

JOANNEUM RESEARCH at TRECVID 2013: Semantic Indexing and Instance Search

Werner Bailer, Harald Stiegler, Roland Mörzinger
JOANNEUM RESEARCH, DIGITAL – Institute for Information and Communication Technologies
8010 Graz, Austria
Email: {firstName.lastName@joanneum.at}

ABSTRACT

We participated in two tasks: semantic indexing (SIN) and instance search (INS).

SIN runs

We experimented with dynamic classifier selection, fusing input runs trained using SVM with RBF kernel on color and texture features. All runs were of type M, using parts of the IACC1 data for training, and IACC1.C as a validation set for fusion. Three runs use different methods for selecting the best classifier and determining the resulting score, the fourth uses a static classifier selection (per concept), also using IACC1.C as a validation set.

- JRS1: best in terms of AP, max. AP as score
- JRS2: best in terms of AP, max. score of all agreeing classifiers
- JRS3: best in terms of number of correct classification, max. score of all agreeing classifiers
- JRS4: static classifier selection based on performance on validation set, per concept

The fused results neither outperform the best of the input classifiers, nor the static fusion. We found that the main reason for this is that our input classifiers yield better results in terms of ranking than in terms of decision boundary, thus the runs achieve better score when their MAP is determined independently rather than when the binary classification are used to select a classifier in fusion.

INS runs

We submitted two runs for INS:

- JRS1: Matching SIFT descriptors using GPU, absolute number of matches
- JRS2: Matching SIFT descriptors using GPU, relative number of matches

The results of the two runs show no differences. For some of the queries, no results were found due to a low number of (reliable) interest points. For the other runs, verification was quite strict, leading to good precision for the top 10-20 ranks, but low recall.

I. SEMANTIC INDEXING

For the TRECVID 2013 [1] semantic indexing task (SIN), we have worked on dynamic fusion methods, i.e. data dependent methods, which do not choose an overall combination of

classifiers, but take the segment to be classified into account by selecting the (combination of) classifier(s) to be used. Keeping constraints from practical workflows in mind, we require that the method is able to treat the individual classifiers as black boxes (including the choice of features used by each of the classifiers), and that retraining of these classifiers is not required as part of the fusion process. Thus, only late fusion methods are to be considered.

The literature reports that data dependent classifier fusion makes weaker assumptions on independence of the individual classifiers, which makes the approach more flexibly applicable. Most of the literature deals with methods for fusing binary decisions. In the TRECVID setting we have ranked scores with different value ranges (the decision threshold may not always be the same), and we also need to generate a ranked list, i.e., we need to determine appropriate scores, not just correct classifications. The underlying assumption of dynamic classifier selection is that each classifier has a "region of expertise" in the feature space where it performs well. We based our method on the one proposed by [2] and select for each segment to be classified a set of similar segments from a validation set, for which ground truth annotations are available. The selection of classifiers and the weighting of the scores is based on the performance of the classifiers on this set. In our experiments, the set of similar items is determined as a superset of similar items based on the different visual features used in the classifiers. To determine the fused classification results, we implement the options based on different criteria and compared them,

A. Basic input runs

In order to generate input runs for fusion, we use a set of low-level features extracted from key frames and train a classifier for each concept using SVMs with RBF kernel. The following MPEG-7 [3], [4] image features were extracted globally:

Color Layout describes the spatial distribution of colors. This feature is computed by clustering the image into 8x8 blocks and deriving the average value for each block. After computation of DCT and encoding, a set of low frequency DCT components is selected (6 for the Y, 3 for the Cb and Cr plane).

Dominant Color consists of a small number of representative colors, the fraction of the image represented by each

color cluster and its variance. We use three dominant colors extracted by mean shift color clustering [5].

Color Structure captures both, color content and information about the spatial arrangement of the colors. Specifically, we compute a 32-bin histogram that counts the number of times a color is present in an 8×8 windowed neighborhood, as this window progresses over the image rows and columns.

EdgeHistogram represents the spatial distribution of five types of edges, namely four directional edges and one non-directional edge. We use a global histogram generated directly from the local edge histograms of 4×4 sub-images.

The input runs for fusion are all generated using the same algorithm, but a different subset of the training data. Table I provides an overview of the runs and their performance. One interesting observation on these runs is the decreasing performance for using the

B. Fusion

The fusion method is the same as used by the SIN submissions of the TOSCA-MP team. Details on the fusion method can be found in the notebook paper of the TOSCA-MP team [6]. For completeness, we describe the implemented approach here.

We have implemented a dynamic classifier selection method similar to DCS_LA proposed by [2]. We use the classifier performance of k items, which are determined based similarity in terms of visual low-level features.

