# TOSCA-MP and HHI at TRECVID 2013: Semantic Indexing

Werner Bailer*, Sebastian Gerke†, Antje Linnemann†, Patrick Ndjiki-Nya†
*JOANNEUM RESEARCH, DIGITAL – Institute for Information and Communication Technologies
8010 Graz, Austria
Email: {firstName.lastName@joanneum.at}
†Fraunhofer Heinrich Hertz Institute
10587 Berlin, Germany
Email: {firstName.lastName@hhi.fraunhofer.de}

## Abstract

We participated in the semantic indexing task, and submitted the following runs. We experimented with dynamic classifier selection, using runs from the HHI and JRS teams as input. All runs runs were of type M, using parts of the IACC1 data for training, and IACC1.C as a validation set for fusion. The four runs use different methods for selecting the best classifier and determining the resulting score, 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.

- TOSCA1: best in terms of AP, max. score of all agreeing classifiers
- TOSCA2: best in terms of number of correct classification, max. score of all agreeing classifiers
- TOSCA3: best in terms of AP, max. AP as score
- TOSCA4: same as TOSCA2, with slightly updated input runs (did not change the fused result)

The fused result does not outperform the best of the input classifiers. 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.

## I. Introduction

For the TRECVID 2013 [1] semantic indexing (SIN) task, 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 is performs well. Our approach is based on the method 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 implemented the options based on different criteria and compared then,

The rest of this paper is organized as follows. As we focused on fusion, Section II discusses the input runs used in the experiments. Section III presents the details of the fusion method. In Section IV we report the results, and Section V concludes the paper.

## II. Overview of Input Runs

This section provides a very brief overview of the input runs used for fusion. Table I provides an overview of the runs and their performance.

The main contribution of the HHI1 is the use of bag-of-word histograms obtained from SIFT features on the optical flow field of a frame. Therefore, optical flow is determined by using Farneback’s method as proposed in [3]. On the optical flow field, densely sampled SIFT features are extracted. Instead of calculating histogram of flow, histogram of gradients are chosen in order to remove the influence of global motion. In the remainder of this paper, we refer to this feature as MotionSIFT. The bag-of-words histogram of MotionSIFT features is then concatenated with the bag-of-words histogram of the OpponentSIFT features.

The HHI2 run is based on a concept-specific bag-of-words vocabulary obtained from densely sampled OpponentSIFT features. Concept-specific means that a different vocabulary for each concept is obtained by clustering only positively labeled frames for that concept. For both HHI runs, a $\chi^2$ kernel based SVM has been used to train and classify the concepts.

The JRS runs are trained on visual color and texture features of key frames, using an SVM with RBF kernel as classifier. The JRS input runs have not been submitted as official TRECVID runs. More details on the JRS runs can be found in the TRECVID notebook paper of the JRS [4] team.

## III. Fusion Approach

In the following, we discuss literature used to design the fusion method and describe the method that has been implemented. The basic assumption of all these methods is that it is not optimal to select a classifier (or fixed set of weights of classifiers) at training time. Instead, classifiers are assumed to have an "area of expertise", i.e., subspaces of the feature space, for which a particular classifier is expected to work well. This information should be taken into account when classifying a specific sample from the test set. The methods use either the training set or a separate validation set for selection/weighting of the different classifiers.

### A. Related work

Huang et al. [5] have proposed a method called behavior-knowledge space (BKS), which uses a space of dimensionality corresponding to the classifiers. The values of data points in the space correspond to one of the classes of a rejection label. The threshold for rejection can be learned during training. The methods cannot handle confidences of the individual classifiers.

Hierarchical mixture of experts (HME) has been proposed in [6]. The method is explicitly data dependent as gating values are determined, which are non-linearly dependent on the sample to be classified. To learn the parameters of the HME, a variant of the EM algorithm is used. The direct HME formulation is defined linear wrt. the input vectors, thus it cannot be applied to kernel-based classifiers using non-linear kernels.

A method called rank combination [7] partitions the feature space based on agreement of classifiers on the top ranks. Then a regression model for classifier combination is trained for each partition.

DS/DW [8] selects based on similar response pattern of classifiers, and calculates local performance from similar responses. The model is treated as a black box, but is designed to work with a larger number of input classifiers.

