EURECOM at TrecVid 2013:
The Semantic Indexing Task

Usman Niaz, Miriam Redi, Claudiu Tanase, Bernard Merialdo
Multimedia Department, EURECOM
Sophia Antipolis, France
{Usman.Niaz,Miriam.Redi,Claudiu.Tanase,Bernard.Merialdo}@eurecom.fr

February 10, 2014

1 Abstract

This year EURECOM participated in the TRECVID 2013 Semantic INdexing (SIN) Task [11] for the submission of four different runs for 60 concepts. Our submission builds on the runs submitted last year at the 2012 SIN task, the details of which can be found in [8]. In 2013, two runs are combinations of basic descriptors. One run adds uploaders bias to the pool of visual features while another run was prepared in collaboration with Aalto University. All runs are trained on annotations provided by the IRIM collaborative effort [4].

When compared with last year system, our runs use a larger set of visual features, and larger visual dictionaries to provide a finer representation of the visual/clustering space and increase the precision of the retrieval system. Like in last year's submission we add a global descriptor to visual features capturing salient details or gist of a keyframe. We further benefit from metadata information by including an uploader bias to increase the scores of videos uploaded by same users. A major difference this year is that we have used a new training algorithm based on a combination of PEGASOS [14] mixed with Homogeneous Kernel Maps [16], while our previous system was based on the libsvm library [5].

Our four runs are organized as follows:

1. **EURECOM_C**: This is our basic run which fuses a pool of visual features, namely SIFT [7] descriptors extracted through dense, log and hessian methods, a Saliency Moments feature [12], a Color Moments global descriptor, the Wavelet Feature, the Edge Histogram, the texture based Local Binary Pattern feature and the dense color pyramid [15]. The order of the Homogeneous Kernel Maps varies from 1 to 11, depending on the size of each descriptor. The classifiers are linear SVM trained with the PEGASOS algorithm. A group-based fusion combines the results of the classifiers to provide the final score.

2. **EURECOM_EC**: This run uses the same descriptors as the previous one, but the classifiers use a quantized version of the feature vectors. This allows to remove some memory

constraints for large vectors, and allows to use a higher order for the Homogeneous Kernel Maps.

3. **EURECOM_ECU**: This run uses uploaders' information to boost the scores of the previous runs for certain concepts and certain users.

4. **EURECOM-PicSOM**: This run combines EURECOM_ECU with a run from Aalto University.

Beside this participation, EURECOM took part in the collaborative IRIM and VideoSense submissions; the details of those systems are included in the respective papers.

The remainder of this paper briefly describes the content of each run (Sec 2-5), including feature extraction, classifier training and fusion methods. Figure 1 gives an overview of the relationship between the 4 runs. In Section 6 results are commented and discussed.

