

PicSOM Experiments in ImageCLEF RobotVision*

Mats Sjöberg, Markus Koskela, Ville Viitaniemi, and Jorma Laaksonen

Adaptive Informatics Research Centre
Aalto University School of Science and Technology
P.O. Box 15400, FI-00076 Aalto, Finland
`firstname.lastname@tkk.fi`

Abstract. The PicSOM multimedia analysis and retrieval system has previously been successfully applied to supervised concept detection in image and video databases. Such concepts include locations and events and objects of a particular type. In this paper we apply the general-purpose visual category recognition algorithm in PicSOM to the recognition of indoor locations in the ImageCLEF/ICPR RobotVision 2010 contest. The algorithm uses bag-of-visual-words and other visual features with fusion of SVM classifiers. The results show that given a large enough training set, a purely appearance-based method can perform very well – ranked first for one of the contest’s training sets.

1 Introduction

In this paper we describe the application of our general-purpose content-based image and video retrieval system PicSOM [4] to the ImageCLEF/ICPR 2010 RobotVision contest task. Among other things, the PicSOM system implements a general visual category recognition algorithm using bag-of-visual-words and other low-level features together with fusion of SVM classifiers. This setup has been used successfully previously, for example in the NIST TRECVID 2008 and 2009 [10] high-level feature detection tasks [11], where events, locations and objects are detected in television broadcast videos. Our goal in the experiments described in this paper is to evaluate the suitability of this general-purpose visual category detection method to a more narrow domain in the indoor location detection setup of the RobotVision contest. We have not included any domain specific features in these experiments, such as depth information from stereo imaging. Thus, the only modality we consider is the current view from one or more forward-pointing cameras.

In addition to autonomous robots [8], a RobotVision-style setup arises, for example, in many applications of mobile augmented reality [2]. In fact, indoor localisation constitutes also one of the sub-tasks of our research platform for accessing abstract information in real-world pervasive computing environments through augmented reality displays [1]. In that context, objects, people, and

* This work has been supported by the Aalto University MIDE project UI-ART.

the environment serve as contextual channels to more information, and adaptive models are used to infer from implicit and explicit feedback signals the interests of users with respect to the environment. Results of proactive context-sensitive information retrieval are augmented onto the view of data glasses or other see-through displays.

Fig. 1 illustrates our visual category recognition approach. Given training images with location labels, we first train a separate detector for each location L_i . Section 2 describes these single-location detectors that employ fusion of several SVM detectors, each based on a different visual feature. The probabilistic outcomes of the detectors are then used as inputs to the multi-class classification step that determines the final location label \hat{L} for each test image. This step is described in Section 3. The predicted \hat{L} is either one of the known locations L_i , or alternatively, the system can predict that the image is taken in a novel unknown location, or declare the location to be uncertain. In Section 4 we describe our experiments in RobotVision 2010 and summarise the results. Finally, conclusions are drawn in Section 5.

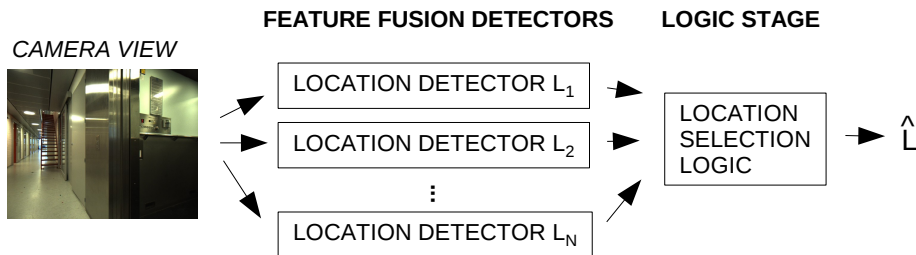


Fig. 1. General architecture for predicting location \hat{L} based on a camera view

2 Single-Location Detectors

For detecting a single location L_i , our system employs the architecture illustrated in Fig. 2. The training phase begins with the extraction of a large set of low-level visual features. The features and binary location labels of the training images (L_i or non- L_i) are then used to train a set of probabilistic two-class SVM classifiers. A separate SVM is trained for each visual feature.

After training, the detector can estimate the probability of a novel test image depicting location L_i . This is achieved by first extracting the same set of visual features from the test image that was extracted from the training images. The trained feature-wise SVM detectors produce a set of probability estimates that are combined to a final probability estimate in a fusion stage. The location-wise estimates are then combined in a multi-class classification stage to determine the location of the test image.

