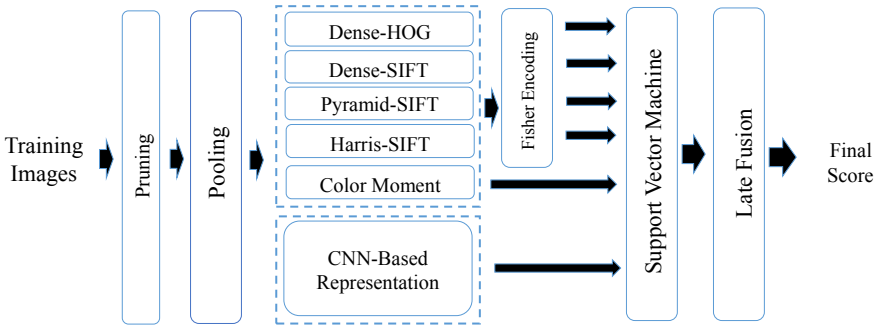

Fig. 1. This figure shows the pipeline used in our approach. Images are pruned and divided into spatial regions. Different features are extracted for each region and a SVM is trained for each feature individually. For a given test video, the samples are fused using our linear programing approach.

3 Features and Descriptors

In our SIN 2014 system, we extracted six different features. Five out of six are widely used features and one of them is based on recent generation of Convolutional Neural Networks (CNN) which plays a major role in the performance of our system.

- CNN-based feature: In order to extract CNN features we used the network proposed in [3]. The network is trained on ImageNet training images [4]. We used the output of the last fully connected layer which is a 4096 dimensional vector to represent each region in the image. We used a total of 8 regions (full image, four grids and three horizontal strips). The final feature is the concatenation of all 8 regions which makes a 32768 dimensional feature vector.
- Dense SIFT and Harris SIFT: We extracted SIFT descriptors with two different sampling strategies. In the first one, we densely sampled every 6 pixels at the scale of 1.2. In the second one, we used Harris corner detector to find the interest points. Then SIFT descriptors were computed at each interest point with scale of 1.2. We encoded these descriptors using Fisher vector[5]. We used PCA to reduce the dimensionality of SIFT descriptors to 80 to decorrelate them. We randomly selected about 1 million low-level descriptors from training data and fit a GMM with 128 components. This GMM was later

used for aggregating low-level descriptors through Fisher vector framework. Power and L^2 normalizations were applied to compute the Fisher vectors. Our final representation was a 163840 dimensional vector per image as we used spatial pyramid (1×1, 2×2 and 3×1).

- P-SIFT: We extracted the P-SIFT according to the method proposed in [6]. We densely sampled P-SIFT every 6 pixel across the entire image. Same as SIFT we used PCA to reduce the feature dimension to 80 and then applied Fisher vector to encode the features with 128 GMM components. Due to the eight pooled regions, the dimension of final Fisher vector representation for an image is 163840.
- Dense HOG: Similar to the Dense-SIFT and P-SIFT, we extracted HOG features densely from the entire image. The features are sampled every 8 pixels. At each location HOG features are extracted for a patch of size 16×16. The cell size is set to 8 in our setup. Similar to SIFT, H-SIFT and P-SIFT we used Fisher vector for feature encoding and the final representation is a 163849 dimensional vector.
- Color Moments: For each image, we extracted 3 moments of image color distribution, The moments are mean, standard deviation and skewness. Each color channel has 3 moments, which makes the final representation for each region a 9 dimensional feature vector. The dimension of final color moment from the 8 pooled regions is 72.

4 Fusion and Classification

After extraction of the these low-level features, each feature is normalized and used to train a Support Vector Machine (SVM). The score of each detector is later combined in a late fusion fashion. Our fusion is based on linear programming. Given classification scores on validation data, our goal is to obtain weights to combine feature scores such that the precision for binary classification is maximized on the validation set. Then, the same weights are applied on test data to perform fusion. The following equation gives the objective function of the optimization which is based on LPBoost [7] .

$$\min_{\beta,\xi} -\rho + \frac{1}{\nu N} \sum_{i=1}^{N} \xi_i. \tag{1}$$

Subject to:

$$y_i \sum_{m=1}^{F} \beta_m f_m(x_i) + \xi_i \geq \rho, \qquad i = 1, \cdots, N, \tag{2}$$

$$\sum_{m=1}^{F} \beta_m = 1, \qquad \beta_m \geq 0, \qquad m = 1, \cdots, F \tag{3}$$

where N is the number of samples in the validation set. y_i are the labels $\{-1, +1\}$, $f_m(x_i)$ is the score of m^{th} classifier on the i^{th} sample and F denotes

the total number of features. ξ_i are the slack variables (greater than zero) and β_m are the weights for the m^{th} feature. Our objective is to obtain a set of β_m and ξ_i such that slack variables are minimized while satisfying the margin set by ρ and maximizing the prediction accuracy on validation set. The βs are further constrained to be non-negative with unit norm in Equation 3. The approach is extremely fast and takes only few seconds to find the optimal weights in the validation set consisting of $10,000$ video shots.

