An Empirical Study of Multi-Label Learning Methods for Video Annotation

Anastasios Dimou*, Grigorios Tsoumakas†, Vasileios Mezaris*, Ioannis Kompatsiaris*, Ioannis Vlahavas†

*Informatics and Telematics Institute
6th Km Charilaou-Thermi Rd
Thessaloniki 57001, Greece
Email: {dimou,bmezaris,ikom}@iti.gr

†Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki 54124, Greece
Email: {greg,vlahavas}@csd.auth.gr

Abstract

This paper presents an experimental comparison of different approaches to learning from multi-labeled video data. We compare state-of-the-art multi-label learning methods on the Mediamill Challenge dataset. We employ MPEG-7 and SIFT-based global image descriptors independently and in conjunction using variations of the stacking approach for their fusion. We evaluate the results comparing the different classifiers using both MPEG-7 and SIFT-based descriptors and their fusion. A variety of multi-label evaluation measures is used to explore advantages and disadvantages of the examined classifiers. Results give rise to interesting conclusions.

1. Introduction

Almost every image and video is depicting content with a large variety of objects, concepts or situations. Therefore the annotation process can tag it with a large number of labels. The correlation between those labels can be exploited towards more robust semantic analysis and retrieval methods. Lately, multi-label based classifiers are becoming popular in the field of semantic image and video analysis [1], [2], [3], [4], [5].

This paper presents an experimental comparison of different approaches to learning from multi-labeled video data. We compare state-of-the-art multi-label methods on the Mediamill Challenge dataset [5]. MPEG-7 and SIFT-based global image descriptors are independently employed. Subsequently, the classifiers are fused using variations of the stacking [6] approach. The evaluation of all possible combinations of classifiers and descriptors, as well as their fusion, is performed with a wide collection of multi-label measures. This thorough investigation gave some insight on the strong and weak points of the examined classifiers and their fusion technique.

The rest of this paper is structured as follows. The next section gives background information on multi-label classification, including descriptions of the algorithms that participate in this empirical study. Section 3 describes all the details of the experimental setup including the approaches followed for fusing the different descriptors. Section 4 presents the results and their discussion. Finally, Section 5 concludes and points to interesting extensions of this work for the future.

2. Multi-label Classification

Traditional *single-label* classification is concerned with learning from a set of examples that are associated with a single label λ from a set of disjoint labels L, $|L| > 1$. In *multi-label* classification, the examples are associated with a set of labels $Y \subseteq L$.

Multi-label classification algorithms can be categorized into two different groups [7]: i) *problem transformation* methods, and ii) *algorithm adaptation* methods. The first group includes methods that are algorithm independent. They transform the multi-label classification task into one or more single-label classification, regression or ranking tasks. The second group includes methods that extend specific learning algorithms in order to handle multi-label data directly. Algorithms of both groups can improve their performance by taking label relationships into account.

We next present the four methods used in the experimental part of this work. For the formal description of these methods, we will use $L = \{\lambda_j : j = 1 \dots M\}$ to denote the finite set of labels in a multi-label learning task and $D = \{(\vec{x_i}, Y_i), i = 1 \dots N\}$ to denote a set of multi-label training examples, where $\vec{x_i}$ is the feature vector and $Y_i \subseteq L$ the set of labels of the i-th example.

Binary relevance (BR) is a popular problem transformation method that learns M binary classifiers, one for each different label in L. It transforms the original data set into

M data sets $D_{\lambda_j}, j = 1 \dots M$ that contain all examples of the original data set, labeled positively if the label set of the original example contained λ_j and negatively otherwise. For the classification of a new instance, BR outputs the union of the labels λ_j that are positively predicted by the M classifiers.

Label powerset (LP) is a simple but effective problem transformation method that works as follows: It considers each unique set of labels that exists in a multi-label training set as one of the classes of a new single-label classification task. Given a new instance, the single-label classifier of LP outputs the most probable class, which is actually a set of labels. The number of different classes is upper bounded by $\min(N, 2^M)$ and despite that it typically is much smaller, it still poses an important complexity problem for LP, especially for large values of N and M. The large number of classes, many of which are associated with very few examples, makes the learning process difficult as well.

The random k-labelsets (RAkEL) method [8] constructs an ensemble of LP classifiers. Each LP classifier is trained using a different small random subset of the set of labels. This way RAkEL avoids LP's scalability and learning problems. A ranking of the labels is produced by averaging the zero-one predictions of each model per considered label. Thresholding is then used to produce a classification as well.