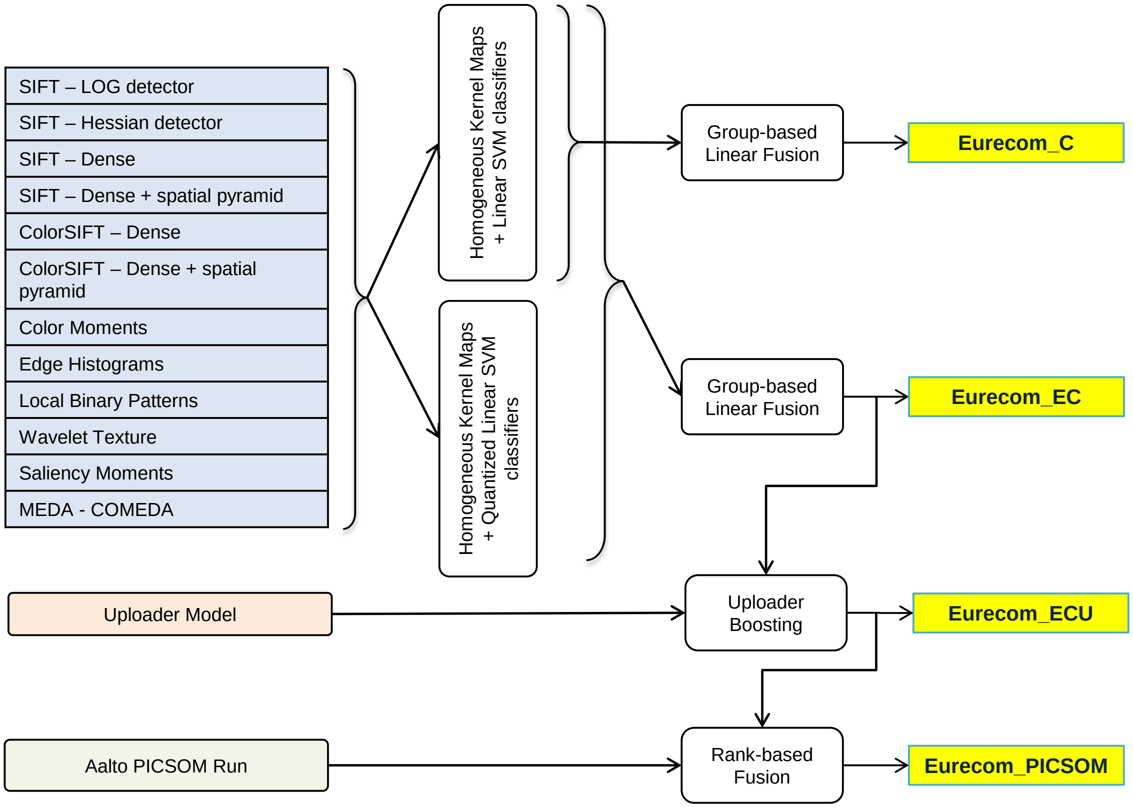

Figure 1: Framework of our system for the semantic indexing task

2 EURECOM Basic Run: EURECOM_C

This run comprises a number of visual features ranging from local features to global image descriptions. In this stage, the following features are computed.

- **Color Moments** This global descriptor computes, for each color channel in the LAB space, the first, second and third moment statistics on 25 non overlapping local windows per image.

- **Wavelet Feature** This texture-based descriptor calculates the variance in the Haar wavelet sub-bands for each window resulting from a 3 × 3 division of a given keyframe.

- **Edge Histogram** The MPEG-7 edge histogram describes the edges' spatial distribution for 16 sub-regions in the image.

- **Local Binary Pattern (LBP)** Local binary pattern describes the local texture information around each point [9], which has been proven effective in object recognition. We employ the implementation in [2] to extract and combine the LBP features with three different radius (1, 2, and 3) and get a 54-bin feature vector.

- **SIFT from keypoints** Two sets of interest points are identified using different detectors:

  1. Difference of Gaussian
  2. Hessian-Laplacian Detector

  For each of the detected keypoints we then compute a SIFT [7] descriptor using the VIREO system [3]. We use the K-means algorithm to cluster the descriptors from the training set into 500, 1000 and 2000 visual words. After quantization of the feature space, an image is represented by a histogram where the bins of this histogram count the visual words closest to image keypoints. We therefore obtain feature vectors of dimension 500, 1000 and 2000.

- **Dense SIFT and ColorSIFT** We also use a dense sampling for the SIFT and ColorSIFT descriptors proposed by Koen Van de Sande [15]. We use their software provided in [1]. We created visual dictionaries of size 1000, 4000 and 10000. We pool the quantized descriptors globally over the whole image. We also consider pooling according to a spatial pyramid (1, 2x2, 3x1), so that the corresponding feature vectors have a dimension 8 times the size of the dictionary.

- **Saliency Moments descriptor** This is a holistic descriptor which embeds some locally-parsed information, namely the shape of the salient region, in a holistic representation of the scene, structurally similar to [10]. First, the saliency information is extracted at different resolutions using a spectral, light-weight algorithm [6]. The signals obtained are then sampled directly in the frequency domain, using a set of Gabor wavelets. Each of these samples, called "Saliency Components", is then interpreted as a probability distribution: the components are divided into subwindows and the first three moments are extracted, namely mean, standard deviation and skewness. The resulting signature vector is a 482-dimensional descriptor [12].

- **MEDA - COMEDA** We have proposed descriptors based on marginal distributions of the local descriptors. They have the advantage of a faster computation than bag-of-word construction, and have shown efficient performance. Those descriptors are described in [13].

Those feature vectors are expanded using the Homogeneous Kernel Maps [16], which approximate the Histogram Intersection kernel by a scalar product. This allows to use linear SVMs classifiers on the expanded vectors, instead of SVMs with more complex kernels. The order of the Homogeneous Kernel Maps is adapted to the dimension of the feature vectors and range from 1 to 11. For the longer feature vectors we use a lower order, so that the total memory usage remains reasonable. An order of 1 corresponds to no change to the feature vector, so that the classifier is a linear SVM on the initial feature vector. For higher orders, the classifier is an approximation of an SVM with Histogram Intersection Kernel of the initial feature vector. The higher the order, the better the approximation.

To train the linear SVMs, we have implemented a variation of the PEGASOS algorithm [14], where we dynamically adapt the weight between positive and negative samples, as well as updating several models in parallel. For each concept, we select the best SVM parameters as those which maximize the Mean Average Precision (MAP) on a validation set. Because of time constraints in the training phase, not all classifiers are available for all features.

