UC3M AT TRECVID 2009

I. González-Díaz, V. Gómez-Verdejo, M. Martínez-Ramon, F. Díaz-de-María and J. Arenas-García

Department of Signal Theory and Communications
Universidad Carlos III de Madrid, Leganés, Spain

ABSTRACT

This paper describes the experiments carried out by the UC3M team for the TRECVID 2009 high-level feature extraction task. Last year, during our first particiation in TRECVID, we developed a modular system to facilitate the testing of several functionalities. This year we have focused on analyzing different configurations: an early/late fusion of some low level features, different classification technologies, validation parameters, the usefulness of a feature extraction stage, and two possible fusion alternatives. This analysis has provided a large set of combinations for the system set-up and, therefore, a large number of potential runs to be submitted to TRECVID; however, in the light of the results on a validation set, we have selected the following four system configurations:

- **RUN 1** (*"A UC3M 2 1"*): this is our baseline, which is characterized by including early fusion keyframe features, considering a set of classical SVMs validated with the AP parameter as classification learners, applying a feature selection process to remove the useless learner outputs, and employing a linear SVM to fuse the selected learner outputs.
- **RUN 2** (*"A UC3M 4 2"*): this run has the same configuration as RUN 1, but KOPLS technology is used as the classification technology instead of a classical SVM.
- **RUN 4** (*"A UC3M 15 4"*): this run also has the same configuration as the baseline, but now the F score as the validation criterion to adjust the SVM free parameters.
- **RUN 3** (*"A UC3M 35 3"*): this run employs the same configuration as RUN 4, but the fused output is generated as a linear combination carried out with a ranking SVM.

Additionally, the last two runs have been used to establish a comparison between discriminative and generative approaches of the well known bag-of-words model. In particular, these runs use only local features described by means of visual words:

- **RUN5** (*"A UC3M 50 5"*): this run tests the performance of a generative bag-of-words approach that models the spatial distribution of visual words along the visual documents.
- **RUN6** (*"A UC3M 51 6"*): this run is associated to a basic discriminative bag-of-words model that uses a SVM over the histograms of visual words.

The six submitted runs have achieved average InfAP values from 0.05 to 0.09 which places our designs in the second quartile of all TRECVID 2009 submitted runs.

1. INTRODUCTION

The main goal of our participation in the TRECVID high-level feature extraction (HFE) competition has been to analyze different configuration alternatives for the components of the modular system we presented last year. This system consists of four processing steps: (1) a low level feature extraction layer, (2) a supervised learning step, (3) a feature selection process to choose the most adequate learner outputs and (4) a final fusion stage. As we have explained, our efforts have mainly focused on analyzing different system configurations. In particular, we have considered the following elements for each step of the system:

- *An early/late fusion of low level features*: we have considered, on the one hand, using each low level feature as the input of an individual classifier and, on the other hand, merging some simple features into a unique one so that the relations between these can be exploited at this early fusion step.
- *Classification technologies*: we have analyzed whether employing other classification technologies different from classical SVM can provide or not significant performance improvements. For this purpose, we have used several classification techniques as base learners, namely: the classical SVM [1], a modified ranking SVM [2], and a KOPLS classifier [3].
- *Validation parameters*: in order to adjust the different free parameters of the above learners, we need an appropriate criterion; due to fact that the final TRECVID

This work is part of the i3media Project (CDTI 2007 1012) and is partially funded by the Centro para el Desarrollo Tecnológico Industrial (CDTI), within the Ingenio 2010 Program.

results are measured in term of InfAP, using the AP parameter as the validation criterion seems reasonable, we have decided to explore other alternatives such as the balanced classification error and the F score [4].

- *Feature selection*: after training all learners, we have included a feature selection step, so that we can remove the worst-performance learners and, consequently, fuse only the most appropriate learner outputs in the following system step.
- *Fusion stage*: two possibilities have been explored to carry out a linear combination of the (selected) learner outputs: a classical linear SVM and linear ranking SVM.

Since the analysis of these configurations has provided a great variety of runs, we have calculated the validation InfAP value at the final system output in order to select the ones that were submitted to TRECVID 2009. According to this idea, we have selected the four runs which showed the best performance, as well as those whose results allowed us to decide which system elements were more appropriate.

Additionally, we wanted to check the isolated performance of individual learners that use only local features. For this purpose, our submissions have also served to establish a comparison between generative and discriminative approaches to the bag-of-words model. It is worth noting that, as observed from the experiments, the bag-of-words model provides the best results when comparing outputs from the basic learners (those learners that take as input only one low level feature). Thus, we have laid special emphasis on the study of this classification technique.

The remainder of this paper is organized as follows: The next section presents the Low Level Features that have been extracted from the video data. In Section 3, we describe in detail the different configurations of our system and the preliminary performance analysis carried out to select the submitted runs. Next, experimental results are analyzed in Section 4. Finally, Section 5 summarizes the conclusions.