BP-MLL [9] is an adaptation of the popular back-propagation algorithm for multi-label learning. The main modification to the algorithm is the introduction of a new error function that takes multiple labels into account.

ML-kNN [2] extends the popular k Nearest Neighbors (kNN) lazy learning algorithm using a Bayesian approach. It uses the maximum a posteriori principle in order to determine the label set of the test instance, based on prior and posterior probabilities for the frequency of each label within the k nearest neighbors.

BR was included in the experimental study as it is a popular, baseline approach for dealing with multi-label data. RAkEL, MLkNN and BP-MLL were included in the experimental study as representative samples of the recent state-of-the-art in multi-label classification.

3. Experimental Setup

3.1. Dataset

The set of experiments conducted was based on the Mediamill Challenge dataset [5]. This contains the video data found in the training set of the 2005 NIST TRECVID competition. It has become a benchmark set due to its diverse real-life content and the extensive multi-label annotation that is available for it. The video sources include Arabic, Chinese, and US news broadcasts that have been recorded during November 2004 by the Linguistic Data Consortium. The dataset contains, in total, 85 hours of video data. Although audio data is also available, it is not taken into consideration in our experiments. Our setup is exclusively based on still image data from the shot keyframes extracted.

A manual annotation for 39 labels was performed by the TRECVID 2005 common annotation effort. This annotation was extended to a lexicon of 101 semantic labels, by the Mediamill team [5]. This extended annotation is offering a collection of multi-labeled shots, suitable for multi-labeled classification.

3.2. Features

The experiments are based on two different feature sets. The features are built on the widely accepted MPEG-7 and the state-of-the-art SIFT-based global image features respectively. A set of MPEG-7 features, namely color structure, color layout, edge histogram, homogeneous texture and scalable color, are extracted using the Experimentation Model (XM) reference software. All features are concatenated in a single MPEG-7 vector.

The SIFT-based feature vector is created for each keyframe in a 2-stage procedure. A set of 500 keypoints, on average, is extracted from each keyframe. Each keypoint is assigned a SIFT descriptor vector [10] (with 128 elements). The SIFT descriptors from all keyframes create a new 128-dimensional feature space. After clustering this feature space, a lexicon of 100 Visual Words is created. Using the "Bag of Words" (BoW) [11] methodology and the aforementioned lexicon, a frequency histogram is created for each keyframe. This histogram is a 100-dimensional feature vector and is used as the second SIFT-based feature in our experiments.

3.3. Algorithms and Parameter Settings

We compare the following four multi-label learning algorithms that were described in Section 2: BR, RAkEL, BPMLL and MLkNN.

For the training of RAkEL, BPMLL and MLkNN, the Mulan open-source library for multi-label classification was used [8]. RAkEL is run using m=100 k=4 and the C4.5 algorithm as underlying single-label classifier. As recommended in [2], MLkNN is run with 10 nearest neighbors and a smoothing factor equal to 1. As recommended in [9], BPMLL is run with 0.05 learning rate, 100 epochs and the number of hidden units equal to 20% of the input units.

For the training of BR, the LIBSVM library for Support Vector Machine (SVM) algorithms was used [12]. The SVM parameters were set by an automatic optimization procedure. The procedure pursues the best possible values for the parameters cost (C) and gamma (g). A grid search is employed for values of C: log2(C)= -5:15 (with a step of 2) and gamma: log2(g)=3:-15 (with a step of 2) to find the optimum values for each label.

3.4. Fusion Approaches

For the fusion of the two feature vectors we employ the paradigm of stacking [6], as simpler fusion schemata - such as those discussed in [13] - have not led to improvements of performance in the past [14]. In specific, we initially learn two base-level models, one from each feature vector, and then we employ a meta-level model that learns from the predictions of the base-level models.

For MLkNN, BPMLL and RAkEL, a 5-fold cross-validation procedure was used on the training set, in order to produce an equal sized meta-level training set containing the Degrees of Confidence (DoCs) for each label and training example.