The inputs for fusion are:

- Similarity matrix between samples in the test set and samples in the validation set
- Score from each classifier for each item in the validation set
- Score from each classifier for each item in the test set
- Ground truth for each item in the validation set

For each test item, the k nearest neighbors in a training or validation set are determined. Only items for which a ground truth annotation is available are considered, and a similarity value is stored for each item. In the TRECVID data sets, ground truth is available for different subsets of shots for each of the concepts. Thus the k -NN sets determined in this step differ per concept. How the similarities are determined is opaque to the fusion method, only similarity scores are needed. In our experiments we used the MPEG-7 ColorLayout, ColorStructure, DominantColor and EdgeHistogram descriptors [4]. We combine the four descriptors by selecting $k = 10$ samples based on each of the descriptors and forming the union of these sets. Apart from parameter k , a maximum threshold for the similarity values for considering an item part of the neighborhood has been determined. This value has been set to 0.01 in the experiments. In our implementation, only the inclusion the k -NN set is considered, but the actual similarity scores of the items in the neighborhood are not taken into account.

One issue is the scaling of the scores of the input classifiers. The following two rules have been implemented. If all scores are positive, they are shifted to have mean 0 and are scaled

to ± 1 . If scores are centered around 0, they are scaled to ± 1 . Note that this may imply different scaling factor positive and negative values. All the input classifiers used in the experiments fall into the second rule.

In the following we describe the different modes for selecting the classifier and determining the classification score.

1) *Best classifier in terms of number of correct classifications (Nbest)*: We count the number of correct classifications of each classifier, and select the one with the highest number of correct classifications. The score is determined as the highest score of all classifiers agreeing with selected one.

2) *Classifier with the highest confidence (Conf)*: Select the classifier with the highest confidence (absolute value) for the classifications. Again, the score is determined as the highest score of all classifiers agreeing with selected one.

3) *Best classifier in terms of fraction of correct classifications (Fbest)*: We count the number of correct classifications of each classifier, and select the one with the highest number of correct classifications (same as Nbest). The score is then determined from the fraction of correct classifications.

4) *Classifier with max. average precision (APbest)*: We determine the average precision for each classifier and we select the classifier with the best AP. The score is determined as the highest score of all classifiers agreeing with the best one. If there are no relevant documents in the k -NN set, we use mode *Nbest*.

5) *Classifier with max. average precision (APmax)*: We determine the average precision for each classifier and we select the classifier with the best AP. The score is determined as the average precision value of the best classifier. If there are no relevant documents in the k -NN set, we use mode *Nbest*.

There is a general fallback solution for all modes, if an item is not found in the k -NN set. We use a majority vote of classifiers, and use the highest score of the agreeing classifiers. If there is a tie we use the classification with the highest total score. However, this fallback solution has only been applied to a very small fraction of samples, so the impact on the overall performance is minimal.

The static fusion method used in the fourth runs determines the MAP for each classifier on the validation. For each concept the classifier with the highest MAP is selected and applied on the test set.

C. Results

We submitted four runs for this task, and performed five additional runs. The runs and their parameters as well as the MAP are shown in Table II.

As can be seen from the MAPs, we did not succeed in outperforming individual classifiers with the fused runs. Also the static fusion method used in JRS4 is not outperformed. The best of the implemented fusion methods is to take score of classifier with maximum number of correct classifications, and using the maximum of the agreeing classifier scores.

We observe some issues with all of the fusion methods. Selecting the best classifier in terms of the number of classifications has the risk of bias by the distribution in the k -NN set. We tried to address this with using the average

Run	description	MAP
JRS1dev	classifier trained on the IACC1 training set (TV2010 training)	0.036
JRS1A	classifier trained on the IACC1.A training set (TV2010 test)	0.013
JRS1B	classifier trained on the IACC1.B training set (TV2011 test)	0.009

TABLE I
PARAMETERS AND MAP OF THE RUNS USED AS INPUT FOR FUSION

Run	input runs	fusion method	validation set	MAP
JRS1	JRS1dev, JRS1A, JRS1B	APbest	IACC1.C	0.018
JRS2	JRS1dev, JRS1A, JRS1B	APmax	IACC1.C	0.009
JRS3	JRS1dev, JRS1A, JRS1B	Nbest	IACC1.C	0.025
JRS4	JRS1dev, JRS1A, JRS1B	static	IACC1.C	0.036
JRS5	JRS1dev, JRS1A, JRS1B	Fbest	IACC1.C	0.015
JRS11	JRS1dev, JRS1A, JRS1B	APbest	IACC1.A	0.003
JRS12	JRS1dev, JRS1A, JRS1B	APmax	IACC1.A	0.002
JRS13	JRS1dev, JRS1A, JRS1B	Nbest	IACC1.A	0.016
JRS15	JRS1dev, JRS1A, JRS1B	Fbest	IACC1.A	0.006

TABLE II
PARAMETERS AND MAP OF THE FUSED RUNS. RUNS 5-9 HAVE NOT BEEN OFFICIALLY SUBMITTED

precision, however, this approach runs into problems when there are no relevant items in the neighborhood. Using the input scores causes issues when the scaling of the scores of the input classifiers differ strongly. Using the fraction of correct classification has the nice property of taking the confidence into the output score. However, as the number of items in the neighborhood is rather small, this approach yields only a relatively small number of distinct values, thus making the ranking not very reliable.