2. LOW LEVEL FEATURE EXTRACTION

2.1. Using Mutual Information to optimize low-level descriptors

Most of the audiovisual low-level features use some parameters that may be adjusted to produce optimal results in terms of detection performance. Hence, a main objective to accomplish before training the high-level classifiers, was to set-up the low-levels descriptors with the optimal configuration.

The *mutual information* (MI) has been used to measure the relevance of each potential input feature X_k to the classification decision Y (shot label). The mutual information I can be defined as follows:

$$I(X_k, Y) = D\left[\rho(x_k, y) || \rho(x_k)\rho(y)\right] \tag{1}$$

where D is the Kullback-Leibler divergence or relative entropy between the joint distribution and the product of the marginal distributions of random variables X_k and Y. Equation (1) can be rewritten as follows:

$$I(X_k, Y) \quad = \quad H(Y) - H(Y|X_k) \tag{2}$$

where $H(Y)$ is the marginal entropy and $H(Y|X_k)$ is the conditional entropy of Y after X_k is known. Intuitively, the *mutual information* measures the amount of uncertainty in Y which is removed by knowing X_k. Obviously, when X_k and Y are totally independent the value of mutual information is zero. In our implementation, a Mutual Information estimator that finds the least dependent components under a linear transformation has been used. The interested reader is referred to [5] for a complete description of the method.

The adjustment process is as follows: for each of the high-level concepts in TRECVID 2008, a training set including both positive (keyframes belonging to shots that contain the concept) and negative images was used to obtain individual MI values. Then, the MI values for each concept were linearly averaged to allow for comparisons between different configurations. Next, we provide the list of low-level features that were involved in the study, as well as the main results obtained from the MI estimator:

- *MPEG-7 Color Structure (CS)*: described in [6]. This feature uses a color quantization that accepts different number of bins nb. In our study, the array $nb = \{16, 32, 64, 128, 256\}$ was tested, obtaining the best result for $nb_{opt} = 32$.
- *MPEG-7 Dominant Color (DC)*: described in [6]. This descriptor allows for the adjustment of two parameters: (a) the color space *sc*=\{*RGB, YCbCr, HSV, HMMD, Linear transformation, Monochrome*\}, and (b) the number of bins used in the color quantization $nb = \{8, 16, 32, 64, 128\}$. The pair that achieved the best results in our experiments was $\{sc, nb\}_{opt} = \{HMMD, 32\}$.
- *MPEG-7 Scalable Color (SC)*: described in [6]. In this case, two parameters were modified, namely (a) the number of coefficients used to represent the histogram $nc = \{16, 32, 64, 128, 256\}$ and, (b) the number of discarded bitplanes $db = \{0, 8\}$. The best pair was found to be $\{nc, db\}_{opt} = \{32, 0\}$.
- *MPEG-7 Homogeneous Texture (HT)*: described in [6]. This descriptor accepts two configurations *ly*=\{*Base layer, Full layer*\}, that are related to the number of

coefficients used in its computation. From our experiments the *Full Layer* was selected as the final configuration.

- *MPEG-7 Texture Browsing (TB)*: described in [6]. Again, this descriptor can work at two different layers *ly*={*Base layer, Full layer*}, but in this case, the *Base Layer* was found to be the optimal one.

- *Color Correlogram (CC)*: proposed in [7], it extends the histogram by incorporating information about the spatial distribution of pixels. Consequently, this feature is computed at several scales. Two parameters were evaluated: (a) the number of bins in the color quantization $nb = \{8, 64\}$, and (b) the number of scales at which the correlogram is computed $ns = \{1, 2, 3, 4\}$. The pair that showed the best performance was $\{nb, ns\}_{opt} = \{8, 3\}$.

- *Color Auto-Correlogram (AC)*: the Auto-correlogram is a particularization of the correlogram that computes spatial distributions of pairs of pixels with the same color. Hence, the descriptor parametrization remains the same as for the Color Correlogram, and the best pair in this case is $\{nb, ns\}_{opt} = \{8, 2\}$.

- *Gray Level Co-ocurrence Matrix (GLCM)*: proposed in [8], the GLCM captures the spatial relations that give place to textures at several scales and orientations. Due to the complexity and length of the GLCM, several measures have been suggested in [8] to represent the matrix in a compact form. In particular, our implementation uses the computed variance at each scale and orientation in order to index the texture information. This descriptor employs a gray-level quantization, thus using a parameter related to the number of bins $nb = \{8, 32, 64, 128, 256\}$. After the experiments, the best value was found to be $nb_{opt} = 64$.

- *Gabor Wavelet (GW)*: In this descriptor a bank of Gabor filters with various scales $sc = \{2, 4, 6\}$ and orientations $or = \{2, 4, 6\}$ is created using the Wavelet transform. Following the implementation given in [9], the output of each filter is employed to compute two measures, the mean and the variance, that are then used for indexing. The pair that turned to be the optimal one was $\{sc, or\}_{opt} = \{4, 4\}$.