For BR, all positive examples are gathered, sorted by their position in the training set, and split in two parts, namely one for the base level training and one for the meta-level training procedure that will follow. The first set of training is assigned the odd numbered positive examples and the fusion training set the even numbered ones. This splitting scheme has been chosen to ensure that keyframes from all parts of the training set will be included in both training sets. Each binary classifier of BR was trained using a random sample of negative examples five times the size of the positive examples. This was done to avoid the use of a highly skewed training set when the number of positive examples available for training is disproportionately low compared to the negative ones. The output of classification for an image, regardless of the employed input feature vector, is a number in the continuous range [0, 1] expressing the DoC that the image relates to the corresponding label. Using these classifiers, DoCs are extracted for the fusion development set. The set of formed DoC vectors (one per image and label) serves as a training set for another set of BR classifiers, that realize the meta-level classification. Their training (i.e. selection of a subset of negative examples; optimization of parameters, calculation of output DoC, etc.) is performed using the methodology described above for the base-level case. Having completed all the training processes, MPEG-7 and BoW feature vector extraction followed by the base and meta SVM classification levels are performed on the test dataset.

3.5. Evaluation Methodology and Measures

The dataset is split into a development and a testing set. The development is used for the training of the label and fusion classifiers. On the other hand, the testing set is used to evaluate the classification models created.

Multi-label classification requires different evaluation measures than traditional single-label classification. A taxonomy of multi-label classification evaluation measures is given in [8], which considers two main categories: *example-based* and *label-based measures*. A third category of measures, which is not directly related to multi-label classification, but is often used in the literature, is ranking-based measures, which are nicely presented in [2] among other publications.

We compare the different approaches using a plethora of different multi-label evaluation measures from the aforementioned categories. Based on the binary decisions of the algorithms, we calculate a) the example-based measures: hamming-loss, accuracy, precision, recall, F_1-measure and subset accuracy, and b) the label-based measures micro and macro precision, recall and F_1-measure. In addition, based on the probability estimates output for each label, by the different algorithms, we further calculate the micro and macro area under the ROC curve (AUC) and the ranking-based measures: one-error, coverage, ranking loss and average precision.

4. Results and Discussion

4.1. Algorithm Comparison

Table 1 presents the results of the algorithms on the different measures using the BoW descriptors. We notice that the MLkNN multi-label learning algorithm achieves the best result in almost all measures. An exception is noticed for the macro-averaged measures, where BR dominates the rest of the algorithms, due to its high recall. In addition, BPMLL achieves the best value for the micro-averaged and example-based recall measures.

	BR	RAkEL	MLkNN	BPMLL
Ham. Loss	0.0568	0.0362	**0.0323**	0.0674
Accuracy	0.2380	0.3841	**0.4174**	0.3097
Precision	0.3620	0.6308	**0.7077**	0.3591
Recall	0.3750	0.4734	0.4646	**0.6918**
F_1	0.3684	0.5409	**0.5609**	0.4728
Subset Acc.	0.0158	0.0567	**0.1098**	0.0001
Micro Prec.	0.3614	0.6305	**0.7311**	0.3581
Micro Rec.	0.3732	0.4377	0.4257	**0.6649**
Micro F_1	0.3672	0.5167	**0.5381**	0.4655
Micro AUC	0.8140	0.8283	**0.9305**	0.9207
Macro Prec.	0.1487	0.2270	**0.2936**	0.0733
Macro Rec.	**0.2734**	0.0658	0.0872	0.1134
Macro F_1	**0.1475**	0.0778	0.1151	0.0690
Macro AUC	**0.7306**	0.5676	0.6450	0.6623
One-error	0.4426	0.2330	**0.1972**	0.2376
Coverage	40.2475	39.0799	**19.3747**	20.4194
Ranking Loss	0.1808	0.2687	**0.0598**	0.0650
Avg. Precis.	0.4337	0.6167	**0.6880**	0.6388

Table 1. Results using the BoW descriptors

Table 2 presents the results of the algorithms on the different measures using the MPEG-7 descriptors. We first notice that BR continues to dominate in the macro measures due to its high recall compared to the rest of the algorithms. However the results are different in the ranking-based measures, where BPMLL is dominating in this case. In the rest of

the measures, there is no clear winner with BPMLL, RAkEL and MLkNN sharing the top performances.