We found that the main reason for the lower performance of the fused runs is the fact, that the ranking provided by the input classifiers used is much better than the actual decision boundary they report. In the standalone evaluation of the runs, the MAP is only calculated from the ranking. In contrast, for the dynamic classifier selection we make use of the binary classification output. There the best classifiers in terms on standalone MAP are often discarded, as they are outperformed by other classifiers. In addition, also the scores generated from these classifiers with lower performance are then used. As discussed above, using a criterion such as average precision for classifier selection does not solve this issue, as there are many neighborhood regions with a low number of relevant samples. A further discussion of fixing these issues can be found in [6].

The results of all the different fusion methods when using the training set of one of the classifiers (IACC1.A) for the validation for fusion are lower, but the relative differences between the fusion methods are similar.

D. Conclusion

We have attempted to use dynamic classifier selection for the TRECVID SIN task. As discussed above, the issue of dealing with ranked classifier outputs and missing or unreliable decision boundaries is not well covered in the existing literature. The approach we followed in our submissions was not able to solve the issues. Thus the fused classifier was outperformed by the best of the used input runs. There are still

a number of parameters in the approach (how to determine partitions, which features to use), for which we have made pragmatic decisions for the TRECVID SIN experiments and which should be further explored.

II. INSTANCE SEARCH

For TRECVID 2013 [1] instance search (INS), we implemented a system that does not perform any preprocessing, but extracts and matches SIFT descriptors extracted from DoG points at query time, using GPU acceleration.

The approach is based on extraction and matching of image areas around salient key points, using the SIFT (Scale Invariant Feature Transform) algorithm [7]. The SIFT algorithm has become very popular due to its powerful performance and is still used as a basic tool in the area of object recognition, near duplicate detection and other various related tasks. Although now almost fourteen years old, SIFT's average performance on object recognition is still state of the art, related algorithms tried to replace some parts of SIFT, e.g. different key point extraction methods (Hessian, Hessian-Affine, etc.), usage of other descriptors [8]–[10], or focused rather on reduction of computing power without or with only a small loss of recognition performance, but SIFT's recognition performance is still competitive.

Both extraction and matching of the descriptors has been implemented on GPU using NVIDIA CUDA¹ in order to speed up processing. Only one field of the input video is used in order to avoid possible side effects of interlaced content. Descriptors are extracted from every 5th frame of the video.

The SIFT algorithm is described in detail in [7], which describes the localization, extraction and matching of key points and its descriptors. The SIFT implementation used in our work follows [7], but nearest neighbor search of key points and their descriptors is implemented differently. Instead of using best bin first (BBF) for nearest neighbor search, a brute

¹http://www.nvidia.com/object/cuda_home_new.html

force matching on a GPU is performed, which is still faster and easier to implement than a CPU based nearest neighbor search based on kd-trees recommended in [7]. Matching verification by usage of an homography still follows [7] again. The process of key point matching results in the set of key points, which support a meaningful homography from one frame to the other. This number is used as a score and input to a two dimensional matrix, where each axis denotes the timeline of a video segment. If key points of frame f_i^A (originating from video A) are matched with key points of frame f_j^B (originating from video B), the number of matching key points is the matrix's value at row f_i^A , column f_j^B . Peaks in the matrix indicate very similar frames, and sequences of matching frames indicate reliable matches.

A. Results

We have submitted two runs using the method described above. The runs differ only by the ranking criterion: JRS1 uses the absolute number of matching interest points, and JRS2 uses the relative number. However, we found no significant differences between the results of the two runs.

For some of the queries, no results were found due to a low number of (reliable) interest points. For the other queries, verification was quite strict, leading to good precision for the top 10-20 ranks, but low recall. Figure 1 shows the results for the queries, for which results were returned. The lower performance for person type runs (including the statue) is expected, as the approach used will match all person rather than discriminating specific individuals.

In an analysis of the results we found that discarding one of the fields was responsible for the low number of matches in several of the cases. Due to the size reducing, the number of key points decreased significantly. We tried to keep the both fields in case that no interlacing artifacts were present, which significantly increased the number of returned items.