2.2. Low-level features

High level feature extraction task relies on low-level features that are generated from the audiovisual content. In our system, low-level descriptors have been computed at different levels or granularities, namely (a) *Keyframe Level features*, that describe image content on each keyframe, (b) *Regular Grid features*, which apply some kind of spatial regionalization by dividing the image into a regular grid, and (c) *Local features* that detect and describe specially discriminant areas in the images. Next, we provide a brief description of them:

Keyframe level features: Each keyframe extracted from the video content is described by means of several image descriptors. In particular, all the descriptors involved in the MI study (Section 2.1), as well as the *MPEG-7 Color Layout* (CL) and *Edge Histogram* (EH) descriptors have been included in our system.

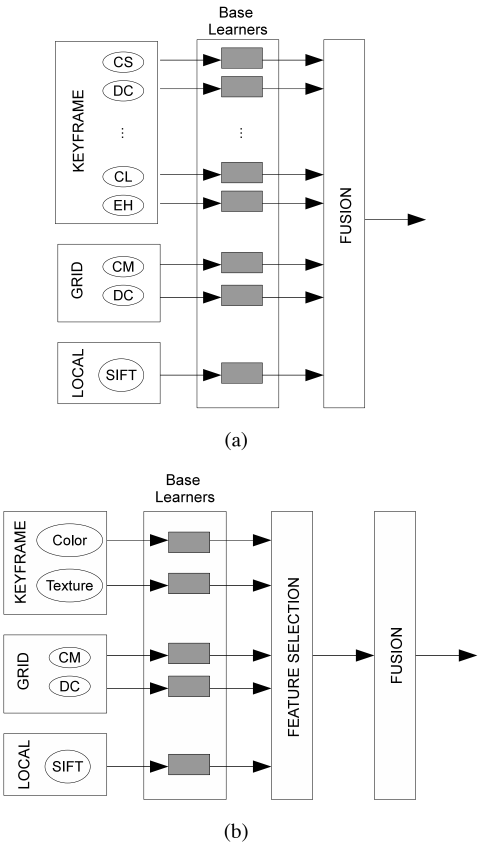
Regular grid features: Each keyframe has been divided using a regular grid of type 4x3. Each cell has been annotated using two compact descriptors: *Color Moments* (CM) (up to 3rd-order) in HSV color space, and *Gabor Wavelets* (GW) with two scales and four orientations.

Local features (bag-of-words): Using two affine covariant region detectors, Hessian Affine Detector and Maximally Stable Extremal Regions Detector (MSER) described in [10], a set of elliptical regions is extracted for each keyframe. Then, each region (also known as keypoint) is described using a 134-dimensional descriptor that concatenates a 128-dimensional SIFT descriptor [11] and a 6-dimensional color descriptor (means and variances in CIELab color space).

Once the SIFT-based features have been extracted, a Bag-of-words model is built which generates a codebook of visual words. A simple clustering technique like K-means has been used to compute those codewords that seem to consistently appear in the video corpus. Then, each image is vector-quantized so that each region descriptor is assigned to its closest codeword and a normalized histogram of words is computed. This model allows the system to work with fixed-length input vectors of size N ($N = 1000$ in our implementation), which corresponds to the size of the vocabulary.

2.3. Mid-level features

The system also incorporates a mid-level feature which is specifically suited for the *Female-human-face-closeup* category. The mid-level feature makes use of the face detector included in OpenCV [12] so that the layouts that contain faces are firstly marked. Then, the same procedure as in the local feature extraction is followed but, now, only the descriptors belonging to marked layouts are processed. The objective of this approach is to generate a highly discriminant bag-of-words model for this category, since it directly works on boxes associated to faces and thus generates a specific vocabulary for them.



(a)

(b)

Fig. 1. General system architecture when Keyframe features are employed in a late fusion scheme (Subfigure (a)) and in an early fusion scheme (Subfigure (b)).

3. HIGH LEVEL FEATURE EXTRACTION

In this section we are going to describe how the different runs have been created. As we have already explained, we have considered, as our starting point, a modular system architecture made up of four processing steps: (1) a low level feature extraction layer, (2) a set of supervised learning machines, (3) a learner outputs selection step and (4) a final fusion stage. Hence, in the following subsections, we will firstly describe which elements have been studied as potential components in the system and which system configurations have been selected as the runs submitted to TRECVID. Secondly, a comprehensive study of the bag-of-words model will be detailed, including a comparison between discriminative and generative approaches. As mentioned before, two of the runs were reserved for this study, mainly due to the high performance achieved by the bag-of-words features in several computer vision tasks.