	BR	RAkEL	MLkNN	BPMLL
Hamming Loss	0.0471	0.0324	**0.0320**	0.0534
Accuracy	0.3651	**0.4409**	0.4329	0.3919
Precision	0.4908	0.6776	**0.6958**	0.4492
Recall	0.5413	0.5298	0.4944	**0.7407**
F_1	0.5148	**0.5947**	0.5781	0.5592
Subset Acc.	0.0524	0.0967	**0.1196**	0.0138
Micro Prec.	0.4714	0.6820	**0.7128**	0.4375
Micro Rec.	0.5411	0.4984	0.4619	**0.7291**
Micro F_1	0.5039	**0.5759**	0.5605	0.5469
Micro AUC	0.8947	0.8541	0.9357	**0.9451**
Macro Prec.	0.2240	**0.3885**	0.3026	0.2517
Macro Rec.	**0.3933**	0.1453	0.1250	0.2151
Macro F_1	**0.2458**	0.1799	0.1501	0.1857
Macro AUC	**0.8070**	0.6238	0.6610	0.7721
One-error	0.2882	**0.2045**	0.2047	0.2078
Coverage	27.6336	35.2847	18.3120	**16.1296**
Ranking Loss	0.1035	0.2267	0.0561	**0.0459**
Avg. Precis.	0.5956	0.6699	0.6975	**0.7039**

Table 2. Results using the MPEG descriptors

Table 3 presents the results of the algorithms on the different measures using the fusion approach. MLkNN performs best in most of the example-base measures. Both MLkNN and BPMLL perform very well on the ranking-based measures compared to the other two algorithms. BR continues to excel in the macro-average label based measures.

	BR	RAkEL	MLkNN	BPMLL
Hamming Loss	0.0326	0.0317	**0.0305**	0.0612
Accuracy	0.4227	0.4455	**0.4494**	0.3690
Precision	0.6781	0.6996	**0.7296**	0.4144
Recall	0.4982	0.5214	0.4991	**0.7586**
F_1	0.5744	**0.5975**	0.5927	0.5360
Subset Acc.	0.0981	0.1080	**0.1363**	0.0088
Micro Prec.	0.6962	0.7047	**0.7529**	0.3978
Micro Rec.	0.4654	0.4877	0.4603	**0.7511**
Micro F_1	0.5579	**0.5765**	0.5713	0.5202
Micro AUC	0.7642	0.8198	0.9329	**0.9448**
Macro Prec.	**0.3783**	0.2724	0.3290	0.1780
Macro Rec.	**0.2414**	0.1307	0.1239	0.2074
Macro F_1	**0.2426**	0.1575	0.1536	0.1449
Macro AUC	0.6434	0.5895	0.6325	**0.7743**
One-error	0.2185	0.1979	**0.1800**	0.1885
Coverage	49.1760	40.5621	19.3747	**16.3603**
Ranking Loss	0.2019	0.3006	0.0585	**0.0474**
Avg. Precis.	0.5768	0.6514	**0.7061**	0.7033

Table 3. Results of the fusion

The results of our experimental study show that there is a difference in the general approach followed between BR and the rest of the multi-label classifiers. BR is oriented towards ranking the results of a certain query label. Given a label, it has the competence to rank the relevance of images to this label taking a fuzzy rather than just a binary decision for this relevance, working on a single label at each time. Thus, its result is a ranked list of images relevant to a given label. On the other hand, BPMLL, MLkNN and RAkEL are working in an orthogonal way; they process simultaneously many labels trying to take advantage of the correlations between them. They are oriented towards outputting all labels that are relevant to a given image/video. BPMLL in specific, directly optimizes a loss function that takes the ranking of the labels into account. Furthermore, MLkNN outputs probabilistic estimates for each label that give very good results with respect to ranking the relevance of each label to the given object.

Given these facts, we can argue that the above methods are suitable for different types of applications. BR is more suitable for retrieval applications where the user interacts with the search engine providing a label as a search query. BR can be used for returning a list of the most relevant results, in descending order, for the given query label. On the other hand, BPMLL and MLkNN can be more suitable for semantic multimedia analysis and inference techniques. During semantic analysis, every image/shot is processed and a ranked list of labels is produced for it. Subsequently, the most plausible label can be chosen using the ranked list and predefined domain knowledge of domain objects and their temporal relationships (in analogy to [15]). The chosen label can be different from the top ranked label based on the aforementioned domain knowledge, allowing us to discard isolated erroneous classification results.

4.2. BoW vs. MPEG-7 vs. Fusion

For the BR approach we first notice that the MPEG-7 features give better results than the BoW features across *all* evaluation measures. We then notice that in micro and macro AUC, micro recall, coverage and ranking loss, the results of the fusion are worse than those when learning from the BoW features. This a strange result, since the fusion process should improve the quality of the model. Given that all of the above measures apart from micro recall, are calculated based on the degrees of confidence for each label, we conclude that the meta-level SVM models output imprecise probability estimates. A potential reason for this could be the significantly reduced feature space at the meta-level (the two confidences of the base-level SVMs).