5 Evaluation

In this section we will review the performance of all four submitted runs as well as the improved version.

- Run1-Donatello: For our first run we used all the features except CNN. For classification we used SVM with linear kernel and all the scores were fused through the proposed fusion method. The final score for a shot is found by taking the maximum concept scores across key-frames of that shot (max-pooling).
- Run2-Leonardo: Our second run is similar to the first run with the difference that CNN features are added the pool of features.
- Run3-Leonardo: For this run we only used our CNN features followed by a linear SVM and max-pooling across key-frames of a shot.
- Run4-Leonardo: This run is similar to the second run. The only difference is that the final score for a shot is obtained by taking the average key-frame scores of a shot.
- Improved-run: In our improved run we used only two features: DSIFT and CNN-based features. The kernel used for the CNN-based features is changed from 'linear' to 'histogram intersection'. For DSIFT features we changed the feature extraction parameters (step size was reduced and more scales were used).

In Fig. 2 the results of all four submitted runs are shown. *Run3-Michelangelo* has the best performance among all the runs. The best results is for concepts *Beach*, *News Studio* and *Ocean*.

Using the released judgment files for IACC.2.B test data we were able to compute the infAP for our improved run. In Fig. 3 we show the performance of the two improved features in our system, DSIFT and CNN-based features, as well as their fusion. The infAP for only the CNN-based features is 19.98% while for DSIFT it is 19.04%. Combining these features boosts the performance to 25.41%. This shows that these two features are complementary to each other and fusion can boost the performance by significant margin. For *Beach*, *Instrumental Music*, *Ocean*, *News and Studio* and *George Bush*, the infAP is close/more than 50%. The lowest performing concepts are: *Bus*, *Basketbal*, *Hand* and *Telephone*. One of the reasons for that is the number of training samples (especially for *Basketball* and *Bus*) is much fewer compare to other concepts.

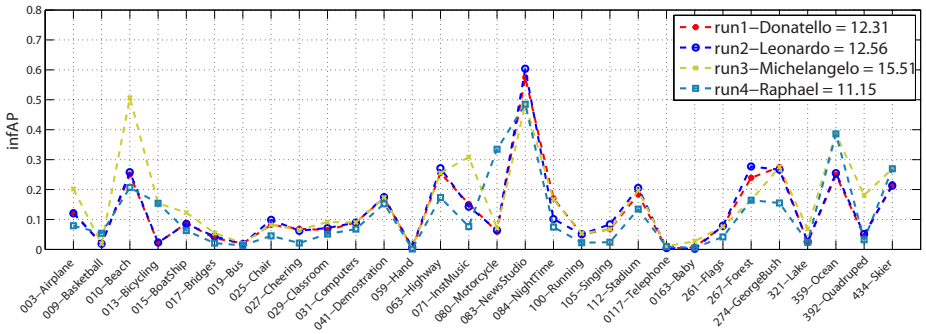

Fig. 2. The average infAP of all four submitted runs: This year only 30 concepts out of 60 concepts were used for evaluation. Run3-Michelangelo has the best performance (infAP=15.51%).

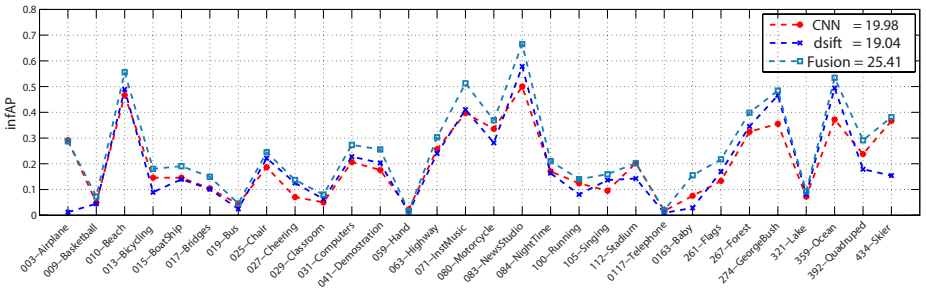

Fig. 3. The infAP of our improved system for DSIFT, CNN-based features and the fusion of all features.