Dynamic classifier selection by local accuracy (DCS_LA) has been proposed by [2]. For the samples to be classified, partitions of the feature space are selected using $k$ nearest neighbors ($k$-NN). For each of these partitions, the best classifier is selected counting the number of correct and false classification. This classifier is then used for samples falling into the respective partition of the features space. Some extensions of these methods have been proposed. DCS_MCB tries to determine the optimal value for $k$ automatically based on similarities in the training data. The authors of [9] propose a method called AO-DCS that also addresses the selection of $k$, some performance improvements and reduce the sensitivity to noise. The main idea is to use a fixed partitioning of data set by "important" attribute values (somewhat comparable to classification trees) and then select best classifier for each partition.

Classifier local accuracy (CLA) has been proposed in [10], and can be used in an "a priori" (without using the class predictions of the individual classifiers) or "a posteriori" way (using them). The method supports adaptive distance metrics and dynamic choice of $k$, but is not directly applicable if the classifiers are not trained on the same features.

GAEGA [11] is related to BKS, as it tests in local regions based on the input classifiers' decision boundaries', which correspond to cells in BKS. A data point can be evaluated in multiple regions and different transform feature spaces are used to determine $k$-NNs. It needs base classifier that provides an explicit decision boundary.

### B. Consideration for use in TRECVID SIN

Most of the literature deals with methods for fusing binary decisions. In our application we have ranked scores with different value ranges, and some classifiers might not even return the decision boundary. For the output, we also need to generate a ranked list, i.e., we need to determine appropriate scores, not just binary classifications. Calculating these scores is a critical issue, especially if the input classifiers return differently (non-linearly) scaled scores.

Correlation of the classifiers is an issue. [12] and [13] report that the error of the fusion step decreases as the correlation of the base classifiers approaches 0, however, authors of dynamic fusion methods reports that dynamic classifier selection methods make less strong assumptions on independence of the individual classifiers.

Most existing dynamic classifier selection approaches use exactly the same features for fusion as in the classifiers involved. However, this is not feasible if we do not have complete control over the classifiers and/or have these original features available. Also, this would allow the application of the approach only in cases where all classifiers are trained on the same set of features, which would render the method unusable for fusing e.g. visual and audio classifiers.

It seems useful to use a separate validation set for the classifier selection than the training set. The drawback is of course additional computational complexity, as e.g. $k$-NN calculation needs also similarities between the test set and the validation set. Only if one assumes that the original classifier generalizes sufficiently well, one would expect to have very similar results when using the training set for $k$-NN generation as when using a different validation set. This of course makes strong assumptions about the representativeness of the training set for the test set.

The similarity calculation requires some notion of similarity which can be based on low-level features, but also other features such as text, genre, creator, etc. could be used. As in the existing literature almost always the same set of features

| 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 |
| HHI1 | MotionSIFT + OpponentSIFT | 0.024 |
| HHI2 | OpponentSIFT, concept-specific codebooks | 0.108 |
| HHI2a | small update to run HHI2 (no impact on results observed) | 0.108 |

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

is used for all the classifiers, the same features are typically also used for similarity calculation.

A nice property of dynamic classifier selection methods is that they could reject a decision, if the classifiers for a particular kind of samples seem to be unreliable, or at least return a confidence. In the TRECVID setting, rejecting a decision cannot be expressed, so the only option is to use this confidence as one of the inputs for the fused classification score.

The considered methods might still be useful for combining classifiers trained on different data sets, similar to bagging or multi-instance learning approaches. This may also be used to adapt a set of classifiers to additional data sets.

### C. Implementation

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 [14]. 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 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 $\pm 1$. If scores are centered around 0, scale to $\pm 1$. Note that this may imply different scaling factor positive and negative values. All the input classifiers used in the experiments fell 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.