For the MLkNN approach, we notice that the MPEG-7 features give better results than the BoW feature across all evaluation measures apart from the example-based precision, micro precision and one-error. So, in this approach we don't notice a complete dominance of the MPEG-7 features over the BoW features. A potential reason could be the higher number of MPEG-7 features (320) compared to the BoW features (100). The intolerance of irrelevant features, is a known problem of kNN based algorithms, which becomes more evident as the number of features grows. The fusion process gives better results than the BoW features as well, apart from the macro AUC measure. The fusion process improves most of the measures compared to the plain MPEG-

7 features, apart from micro and macro AUC, micro and macro recall, coverage and ranking loss. Again the decrease of performance is mostly in measures calculated based on degrees of confidences. Since in this case the feature vector is not small, as in BR, we hypothesize that a potential reason is that the meta-level data are more noisy compared to the raw MPEG-7 and BoW data, as they contain the probability estimates of the base-level models.

For the BPMLL approach, the bow features are clearly dominated by both the MPEG-7 features and the fusion process in *all* measures. Interestingly, the fusion process leads to worse results compared to the plain MPEG-7 features for most of the measures, apart from one-error, micro recall, example-based recall and macro AUC. The most important loses of the fusion process compared to the MPEG-7 features is in the precision related measures.

For the RAKEL approach, we notice that similarly to BR, the MPEG-7 features give better results than the BoW features across *all* evaluation measures. Again similarly to BR we notice that the fusion gives worse results compared to the BoW features in just three confidence related measures: micro AUC, coverage and ranking loss. The performance of the fusion process is similar to that of the MPEG-7 features. In specific, the MPEG-7 features give worse results compared to the fusion process in the example-based measures of hamming loss, accuracy, precision, F_1 and subset accuracy, the one-error ranking-based measure and the micro F_1 and micro precision label-based measures.

Having discussed the performance of the different feature vectors and their fusion for all algorithms we can argue about the following general conclusions. Firstly, the MPEG-7 features are much more valuable compared to the BoW features. We must note however, that the performance of BoW features might be affected by their low dimensionality (100), as other studies have used this kind of representation with success. Secondly, the fusion process leads to worse probability estimates for each label compared to the original features, probably due to the imprecise degrees of confidence output by the base-level models. Finally, the fusion process does not always improve the overall results compared to the MPEG-7 features.

4.3. Intuitions from Label-Specific Results

The performance of the 20 most common labels in the Mediamill Challenge dataset, using F-measure as a metric, for both BR and MLkNN is depicted in Figure 1. Although it is not a general rule, for labels that have a strong correlation, MLkNN seems to have a performance advantage over BR; "People" and "Face" are an example of labels that are strongly correlated and MLkNN outperforms BR in both. On the other hand, BR tends to outperform MLkNN in labels that seem uncorrelated with any other considered label, like "overlayed_text".

5. Conclusions and Future Work

This paper has performed an experimental study of several state-of-the-art methods for learning from multi-labeled video data. It has been shown that different methods perform well in different evaluation measures, and that the appropriate algorithm can be selected based on the application requirements and constraints.

One of the main issues worth of further investigation is how to improve the effectiveness of the fusion process, especially for BPMLL, MLkNN, RAkEL and other multi-label learning algorithms that don't treat each label independently as BR does. The fusion of different feature representations of multi-labeled objects has not been substantially studied in the literature, to the best of our knowledge, with the exception of [14]. For example, one problem with using Stacking as the fusion process is that the size of the feature space of the meta-level training data can easily grow significantly large, as it is equal to the product of the number of different representations and the number of labels.

Another interesting future research direction is the investigation of the ranking performance of the different methods for each specific label. This will allow a more careful assessment of the performance of methods for information retrieval tasks.

One of the issues neglected in the present paper is an account of the training and testing time of the different approaches. The optimization of SVMs via a grid-search process is highly time-consuming, while significant time is also required by the ensemble method RAkEL. BPMLL and MLkNN on the other hand are much more efficient with respect to the computational cost. A more detailed account of this subject is among our future plans.

References

[1] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[2] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[3] S. Yang, S.-K. Kim, and Y. M. Ro, "Semantic home photo categorization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 3, pp. 324–335, 2007.

[4] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*. New York, NY, USA: ACM, 2007, pp. 17–26.