3.1. Early/late fusion of keyframe low-level features

Once all the low level features have been extracted from the video data (as it has been described in Section 2), they are employed to train a classifier which solves, for each category, the desired classification problem. Regular grid features, Local features and the Mid-level features (if *Female-human-face-closeup* category is considered) are directly used as inputs of the classifiers; however, keyframe features have been grouped according to two different architectures (see Figure 3.1):

- *Late fusion*: each feature becomes an input of a different base learner.
- *Early fusion*: the keyframe level features have been merged into two new groups:
  - Color Early Fusion: which includes all keyframe level features which are related to color.
  - Texture Early Fusion: which includes all keyframe level features related to texture.

3.2. Classification technologies

Most TRECVID participants consider standard SVMs as their preferred classification technology, mainly due to their good generalization capability in the absence of large training corpus (in the TRECVID case, the number of positive samples is not very large). However, this year we wanted to check whether other classification technologies can provide similar o better performance. For this purpose, we have considered three alternatives for the base learners in our system:

- *Standard SVM*: this first classifier consists in a C-SVM [1] trained with the LIBSVM toolbox [13]. Due to the fact that the training data set has few positive instances in comparison to the negative ones, different weights have been assigned to the positive and negative classes to alleviate this problem.
- *A modified ranking SVM*: due to the fact that our final accuracy parameter is the InfAP, we must rank the video data so that all relevant videos are placed in the first ranking positions. For this purpose, it can be more adequate to employ a ranking SVM [14] instead of a classical C-SVM. Ranking SVM modifies the constraints of the classical SVM to force the outputs to follow a predefined rank. For instance, in document or video retrieval applications, it is convenient to employ a modified ranking SVM [2] which can force relevant data outputs to be larger than the outputs of non-relevant instances. Let us denote the subset of N^+ relevant data as $\{\mathbf{x}_i^+\}_{i=1}^{N^+}$ and the subset of N^- irrelevant data as $\{\mathbf{x}_j^-\}_{j=1}^{N^-}$. Then, the modified ranking SVM enforces:

$$f\left(\mathbf{x}_i^+\right) > f\left(\mathbf{x}_j^-\right) \quad i = 1, \ldots, N^+,\ j = 1, \ldots, N^- \quad (3)$$

This modified ranking SVM not only seems more appropriate to maximize the AP parameter, but it also alleviates the problem of the unbalanced number of positive/negative samples in each class.

- *A KOPLS approach*: KOPLS is a kernel multivariate analysis technique for feature extraction which finds optimal projections of the input data in the feature space. KOPLS extracted features are optimal in the sense that they minimize the quadratic error when used to reconstruct the labeled data. In the proposed system, we use the compact approximation to KOPLS presented in [3], followed by a simple least squares classifier.

3.3. Performance measures for validation

Another aspect to be taken into account is the performance measure that serves to adjust the free parameters of the above learners by means of a validation procedure. TRECVID results are measured in terms of InfAP, therefore, it seems reasonable to use the AP as the validation criterion. However, due to the high influence of the first ranked documents in the final value, using the AP may not be robust enough. Including the AP, we have considered three different performance measures:

- *Average precision*: Due to fact that validation data are labeled, we can directly obtain the average precision (AP) [15] value rather than the infAP. Considering the sorted list of videos returned by the system, the AP is the average of the precisions computed after truncating the list after each of the relevant videos:

  $$AP = \frac{\sum_{r \in R} \text{\# relevant videos in the first } r \text{ elements}/r}{\text{\# relevant videos}} \quad (4)$$

  where R is a index set with the positions of the relevant videos.

- *"Balanced" classification error*: This parameter is computed by averaging the classification error over the positive instances (CE^{+}) and over the negative ones (CE^{-}), i.e.,

  $$\text{Balanced CE } = \frac{1}{2}\left(CE^{+} + CE^{-}\right) \quad (5)$$

  This approach tries to compensate the unbalanced distribution of data (positive/negative samples) in each class.

- *F-score*: The F-score [4] is often used in the field of information retrieval for measuring search, document classification, and query classification performance. It can be calculated as the harmonic mean of precision and recall:

  $$\text{F-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

  where the precision is the number of correct returned results divided by the total number of returned results (i.e., the total number of documents classified as positive), and the recall is the number of correct returned results divided by the total number of positive samples. Then, the F-score parameter reaches its maximum value at 1 and provides the worst score at 0.

3.4. Classifier output selection

After all base learners have been trained, their outputs have to be combined to obtain the final output of the global scheme. However, we have observed that some of the learner outputs show a very poor performance; therefore, removing these learners from the combination can be very useful, thus enhancing the overall system performance. For this purpose, we have tested some of the feature selection methods found in the literature [16]. However, these approaches usually selected the SIFT features as the unique relevant features. To obtain a richer representation, we have preferred to use a wrapping method and apply an exhaustive search to find the best subset of classifier outputs. Note that this method is not efficient in computational terms but, since just early fusion features are considered, the number of subsets to be evaluated is 32, which is computationally feasible[1].