### D. Improved implementation

We implemented a modified version of the *APbest* mode. One challenge is the scaling of the scores of the input classifiers. Figure 1 shows examples of order scores of two different classifiers on the same data. It is evident, that we cannot easily combine the scores, even if we perform typical

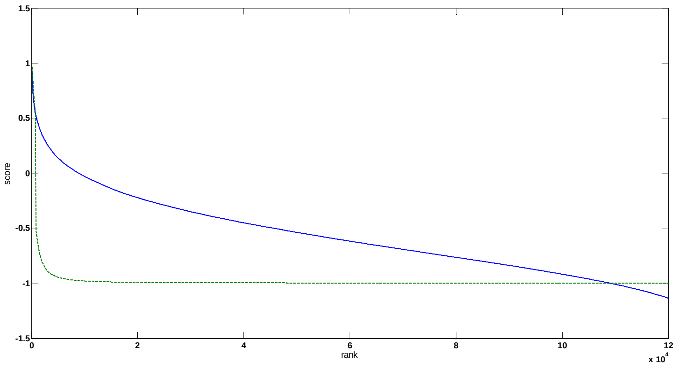

Fig. 1. Ranked scores of two classifiers to be fused.

normalization (e.g., scaling to zero mean and variance of one). In order to deal with this problem, we found that it is better to rely neither on the binary decisions nor on the actual scores of the classifiers, but only on the ranked list of classifications. We thus replace the scores with a rank, supporting also ties in case of identical input scores. The ranks are then normalised to the range $[0;1]$. In order to determine the performance of the classifiers on the validation set we use average precision ($ap$), as this allows us to compare only the rankings resulting from the classifiers, without the need to make hard decisions about correct or incorrect classifications. It turns out that the neighborhood set contains very few relevant samples for many of the test samples. We thus determine also the number of relevant items in a neighbourhood set without relevant items (i.e., the false positive rate $fp$).

The output score is determined by softly switching between the maximum score and a weighted combination of all input scores by the relative performance on the neighbourhood set. Switching is based on the absolute sum of disagreements between the classifiers. The disagreement score is normalized to 1, and reweighted using a sigmoid function, yielding $agree_w$. The maximum score of classifiers is then determined as

$$\begin{aligned} score_{out} = \quad & (1 - agree_w)\max_i(score_{in}(i)) + agree_w \\ & (w(i)) \,, \end{aligned} \tag{1}$$

with

$$w() = \quad \begin{aligned} & \frac{\sum_i ap(i) score_{in}(i)}{\sum_i ap(i)}, \text{if } nrel > 0 \\ & \frac{\sum_i fp(i) score_{in}(i)}{\sum_i fp(i)} \text{otherwise,} \end{aligned} \tag{2}$$

where $nrel$ is the number of relevant items in the neighbourhood set.

## IV. Results

### A. Official runs

We submitted four runs for this task. The runs and their parameters as well as the MAP are shown in Table II. Further fusion experiments involving only the JRS runs are reported in the JRS notebook paper [4].

As can be seen from the MAPs, we did not succeed in outperforming individual classifiers with the fused runs. 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 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.

The HHI2 run is much better than other input runs used. However, adding this run did not contribute to a significant improvement of the performance of the fused run. We found that the main reason for this issue 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. 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.

### B. Further experiments

We have used the improved algorithm for further experiments. Figure 2 shows the results after modification of the fusion method. While the overall mAP is still slightly worse than the selection of the best classifier per concept, the fusion method outperforms the best input classifier for about half of the concepts.

## V. 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 for half of the concepts. 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.

| Run | input runs | fusion method | MAP |
|---|---|---|---|
| TOSCA1 | JRS1dev, JRS1A, JRS1B, HHI1, HHI2 | APbest | 0.022 |
| TOSCA2 | JRS1dev, JRS1A, JRS1B, HHI1, HHI2 | APmax | 0.011 |
| TOSCA3 | JRS1dev, JRS1A, JRS1B, HHI1, HHI2 | Nbest | 0.027 |
| TOSCA4 | JRS1dev, JRS1A, JRS1B, HHI1, HHI2a | APmax | 0.011 |