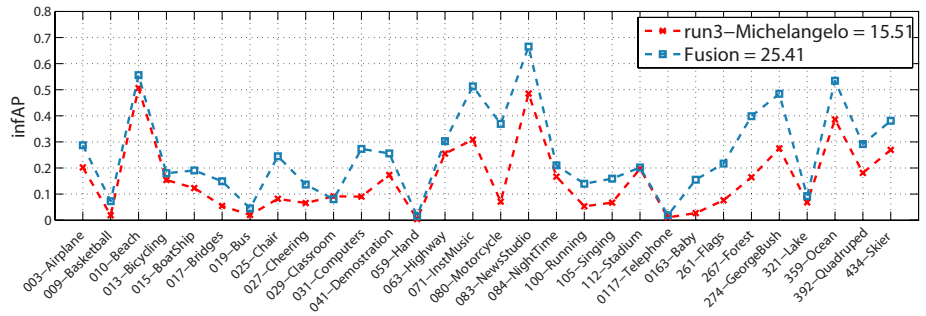

Fig. 4. The comparison of our best submitted run with the improved run. We were able to improve our system by $\sim 10\%$

In Fig. 4 we compare our best submitted run with the improved system. As can be seen the performance is increased by almost 10%.

Fig. 5 shows the best and average infAP reported among all the submission to TRECVID SIN 2014 for each concept. Almost for all the concepts , the infAP obtained by our system is significantly higher than average which proves the effectiveness of the features used in our system. The most difficult concepts to detect are *Basketball*, *Bus*, *Hand*, *Telephone* and *Lake*.

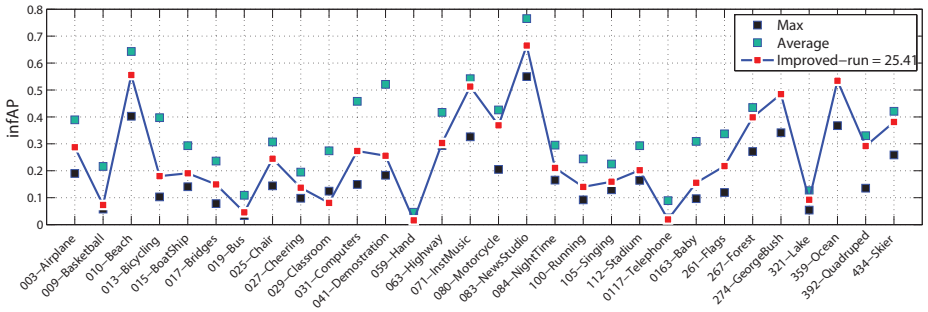

Fig. 5. This figure shows comparison of our method with mean and maximum infAP reported for each concept.

Finally in Fig. 6 we show the ranking of our system compared to other participants before and after submission (only the top 54 out of 74 submitted runs are shown). With the infAP=25.41% we are ranked 6^{th} among all the teams participated in TRECVID SIN 2014.

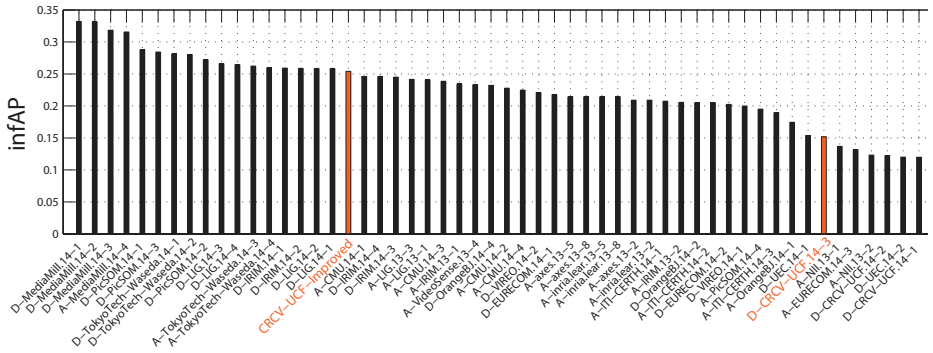

Fig. 6. Top 54 submitted runs to TRECVID SIN 2014. Our runs are the ones shown in orange.

Acknowledgment

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number 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. Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Kraaij, W., Smeaton, A.F., Quenot, G.: Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In: Proceedings of TRECVID 2014, NIST, USA (2014)
2. Ayache, S., Qunot, G.: Video corpus annotation using active learning. In: 30th European Conference on Information Retrieval (ECIR'08). (2008)
3. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., Weinberger, K., eds.: Advances in Neural Information Processing Systems 25. Curran Associates, Inc. (2012) 1097–1105
4. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge (2014)
5. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Computer Vision–ECCV 2010. Springer (2010) 143–156
6. Seidenari, L., Serra, G., Badanov, A.D., Bimbo, A.D.: Local pyramidal descriptors for image recognition. In: PAMI. (2013)
7. Demiriz, A., Bennett, K.P., Shawe-Taylor, J.: Linear programming boosting via column generation. In: JMLR. (2002)