3.5. Fusion stage

Finally, the (selected) classifier outputs have been linearly combined in order to obtain the global system output. To carry out this combination, we have considered the two kind of SVMs that have been mentioned before: a classical SVM that maximizes the classification accuracy, and a ranking SVM that enforces that the outputs for relevant data are larger than the values for the irrelevant ones. Unlike the validation process for the base learners, here we just consider the AP to adjust the free parameters of the combination, since this is the ultimate performance measure we want to optimize.

3.6. Selection of the submitted runs

Combining all the previous alternatives in our system, we have built a great variety of possible runs. In Table 1, we show a selection of the most relevant system configurations, indicating which elements make up the system, as well as their final validation AP value (averaged over the 20 categories). The configurations that have been submitted for evaluations are pointed out in boldface and their run number is included in the first column.

As a first conclusion, we see that very similar APs are obtained in all cases, which indicates that there are not very significant differences among the studied combinations. However, we can extract some preliminary conclusions:

[1] This is the reason why the feature selection stage has only been included in the scheme with early fusion (see Fig. 1(b)).

Table 1. Validation AP values of the most relevant system configurations.

# RUN	# conf.	Keyframe Features	Base learners	Validation method	F.E.	Fusion method	AP value
	1	Late	SVM	AP	No	SVM	0.1202
	2	Early	SVM	AP	No	SVM	0.1343
1	**3**	**Early**	**SVM**	**AP**	**Yes**	**SVM**	**0.1545**
	4	Early	KOPLS	AP	No	SVM	0.1357
2	**5**	**Early**	**KOPLS**	**AP**	**Yes**	**SVM**	**0.1494**
	6	Early	SVM$_{\text{rank}}$	AP	No	SVM	0.0955
	7	Early	SVM$_{\text{rank}}$	AP	Yes	SVM	0.1362
	8	Early	SVM	Balanced CE	No	SVM	0.1235
	9	Early	SVM	Balanced CE	Yes	SVM	0.1439
	10	Early	SVM	F score	No	SVM	0.1463
4	**11**	**Early**	**SVM**	**F score**	**Yes**	**SVM**	**0.1596**
	12	Late	SVM	AP	No	SVM$_{\text{rank}}$	0.1266
	13	Early	SVM	AP	No	SVM$_{\text{rank}}$	0.1395
	14	Early	SVM	AP	Yes	SVM$_{\text{rank}}$	0.1534
	15	Early	KOPLS	AP	No	SVM$_{\text{rank}}$	0.0602
	16	Early	KOPLS	AP	Yes	SVM$_{\text{rank}}$	0.1278
	17	Early	SVM$_{\text{rank}}$	AP	No	SVM$_{\text{rank}}$	0.1197
	18	Early	SVM$_{\text{rank}}$	AP	Yes	SVM$_{\text{rank}}$	0.1357
	19	Early	SVM	Balanced CE	No	SVM$_{\text{rank}}$	0.0382
	20	Early	SVM	Balanced CE	Yes	SVM$_{\text{rank}}$	0.1064
	21	Early	SVM	F score	No	SVM$_{\text{rank}}$	0.1450
3	**22**	**Early**	**SVM**	**F score**	**Yes**	**SVM$_{\text{rank}}$**	**0.1591**

- The early fusion architecture shows better performance than the late fusion scheme (compare configurations 1 *versus* 2 and 12 *versus* 13). This result suggests that combining some subsets of low level features can help to exploit the relations between these features, thus improving the final system performance. Besides, the early fusion scheme has an additional computational advantage, since it reduces the number of learners in the subsequent phases.

- Removing the useless classifier outputs (i.e., applying the feature selection process) systematically improves system performance, what was to be expected due to the fact that an exhaustive wrapping search has been applied.

- Using a classical SVM as the base classification machine is the best option in most cases (compare configurations 3 *versus* 5 and 7, or 14 *versus* 16 and 18); however, in the special case of the linear combination of the base classifier outputs, the modified ranking SVM outperforms the classical SVM in some cases (see configurations 1 and 2 *versus* 12 and 13).

- The performance measure that is used in the validation of base learners also influences system performance, being the F-score and the balanced CE the ones that achieve the best and worst results, respectively.

To corroborate the above conclusions, we have selected the following four runs that were submitted for evaluation to TRECVID:

- **RUN 1** (*“A UC3M 2 1”*): this is our baseline, which is characterized by including early fusion keyframe features, considering a set of classical SVMs validated with the AP parameter as classification learners, applying a feature selection process to remove the useless learner outputs, and employing a linear SVM to fuse the selected learner outputs.

- **RUN 2** (*“A UC3M 4 2”*): this run has the same configuration as RUN 1, but KOPLS technology is used as the classification technology instead of a classical SVM.

- **RUN 4** (*“A UC3M 15 4”*): this run also has the same configuration as the baseline, but now the F score as the validation criterion to adjust the SVM free parameters.

- **RUN 3** (*“A UC3M 35 3”*): this run employs the same configuration as RUN 4, but the fused output is generated as a linear combination carried out with a ranking SVM.