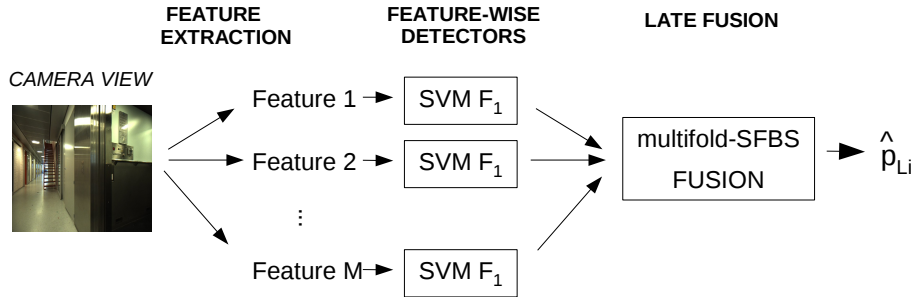


Fig. 2. Architecture for estimating the probability \hat{p}_{Li} that the given camera view is from location L_i

2.1 Feature Extraction

From each image, a set of low-level visual features is extracted. We use our own implementations of the following MPEG-7 descriptors: *Color Layout*, *Dominant Color*, *Scalable Color*, and *Edge Histogram* [3]. Additionally, we calculate several non-standard low-level appearance features: *Average Color*, *Color Moments*, *Texture Neighbourhood*, *Edge Histogram*, *Edge Co-occurrence* and *Edge Fourier*. The non-standard features are calculated for five spatial zones of each image (Figure 3) and the values concatenated to one image-wise vector.

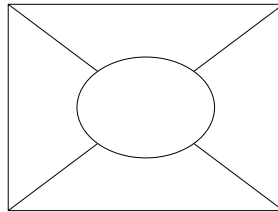


Fig. 3. The five-part center-surround zoning mask for image feature extraction

Of the non-standard features, the Average Color feature is a three-element vector that contains the average RGB values of all the pixels within the zone. The Color Moments feature treats the HSV colour channels from the zone as probability distributions, and calculates the first three central moments (mean, variance and skewness) of each distribution. For the Texture Neighbourhood feature, relative values of the Y (luminance) component of the YIQ colour representation in all 8-neighbourhoods within the zone are characterised. The probabilities for neighbouring pixels being more luminous than the central pixel are estimated separately for all the eight surrounding relative pixel positions, and collected as a feature vector. Edge Histogram is the histogram of four Sobel edge

directions. The feature differs in details from the similarly named MPEG-7 descriptor. Edge Co-occurrence gives the co-occurrence matrix of four Sobel edge directions.

Furthermore, eight different bag-of-visual-words (BoV) features are also extracted. In the BoV model images are represented by histograms of local image descriptors. The eight features result from combining number of independent design choices. First, we use either the *SIFT* [6] or the opponent colour space version of the *Color SIFT* descriptor [9]. Second, we employ either the Harris-Laplace detector [7] or use dense sampling of image locations as the interest point detector. Third, we have the option to use the soft-histogram refinement of the BoV codebooks [9]. Finally, for some of the features, we have used the spatial pyramid extension of the BoV model [5].

2.2 Feature-Wise Detectors

In our location recognition system, the association between an image's visual features and its location is learned using the SVM supervised learning algorithm. The SVM implementation we use in our system is an adaptation of the C-SVC classifier of the LIBSVM¹ software library. For all histogram-like visual features we employ the χ^2 kernel

$$g_{\chi^2}(\mathbf{x}, \mathbf{x}') = \exp \left(-\gamma \sum_{i=1}^d \frac{(x_i - x'_i)^2}{x_i + x'_i} \right). \quad (1)$$

The radial basis function (RBF) SVM kernel

$$g_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2) \quad (2)$$

is used for all the other features. The motivation for this is the well-known empirical observation that χ^2 distance is well-suited for comparing histograms.

The free parameters of the SVMs are selected with an approximate 10-fold cross-validation search procedure that consists of a heuristic line search to identify a promising parameter region, followed by a grid search in that region. To speed up the computation, the data set is radically downsampled for the parameter search phase. Further speed-up is gained by optimising the C-SVC cost function only very approximately during the search.

For the final detectors we also downsample the data set, but less radically than in the parameter search phase. Usually there are much fewer annotated example shots of a location (positive examples) than there are example shots not exhibiting that location (negative examples). Consequently, for most of the locations, the sampling is able to retain all the positive examples and just limit the number of negative examples. The exact amount of applied sampling varies according to the computation resources available and the required accuracy of the outputs. Generally we have observed the downsampling to degrade detection accuracy.