TABLE II
PARAMETERS AND MAP OF THE FUSED RUNS

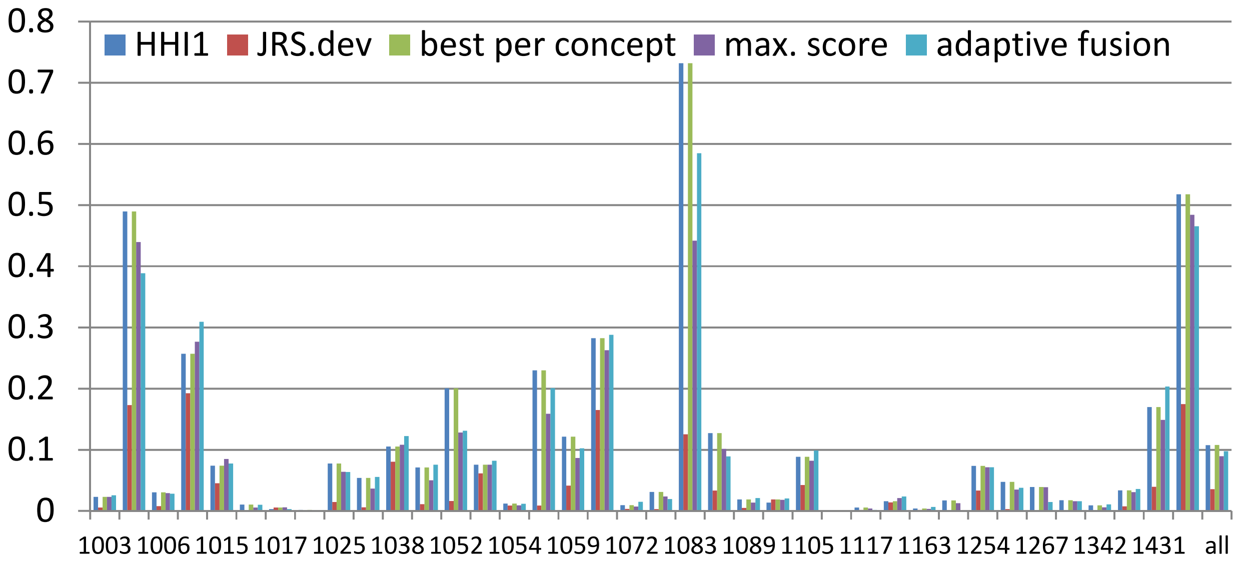

Fig. 2. Performance of input and fused runs using the improved fusion algorithm.

## ACKNOWLEDGMENTS

The authors would like to thank Roland Mörzinger, Harald Stiegler and Georg Thallinger for their feedback and support.

The research leading to these results has received funding from the European Union's Seventh Framework Programme under grant agreements n° 287532, "TOSCA-MP - Task-oriented search and content annotation for media production" (http://www.tosca-mp.eu).

## 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] G. Farneback, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, ser. Lecture Notes in Computer Science, J. Bigun and T. Gustavsson, Eds. Springer Berlin Heidelberg, 2003, vol. 2749, pp. 363–370. [Online]. Available: http://dx.doi.org/10.1007/3-540-45103-X_50

[4] W. Bailer, H. Stiegler, and R. Mörzinger, "JOANNEUM RESEARCH at TRECVID 2013: Semantic Indexing and Instance Search," in *Proceedings of TRECVID Workshop*, Nov. 2013.

[5] Y. S. Huang and C. Y. Suen, "The behavior-knowledge space method for combination of multiple classifiers," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 1993.

[6] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural Comput.*, vol. 6, no. 2, pp. 181–214, Mar. 1994. [Online]. Available: http://dx.doi.org/10.1162/neco.1994.6.2.181

[7] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 1, pp. 66–75, Jan. 1994. [Online]. Available: http://dx.doi.org/10.1109/34.273716

[8] C. J. Merz, *Dynamical selection of learning algorithms*. New York: Springer, 1996, pp. 281–290.

[9] X. Zhu, X. Wu, and Y. Yang, "Dynamic classifier selection for effective mining from noisy data streams," in *Fourth IEEE International Conference on Data Mining*, 2004.

[10] L. Didaci and G. Giacinto, *Dynamic classifier selection by adaptive k-nearest-neighbourhood rule*. Berlin Heidelberg: Springer, 2004, pp. 174–183.

[11] L. Chen and M. S. Kamel, "A generalized adaptive ensemble generation and aggregation approach for multiple classifier systems," *Pattern Recogn.*, vol. 42, no. 5, pp. 629–644, May 2009.

[12] N. M. Wanas and M. S. Kamel, "Weighted combination of neural network ensembles," in *International Joint Conference on Neural Networks*, 2002.

[13] K. Tumer and J. Ghosh, "Estimating the bayes error rate through classifier combining," in *Proceedings of the Thirteenth International Conference on Pattern Recognition*, Vienna, Austria, August 1996, pp. IV:695–99.

[14] 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.