B. Conclusion

As could be expected, an interest point based approach will fail on small and hardly textured objects. Also, typical gradient based descriptors are not able to discriminate between persons. For the queries where results were found, the performance is good at the top ranks, leading to high recall for these queries. However, due to several spatial and temporal validation steps, few results beyond rank 25 have been submitting, resulting in low recall and thus overall low MAP.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Seventh Framework Programme under grant agreement n° 287532, "TOSCA-MP – Task-oriented search and content annotation for media production" (<http://www.tosca-mp.eu>), and by the project grant "QuOIMA" under Austrian National Security Research Development Programme KIRAS.

REFERENCES

- [1] 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.
- [2] K. S. Woods, W. P. Kegelmeyer, and K. W. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 405–410, 1997.
- [3] "Information technology-multimedia content description interface: Part 3: Visual," ISO/IEC 15938-3, 2001.
- [4] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada, "Color and texture descriptors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 703–715, Jun. 2001.
- [5] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
- [6] W. Bailer, S. Gerke, A. Linnemann, and P. Ndjiki-Nya, "TOSCA-MP at TRECVID 2013: Semantic Indexing," in *Proceedings of TRECVID Workshop*, Nov. 2013.
- [7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [8] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005. [Online]. Available: <http://lear.inrialpes.fr/pubs/2005/MS05>
- [9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.cviu.2007.09.014>
- [10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893.

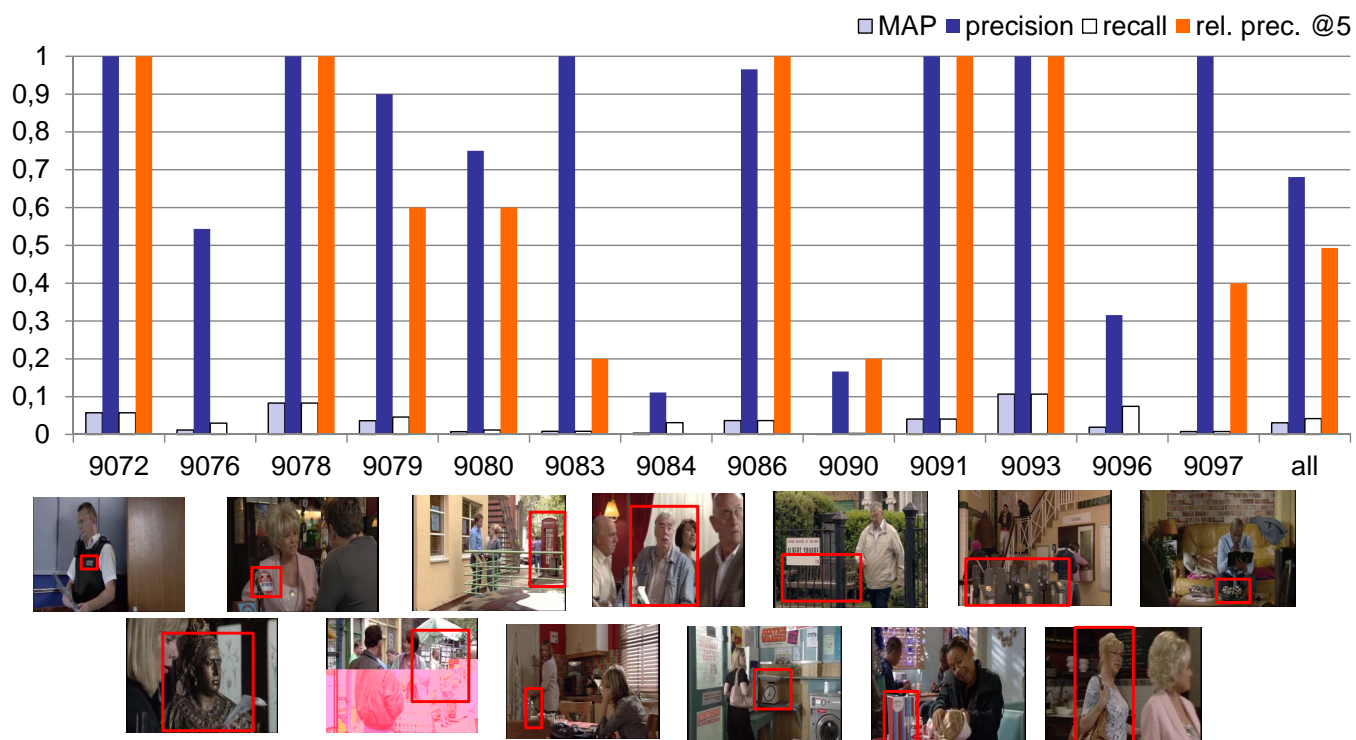


Fig. 1. Precision at top ranks and MAP for queries, for which results were returned.