¹ <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

2.3 Fusion

The supervised fusion stage of our location recognition system is based on the geometric mean of feature-wise detector outcomes. However, instead of calculating the mean of all feature-wise detectors we select the set using sequential forward-backward search (SFBS). This supervised variable selection technique requires detector outcomes also for training images. These outcomes are obtained via 10-fold cross-validation.

Our search technique refines the basic SFBS approach by partitioning the training set into multiple folds. In our implementation we have used a fixed number of six folds. The SFBS algorithm is run several times, each time leaving one fold outside the training set. The final fusion outcome is the geometric mean of the fold-wise geometric means.

3 Multi-class Classification

The fusion of the feature-wise detector scores described in the previous section provides probability estimates for each location given a particular image. The final classification step is a traditional multi-class classification, where we combine several one-versus-the-rest SVM classifiers. The straightforward solution is to classify the image to the class with the highest probability estimate. However, in the current scenario, we must also be able to detect *unknown* categories, i.e. images of new locations that have not been seen before. We have implemented this by a heuristic method with two thresholds. First, if there are detector scores above a high threshold T_1 , then we deem the system to be confident enough to simply pick the class with highest score. Second, if all scores are below a low threshold T_2 , this is interpreted to mean that we are seeing an unknown class for which we have not trained a detector. Finally, if none of the above conditions apply, there must be one or more scores that are above T_2 , which can be seen as potential detections.

These scores of such potential detections are all smaller than T_1 (since the first condition was not true), and can be seen as potential, but not strong detections. Trivially, if the number of such scores equals one, this one is selected as the detected class. If the number of such detection scores is higher than two we deem the situation to be too uncertain and decline to classify it (i.e. it is the *reject* class). If the number equals two we select the highest one. This is due to the particular performance measure used in RobotVision, which rewards a correct choice with +1.0 and an incorrect choice with -0.5. This means that if the correct class is either of the two potential candidates the expectation value of the performance measure score is still positive even by selecting either at random.

4 Experiments

4.1 Recognition with Stereo Images

In the RobotVision setup, the presence of a stereo image pair demands some additional considerations. For example, we might learn a separate model for

each camera, i.e. independent models for the images of left camera and right camera. On the other hand, since the two cameras show the same scene from somewhat different angles, they are certainly not independent. In fact, if we consider the set of images taken from each point in every possible angle, the left and right images are just two samples from the same distribution. I.e. the image seen in the left camera might be seen in the right camera at some other point in time if the robot happens to be at the same point in space but at a slightly different angle. This view would then support the approach of simply using all images as training data discarding the left/right distinction. In the end we made two models, one using all images, and one using only the images from the left camera (i.e. only half of the data). After the competition we also made a model from the right camera images for comparison.

In the final stage, when categorising stereo images at particular times, one must have a strategy for combining the detections scores for the two cameras. We tried taking their average, maximum, minimum, or only taking the left camera, or only right camera result. The stereo image pair could also be utilised for stereo imaging and the contest organisers provided the camera calibration data for the image sequences. Depth information would undoubtedly be an useful feature for location recognition. We have, however, not utilised such domain specific features in the present work.

4.2 Parameter Selection

The combining logic described in Section 3 was used, and the two thresholds, T_1 and T_2 were determined by simple grid search maximising the performance score in the validation set. Because the testing set had four unseen rooms, we tried to simulate this situation in the training by leaving out three rooms (roughly the same ratio of unseen to seen) and use this setup when determining the thresholds. We did this both with the regular detectors trained on the full set of rooms, and with detectors trained on the reduced set. Those trained on the full set we thought might be unrealistic since they had used the three removed rooms as negative examples in their training. Using detectors trained on the reduced set tended to increase the lower threshold T_2 from the level it had when using detectors trained on all known rooms. It turned out however that the lower thresholds worked better in the testing set.

The threshold T_1 , which gives a limit for when to decline from classification (*reject*) did not affect the results significantly, and is not included in the results presented in the following.

4.3 Results

Our best submitted result for the easy set received a score of 2176.0, which is 85% of the best possible score. This result was based on detectors trained on the left camera data only, and it obtained the overall highest score in the competition for the easy set. The same setup achieved our best result (1117.0) for the hard set as well. Fig. 4 visualises this run compared to the groundtruth. For the hard set, our result was slightly above the median of the submitted results using the hard training data. The overall best submitted run to the hard set was 1777.0.