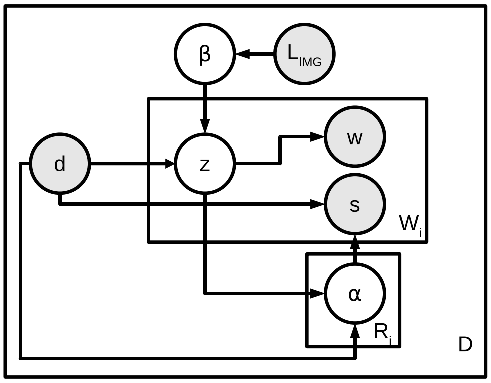


Fig. 2. Supervised Generative Model. Shaded circles represent observations from the images while white circles indicate hidden variables that need to be inferred.

3.7. Bag-of-words model: discriminative vs generative approaches

Additionally, this year we wanted, to compare discriminative and generative approaches to the bag-of-words model. Since this model have shown great performance in several computer vision tasks, such as object detection and image classification, it is worth making special emphasis on its study. Hence, the last two submitted runs are associated to this kind of models, namely:

- **RUN5** (*"A UC3M 50 5"*): this run tests the performance of a generative bag-of-words approach that models the spatial distribution of visual words along the visual documents.

- **RUN6** (*"A UC3M 51 6"*): this run is associated to a basic discriminative bag-of-words model that uses a SVM over the histograms of visual words.

Generative Model for concept detection

This approach tries to extend the well-known bag-of-words techniques in order to model the spatial distribution of visual words along a document. Bag-of-words models have shown great performance in several computer vision tasks, such as image classification or topic discovery. However, they do not take into account the spatial distribution of visual words in an image, thus, showing a strong limitation in their performance. Good examples of basic bag-of-words models can be found in the literature, such as the discriminative approaches in [17] and [18], and the well-known generative models *Probabilistic Latent Semantic Analysis* (PLSA) [19] and *Latent Dirichlet Allocation* (LDA) [20].

Since in the aforementioned models there are no constraints on the spatial position of the words (associated to keypoints in images), the distribution of topics along the visual words is often inaccurate. Consequently, the localization of specific topics in images is weak, thus unrealistically modelling semantic concepts.

This approach builds upon those works and proposes a generative model that considers a document as a set of interrelated regions that interact according to the image geometric layout. The objective of the model is to integrate local descriptors that have been found to be highly discriminative with global segmentations that depict the spatial structure of the documents. This implies the use of a prior segmentation step which, in our case, employs a simple and fast algorithm that is particularly configured to produce about 30-60 regions (oversegmentation). This configuration ensures that most of the regions do not contain pixels associated to more than one semantic object. Particularly, a Recursive Shortest Spanning Tree (RSST) [21], is employed to generate color-based segmentations for each of the images.

In Fig. 2, each document d is viewed as a mixture of latent topics z that model the occurrences of local visual words w at a predefined set of spatial locations s. Visual words w describe the local appearance of a local patch (texture and color) while the spatial location s basically denote which of the R regions a local patch belongs to. The spatial positions s are document-dependent and also generated by a term α that provides interactions between regions. This approach allows one region to influence each others so topics spread along an image, thus producing more coherent classifications. In TRECVID 2009, the supervised version of the algorithm is employed so that an image label L_{IMG} generates a Dirichlet prior distribution (with parameter β) over the probabilities of topics given the document. A complete description of the generative model as well as the detailed formulation can be found in [22].

The generative model is used to compute a vector with the topic probabilities given a document. Then, this vector feeds a SVM that produces the final output of the detector.

4. PERFORMANCE EVALUATION

In this section we evaluate the performance of all six runs in terms of InfAP. To start with, Figure 4 illustrates the achieved InfAP averaged over the 20 high-level concepts. The best result, median and the 25% and 75% percentiles are also shown in the figure as a reference for comparison. Our submitted runs have achieved average InfAP values from 0.05 (run 6) to 0.091 (run 3), what places our designs in the second quartile of all TRECVID 2009 submitted runs.

As it was expected, runs generated with the general system architecture present better performance than runs 5 and 6, which only used bag-of-words models. However, the differences between run 6 and runs 1, 2, 3 and 4 are not very significant, what demonstrates that the main contribution to the good system performance is provided by the discriminative bag-of-words model.

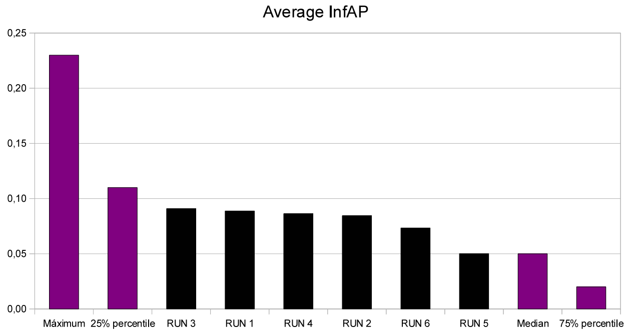


Fig. 3. Average InfAP of all our submitted runs. Results are shown in comparison with the best performing run, median, and 25% and 75% percentiles.