We fuse the results of the classifiers using a two-level scheme. First, we gather the classifiers into groups, where the groups contain similar features, for example a descriptor with different sizes of dictionary. We first fuse the descriptors in each group, then fuse the results of all groups altogether. The fusion is a linear fusion where the weights are optimized on a validation set. To build the submission run, we apply all the classifiers to the test data, then perform the same fusion mechanism to obtain the final scores.

3 EURECOM Second Run: EURECOM_EC

This run is an expansion of the previous one, where we add new classifiers which operate on a quantized version of the feature vectors. The quantization reduces the memory requirements, so that it is possible to use a higher order of the Homogeneous Kernel Maps, even for the largest feature vectors. The fusion mechanism remains the same, with new groups being added for the quantized classifiers.

4 EURECOM Third Run: EURECOM_ECU

As introduced last year, we also exploit the metadata provided with the TRECVID videos to benefit from the video uploader's information. The TRECVID training and test data come from the same distribution and we have found that videos from several uploaders are distributed evenly over the corpus. We benefit from this information based on the assumption that an uploader is likely to upload videos with similar content. In other words most if not all videos uploaded by one user represent information that is not much different from one another. For example if a user runs a video blog about monuments in a certain city then almost all videos uploaded by that user will contain concepts like *sky* or *outdoor*. This information thus increases our confidence in the predictions of the concepts *sky* and *outdoor* if the test video is uploaded by the same user. This model is applied to selected concepts on top of Run 2.

The uploader model simply calculates the ratio of video shots uploaded by the uploader for each concept from the training data and modifies the output score of each new video shot if that video is uploaded by the same person. This uploader bias allows us to rerank the retrieval results. For each concept we calculate the probability of concept given uploader as:

$$p(c/u) = \frac{W_u^c}{|V_u|}$$

where V_u is the set of videos uploaded by uploader 'u' and W_u^c is the weightage of videos uploaded by uploader 'u' for the concept 'c'. This quantity:

$$W_u^c = \sum_{v \in V_u} \frac{|s \in v, s.t.s = c|}{|s \in v|}$$

is the sum of ratios of the number of shots labeled with concept 'c' to the total shots in that video for all the videos uploaded by 'u'.

We also calculate average uploader's probability for each concept as:

$$p(c) = \frac{W^c}{|V|}$$

where W^c is the total weightage of all the videos uploaded for concept 'c', given by:

$$W^c = \sum_u W_u^c$$

and V is the number of videos, or $V = \sum_u V_u$.

This model is computed on the training data separately for each concept. To apply the uploader's model to the test videos we calculate the coefficient α as:

$$\alpha(c,u) = \max\left(\frac{p(c/u) - p(c)}{p(c/u) + p(c)}, 0\right)$$

The score of each shot $P(c|s)$ from the previous run is modified in the following way:

$$p_u(c|s) = p_2(c|s) * (1 + \alpha(c,u))$$

Cross validation on the development set showed that the uploader model has a positive impact for 21 out of 60 concepts. It was therefore applied to those 21 concepts for the final run and the other 39 were left unchanged.

5 EURECOM Fourth Run: EURECOM-PicSOM

This run is a collaboration between Eurecom and Aalto University. Aalto provided the scores of one of their runs for all concepts. We used these to combine with the scores from our EURECOM_ECU run. Because of time constraint, we were not able to share information on the training data, therefore we had no material to optimize a potential fusion mechanism. Moreover, as the scores were obtained through completely different processing chains, their numerical values (range and distribution) were not comparable. We decided to rely on a rank-based fusion: for each run, we constructed an artificial score from the rank of each shot, equal to the inverse rank (adjusted by an offset). Then, we simply averaged those scores and rerank the shots.

$$score(s) = \frac{1}{2} * \frac{1}{offset + rank1(s)} + \frac{1}{2} * \frac{1}{offset + rank2(s)}$$

The value of the offset was taken as 100.

6 Results Analysis

Run	MAP
EURECOM-PicSOM	0.2000
EURECOM_ECU	0.1647
EURECOM_EC	0.1289
EURECOM_C	0.1171

Figure 2: Evaluation results for our (corrected) runs

Unfortunately, a bug in the last stage of the preparation of the runs causes the results lists between different concepts to be mixed together. So, although the scores for each concept were correctly computed, the sorted lists were completely erroneous, which caused the evaluation of our runs on the test data to give irrelevant scores.