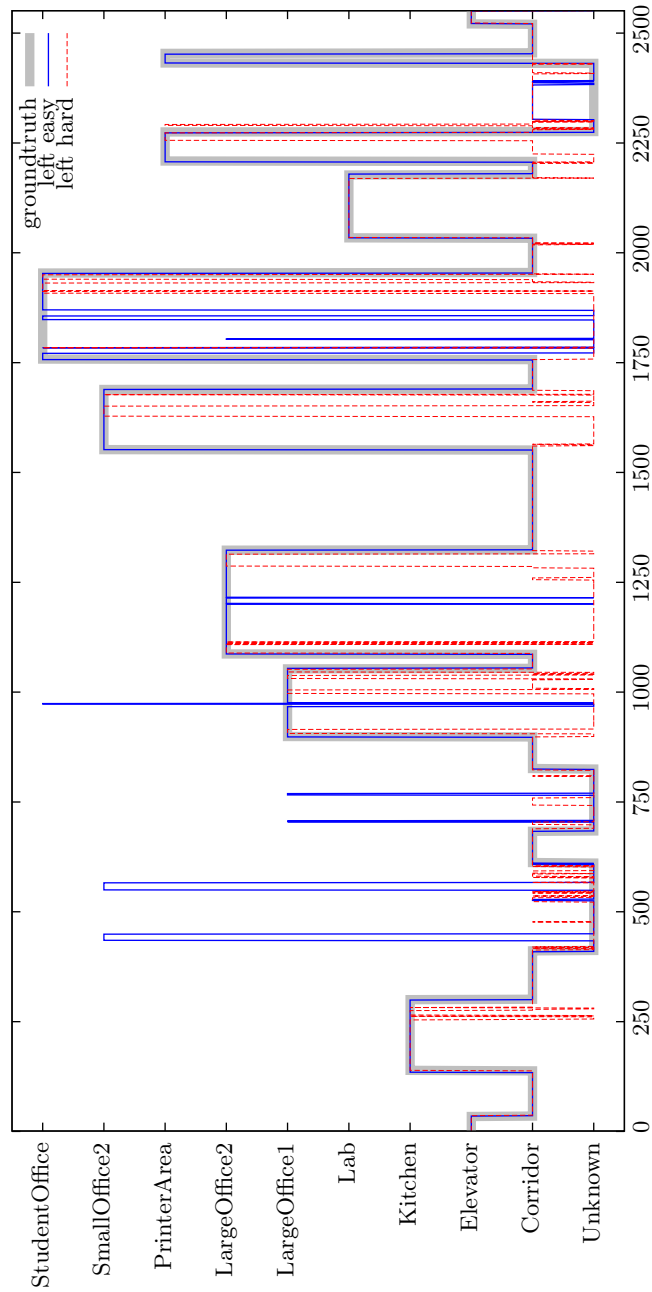


Fig. 4. Recognition results over time (frame indices) based on left camera images, trained on easy (blue line) and hard (red dashed) sets. The groundtruth is shown as a thick grey line.

These and some additional runs are summarised in Table 1, with “•” denoting that the run was submitted to the competition. The first column in the table specifies how the training data was selected with regard to the cameras. The word “separate” indicates that separate models were trained for each camera and then averaged, while “both” uses all images to train a single model. The camera-wise scores were combined by taking the average value when using datasets with images from both cameras. The second column states whether fusion of single-feature classifiers (Section 2.3) or just the single best performing feature (ColorSIFT with dense sampling over a spatial pyramid and soft clustering) is used.

Somewhat surprisingly, using information from both cameras does not improve the results, in fact using a single camera works better than using a single model trained on all images. This difference is especially notable on the hard training set. Also, using the separate left and right models together gives no improvement over using just one of them. Finally, in Table 1 we can also see that the feature fusion is highly beneficial: with a single feature the results are significantly weaker.