We cannot appreciate important differences among runs 1 to 4. When we analyze the results obtained in the different classes (see Figure 4), we observe that runs 1, 2, 3 and 4 obtain the best performances in 8, 5, 4 and 3 of the target classes, respectively. Furthermore, in categories such as *Classroom*, *Intersection*, *Doorway*, *Eating*, *Hand* or *Female*, some of our runs were above the 25% percentile. On the contrary, runs 5 and 6 which are just based on bag-of-words features, never got values over the 25% percentile.

Regarding the comparison between the bag-of-words models, it is noteworthy that the discriminative approach achieves better average performance than the generative one, a result that is consistent along most of the categories (see Fig. 4(b)). The rationale behind is that the generative model assumes independence between visual words: although it models the spatial distribution of words along a document, the probabilities of the visual words given the topics are fully factorized, what does not happen in the discriminative approach. In the latter case, the histograms can be considered as probability distributions, thus capturing inter-relations between words. However, since the complexity of the generative model is linear with the size of visual codebook, this model provides a dimensionality reduction of the inputs (from 1000 visual words to a maximum of 20 topics in our tests).

5. CONCLUSIONS

In this document we have presented the architecture of our system for high-level feature extraction, describing the involved modules and indicating which of them have been implemented so far. Experiments this year have been focused on analyzing different system configurations and comparing discriminative and generative approaches to the bag-of-words model.

Following the first line, we have submitted 4 runs, achieving in all cases performances close to TRECVID 25% percentile. To compare bag-of-words models we have submitted two additional runs, which achieved performances between the median and 25% percentile of all submitted runs.

The obtained results have pointed out the importance of including local features and the bag-of-words model in the global design. Hence, one of the main research lines for future participations in TRECVID will consist in experimenting with different parameters in the bag-of-words model (type of region detectors, visual descriptors), as well as testing different extensions (by means of advanced discriminative and generative models).

The classification technologies, validation parameters, and fusion schemes, play also an important role; however, using classical SVMs that are validated with the AP parameter, and fusing their outputs with a linear SVM not only provided a simple system configuration but also an accurate solution.

6. REFERENCES

[1] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

[2] Y .Cao, J. Xu, T. Y.Liu, H. Li, Y. Huang, and H. W. Hon, “Adapting ranking SVM to document retrieval,” in *SIGIR ’06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 2006, pp. 186–193, ACM.

[3] J. Arenas-García, K. B. Petersen, and L. K. Hansen, “Sparse kernel orthonormalized pls for feature extraction in large data sets,” in *Advances in Neural Information Processing Systems 19*, MIT press, Ed., Cambridge, MA, 2007.

[4] C. J. Van Rijsbergen, *Information Retrieval*, Butterworth-Heinemann, Newton, MA, USA, 1979.

[5] H. Stögbauer, A. Kraskov, S. A. Astakhov, and P. Grassberger, “Least-dependent-component analysis based on mutual information,” *Phys. Rev. E*, vol. 70, no. 6, pp. 066123, Dec 2004.

[6] J. M. Martinez, “Overview of the MPEG-7 Standard, v 10.0,” *ISO/IEC JTC1/SC29/WG11 N4674*, Jeju, Mar. 2002.

[7] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih, “Image indexing using color correlograms,” p. 762, 1997.

[8] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 3, no. 6, pp. 610–621, 1973.

[9] B. S. Manjunath and W.Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI - Special issue on Digital Libraries)*, vol. 18, no. 8, pp. 837–42, Aug 1996.

[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, “A comparison of affine region detectors,” *International Journal of Computer Vision*, vol. 65, no. 1/2, pp. 43–72, 2005.

[11] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[12] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[13] C. Chang and C. Lin, “LIBSVM: a library for Support Vector Machines,” 2001, Software available at http://www.csie.ntu.edu.tw/c̃jlin/libsvm.

[14] R. Herbrichv, T. Graepel, and K. Obermayer, “Large margin rank boundaries for ordinal regression,” in *Advances in Large Margin Classifiers*, A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds., Cambridge, MA, 2000, pp. 115–132, MIT Press.

[15] C. Buckley and E. M. Voorhees, “Evaluating evaluation measure stability,” in *In Proc. 23rd Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, ACM press, Ed., 2000, pp. 33–40.

[16] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal Machine Learning Resource*, vol. 3, pp. 1157–1182, 2003.

[17] C. Wallraven, B. Caputo, and A. Graf, “Recognition with local features: the kernel recipe,” in *IEEE International Conference on Computer Vision*, Oct. 2003, pp. 257–264 vol.1.