[5] C. G. Snoek, M. Worring, J. C. van Gemert, J. Geusebroek, and A. W. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proc of ACM Multimedia*, Santa Barbara, USA, 2006, pp. 421–430.

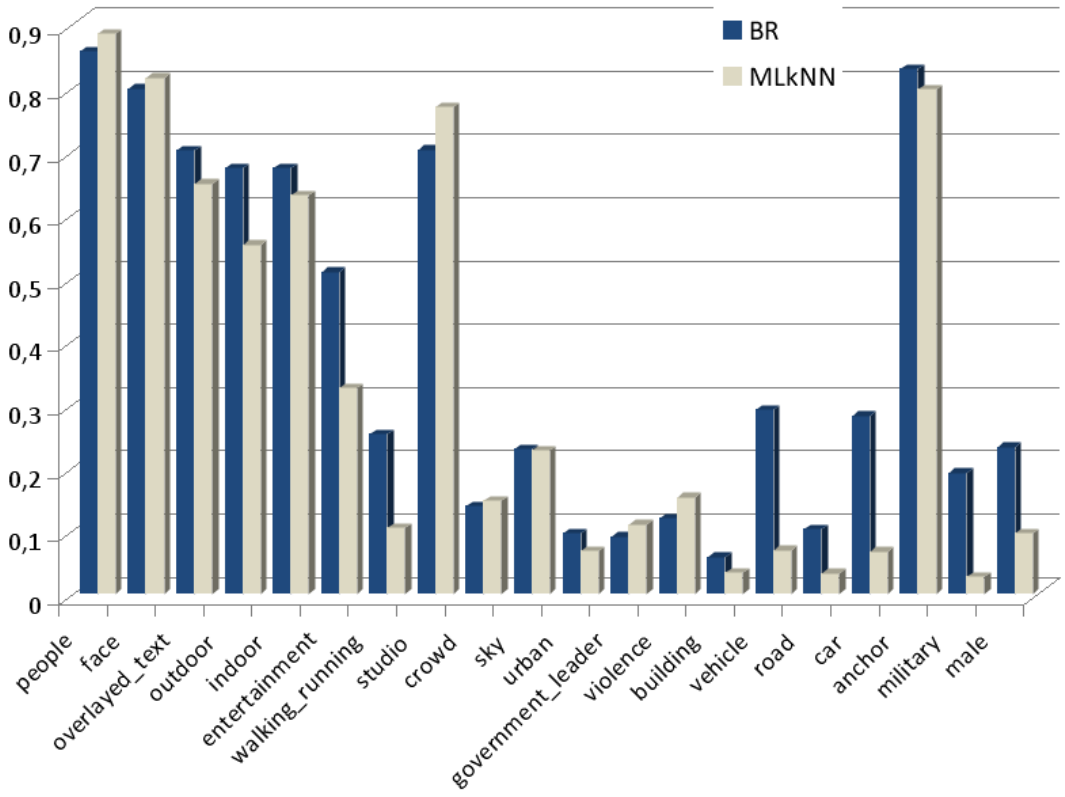

Figure 1. Classification results (F-measure) after the descriptor fusion stage, for the 20 most common labels of the dataset (ordered by the number of appearances in the dataset, in descending order)

[6] D. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[7] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[8] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Proc. of the 18th European Conference on Machine Learning (ECML 2007)*, Warsaw, Poland, September 17-21 2007, pp. 406–417.

[9] M.-L. Zhang and Z.-H. Zhou, "Multi-label neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–110, 2004, software available at http://www.cs.ubc.ca/ lowe/keypoints/.

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. of the 9th IEEE int Conf. on Computer Vision*, vol. 2. ICCV. IEEE Computer Society, Washington, DC, 1470, October 2003.

[12] C.-C. Chang and C.-J. Lin, "Libsvm : a library for support vector machines," 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[13] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[14] H.-P. Kriegel, P. Kröger, A. Pryakhin, and M. Schubert, "Using support vector machines for classifying large sets of multi-represented objects," in *Proc. 4th SIAM Int. Conf. on Data Mining*, 2004, pp. 102–114.

[15] G. Papadopoulos, V. Mezaris, S. Dasiopoulou, and I. Kompatsiaris, "Semantic image analysis using a learning approach and spatial context," in *Proc. First Int. Conf. on Semantics and Digital Media Technology (SAMT 2006)*, vol. 4306. Athens, Greece: Springer LNCS, December 2006, pp. 199–211.