Table 1. RobotVision recognition scores

cameras	features	easy	hard	total
• left only	fusion	2176.0	1117.0	3293.0
right only	fusion	2210.5	1072.0	3282.5
separate	fusion	2207.5	1057.0	3264.5
• both	fusion	2065.0	665.5	2730.5
• both	single	964.0	554.5	1518.5

After the competition, each participant was given access to the labels for the testing dataset, and we were able to perform some more tests, and determine optimal parameters T_1 and T_2 in the testing set. These additional tests are summarised in Table 2 using fusion of single-feature classifiers. The first column specifies how the training data was selected with regards to the cameras: using images from both together, or just using the images from the left camera. The second column shows how the detection scores from the two cameras were combined to form the final detection score: by taking the average of the left and the right, by taking the maximum or minimum, or by simply taking the left or the right camera scores directly. It can clearly be seen that using different ways of combining the stereo-vision scores makes very little difference, the choice of training data is much more important. Even using a model trained on the left camera images for the right camera is better than using the dataset with images from both cameras with any score combination method.

Note that the results shown in Table 2 are not comparable with other competition submissions since they have been optimised against the testing set, and are thus “oracle” results. They are however interesting for a comparison between

Table 2. “Oracle” detection scores

cameras	selection	easy	hard	total
both	average	2126.5	910.0	3036.5
left	average	2176.0	1144.5	3320.5
both	left	2090.5	908.5	2999.0
left	left	2174.5	1160.5	3335.0
both	max	2114.5	912.5	3027.0
left	max	2179.0	1148.5	3327.5
both	min	2099.5	901.5	3001.0
left	min	2164.0	1157.5	3321.5
both	right	2116.0	905.0	3021.0
left	right	2152.0	1135.0	3287.0

different alternations of our method. Furthermore, it can be observed that the oracle results are only slightly better than the submitted ones, indicating that the system performance is not very sensitive to the threshold parameters.

5 Conclusions

Our results indicate that a general-purpose algorithm for visual category recognition can perform well in indoor location recognition, given that enough training data is available. The generality of our approach is illustrated e.g. by its successful application to image and video retrieval [10]. With limited training data, however, the performance of our purely appearance-based method is less competitive. There are several possible explanations for this. It might be that the generic scene appearance features utilise the limited training data uneconomically and other domain-specific modalities would be needed to take best use of the scarce training examples. For location recognition, these could include the depth information, the temporal continuity of the frame sequence and information based on pair-wise matching of images.

Yet, it is also possible that better performance could be achieved on basis of the generic appearance features by better system design. In particular, there might be some overlearning issues. With the larger training set, just memorising all the camera views appearing in the training material might be a viable strategy, whereas the smaller training set calls for generalising between views. A naive use (such as here) of a rich and distinctive scene representation might actually lead to worse performance than a feature extraction scheme with more limited distinguishing power if the inter-view generalisation issue is not properly taken care of. Our experiments reported here are insufficient to confirm either one of these hypotheses.

Our experiments back up our earlier findings that fusion of a large set of features consistently results in a much better visual category recognition accuracy than the use of any single feature alone.

References

1. Ajanki, A., Billinghamurst, M., Kandemir, M., Kaski, S., Koskela, M., Kurimo, M., Laaksonen, J., Puolamäki, K., Tossavainen, T.: Ubiquitous contextual information access with proactive retrieval and augmentation. In: Proceedings of 4th International Workshop on Ubiquitous Virtual Reality 2010 at Pervasive 2010, Helsinki, Finland (May 2010)
2. Feiner, S., MacIntyre, B., Höllerer, T., Webster, A.: A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. *Personal and Ubiquitous Computing* 1(4), 208–217 (1997)
3. ISO/IEC: Information technology - Multimedia content description interface - Part 3: Visual, 15938-3:2002(E) (2002)
4. Laaksonen, J., Koskela, M., Oja, E.: PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. *IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing* 13(4), 841–853 (2002)
5. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. of IEEE CVPR, vol. 2, pp. 2169–2178 (2006)
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
7. Mikolajczyk, K., Schmid, C.: Scale and affine point invariant interest point detectors. *International Journal of Computer Vision* 60(1), 68–86 (2004)
8. Pronobis, A., Caputo, B.: COLD: COsy Localization Database. *The International Journal of Robotics Research (IJRR)* 28(5) (May 2009)
9. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (in press, 2010)
10. Sjöberg, M., Viitaniemi, V., Koskela, M., Laaksonen, J.: PicSOM experiments in TRECVID 2009. In: Proceedings of the TRECVID 2009 Workshop, Gaithersburg, MD, USA (November 2009)
11. Smeaton, A.F., Over, P., Kraaij, W.: High-Level Feature Detection from Video in TRECVID: a 5-Year Retrospective of Achievements. In: Divakaran, A. (ed.) *Multimedia Content Analysis, Theory and Applications*, pp. 151–174. Springer, Berlin (2009)