In Figure 2, we indicate the performances (MAP) of our runs, as described in the previous chapter, but after correction of this bug. We can see that the uploader model had a subtantial impact over the our second run, even though it was only applied over about one third of the concepts. The performance of the initial PICSOM run was 0.18, so we can also observe that mixing our run with the Aalto run was greatly beneficial.

In figure 3 we display the comparative performance of the four runs on each of the concepts evaluated by NIST.

The difference in performance between Run 1 and Run 2 is due to the use of a larger order for the Homogeneous Kernel Maps.

Run 3 improves further on Run 2. The uploader model improves average precision for every concept, especially for concepts *Dancing*, *Government_Leader*, *Motorcycle*, *George_Bush*, *Quadruped* and *Studio_With_Anchorperson.*

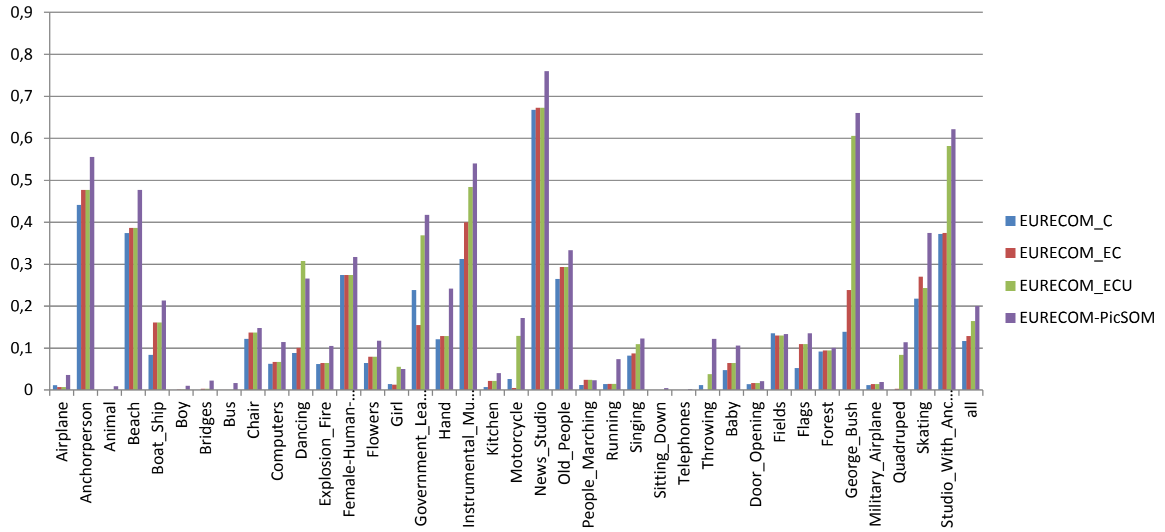

Figure 3: Results on the test set evaluated by NIST

7 Conclusions

This year EURECOM presented a set of systems for the Semantic Indexing Task. As last year we confirmed that using uploader information allows to greatly improve the detection of certain concepts. We also showed that combining results from two completely different systems, even if the combination has not been optimized on validation data, leads to a net improvement of the performance on test data.

References

[1] Color descriptors, http://koen.me/research/colordescriptors/.

[2] Local binary pattern, http://www.ee.oulu.fi/mvg/page/home.

[3] Vireo group in http://vireo.cs.cityu.edu.hk/links.html.

[4] S. Ayache and G. Quenot. Video Corpus Annotation using Active Learning. In *European Conference on Information Retrieval (ECIR)*, pages 187–198, Glasgow, Scotland, mar 2008.

[5] C. Chang and C. Lin. LIBSVM: a library for support vector machines. 2001.

[6] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.

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

[8] U. Niaz, M. Redi, C. Tanase, and B. Merialdo. EURECOM at TRECVID 2012: The light semantic indexing task. In *TRECVID 2012, 16th International Workshop on Video Retrieval Evaluation, 2012, National Institute of Standards and Technology, Gaithersburg, USA*, Gaithersburg, UNITED STATES, 11 2012.

[9] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971 –987, jul 2002.

[10] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[11] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[12] M. Redi and B. Mérialdo. Saliency moments for image categorization. In *ICMR'11, 1st ACM International Conference on Multimedia Retrieval, April 17-20, 2011, Trento, Italy*, 04 2011.

[13] M. Redi and B. Merialdo. Direct modeling of image keypoints distribution through copula-based image signatures. In *ICMR 2013, ACM International Conference on Multimedia Retrieval, April 16-19, Dallas, Texas, USA*, Dallas, ÉTATS-UNIS, 04 2013.

[14] 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*, ICML '07, pages 807–814, New York, NY, USA, 2007. ACM.

[15] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[16] 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.