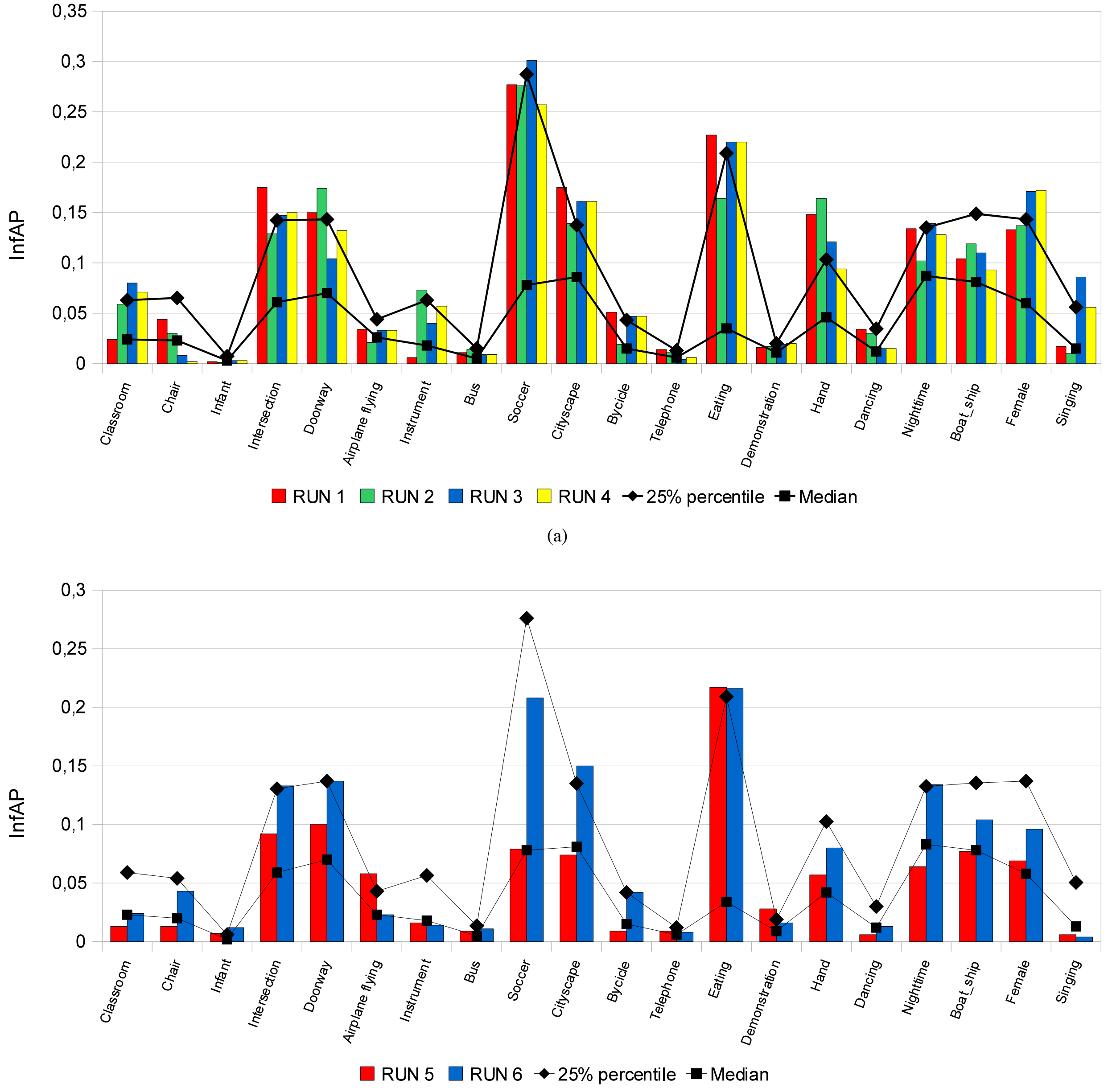
[18] K. Grauman and T. Darrell, “The pyramid match kernel: discriminative classification with sets of image features,” in *IEEE International Conference on Computer Vision*, Oct. 2005, vol. 2, pp. 1458–1465 Vol. 2.

[19] T. Hofmann, “Unsupervised learning by Probabilistic Latent Semantic Analysis,” *Machine Learning*, vol. 42, no. 1/2, pp. 177–196, 2001.

[20] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty, “Latent Dirichlet Allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 2003, 2003.

[21] S. H. Kwok and A. G. Constantinides, “A fast recursive shortest spanning tree for image segmentation and edge detection,” *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 328–332, Feb 1997.

[22] I. González-Díaz, D. García-García, and F. Díaz de María, “A spatially aware generative model for image classification, topic discovery and segmentation,” in *IEEE International Conference on Image Processing, 2009. (ICIP’09).*, 2009.



(a)

(b)

Fig. 4. InfAP per class of all submitted runs. TRECVIDs 25th percentile and median are also included as a reference for comparison.