Sparse Patch-Histograms for Object Classification in Cluttered Images

Thomas Deselaers, Andre Hegerath, Daniel Keysers, and Hermann Ney

Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Aachen, Germany
deselaers@cs.rwth-aachen.de

Abstract. We present a novel model for object recognition and detection that follows the widely adopted assumption that objects in images can be represented as a set of loosely coupled parts. In contrast to former models, the presented method can cope with an arbitrary number of object parts. Here, the object parts are modelled by image patches that are extracted at each position and then efficiently stored in a histogram. In addition to the patch appearance, the positions of the extracted patches are considered and provide a significant increase in the recognition performance. Additionally, a new and efficient histogram comparison method taking into account inter-bin similarities is proposed. The presented method is evaluated for the task of radiograph recognition where it achieves the best result published so far. Furthermore it yields very competitive results for the commonly used Caltech object detection tasks.

1 Introduction

In the last years, part-based models in general, and patch-based models in particular, have gained an enormous amount of interest in the computer vision community [1, 2, 3]. These approaches offer some immediate advantages such as robustness against occlusion and translation invariance because the parts can be modeled more or less independently and thus an object that is partly occluded can be classified correctly as long as the visible parts can be recognized.

Nearly all aproaches presented extract features only from a subset of positions in the images: most approaches use interest point detectors [1, 2, 3], random points [4], or points from a regular grid [5]. Obviously, by choosing a subset of feature extraction points, image information is lost which may result in decreased recognition performance. This may be passable in the case of general object recognition and detection, but can be unsuitable in the case of medical image analysis where no details may be missed. In contrast to all these approaches, the method presented here can efficiently deal with arbitrarily many features and thus we choose to extract several features at each position of the image. Only recently, some approaches that extract local features from all positions in the image were proposed [6, 7].

Similar to other approaches [5, 8], the presented approach uses patches, i.e. subimages, extracted from the images. Feature vectors representing the patches are derived from a PCA dimensionality reduction. These feature vectors are then stored in a special histogram structure that allows us to store high-dimensional feature vectors, which are then classified using various classification methods.

Another type of information that is often discarded when part-based models are applied is the spatial relationship between the parts. Many approaches completely discard these data [1, 5], and other approaches that explicitly model spatial relationships [8] have to be greatly simplified in order to become computationally feasible [2]. In the model presented here, the positions of the patches can be integrated directly without significant increase in computation time or storage requirements.

Furthermore, many approaches require time-consuming preprocessing steps such as vector quantization, to create a code-book of possible object parts [5,8,9]. Our approach skips this step and instead uses a generalized form of a code-book that is identical for all kinds of data. That is, the code-book is not learned from training data but is fixed before we know what data we will deal with. Obviously, this code-book needs a large amount of possible '*code-words*' but due to an efficient representation this becomes computationally feasible.

The remainder of this paper is structured as follows: In the next section, we introduce the feature extraction technique and the sparse histogram representation of the images. In Section 3 we shortly introduce the three classification methods that are used to recognize the images represented by the sparse histograms. Section 4 describes the databases used to evaluate the methods and Section 5 presents and compares the experimental results with the best results published so far. Finally, the paper is shortly summarized and concluded in Section 6.

2 Sparse Histograms of Image Patches

Histograms are a well-known method to represent the distribution of data and are applied in the field of computer vision in various ways. One problem with histograms is that they become difficult to handle if the dimensionality of the input data is large, because the number of bins in a histogram grows exponentially with the number of dimensions of the data. For example, given 8 dimensional input data and only 4 subdivisions per dimension results in $4^8 = 65,536$ bins.

To overcome this problem, we propose to use a sparse representation of the histograms, i.e. we store only those bins whose content is not empty. Sparse histograms have been used for other applications before [10]. This representation allows us to create histograms for data of arbitrary dimensionality. The only practical limitation to the size of the histogram is that for very large sizes, most of the bins that actually contain an element will contain only one single element, and this makes the comparison of histograms very unreliable.

2.1 Features.

It has been shown that patches extracted from the images are a suitable means of representing local structures in images [5, 8, 9]. Thus, we choose to extract patches of different sizes at every position in each image. More precisely, we extract square patches with the edge lengths 7, 11, 21, and 31 pixels, which are then scaled to a common size of 15 pixels to be able to process them jointly later. These multiple patch sizes allow to account for objects of various sizes and lead to a certain invariance with respect to scale changes. A very similar approach to account for different scales was used in [11].

All patches are extracted from all training images and then a PCA transformation is jointly estimated. Using this PCA transformation all patches are reduced in dimensionality.

2.2 Creation of histograms.

The distribution of the feature vectors described in the previous section is then approximated using a histogram. To reduce the necessary storage, the histograms are created without explicitly storing any feature vector. Thus, the creation of the histograms is a three step procedure: in the first step, the PCA transformation is determined as described above. In the second step, the mean and the variance of the transformed patches are calculated to determine a reasonable grid for the histograms. In the last step, the histograms themselves are created. For each of these steps, all training images are considered.

1. In the first step, all possible patches in various sizes from all training images are extracted and their mean and the covariance matrix are estimated to determine the PCA transformation matrix.
2. Given this PCA transformation matrix and the means, the mean μ_d and the variance σ_d^2 for each component d of the transformed vectors is calculated to determine the bin boundaries for the histograms. The bins for component d are uniformly distributed between $\mu_d - \alpha\sigma_d$ and $\mu_d + \alpha\sigma_d$.
3. Then, we consider all dimensionality reduced patches from the training images and create one histogram per training image. This step is depicted in Figure 1. The processing is from left to right: first the patches are extracted, then PCA transformed, then the position of the patch is concatenated to the PCA transformed feature vector, and finally the vectors are inserted into the sparse histogram data structure.

As mentioned above, the patches are not explicitly stored in any of these steps as this would lead to immense memory requirements.

Informal experiments have shown that 6 to 8 dimensions for the PCA reduced vectors lead to the best results, and that $\alpha = 1.5$ is a good value to determine bin boundaries. Values exceeding the given range are clipped.

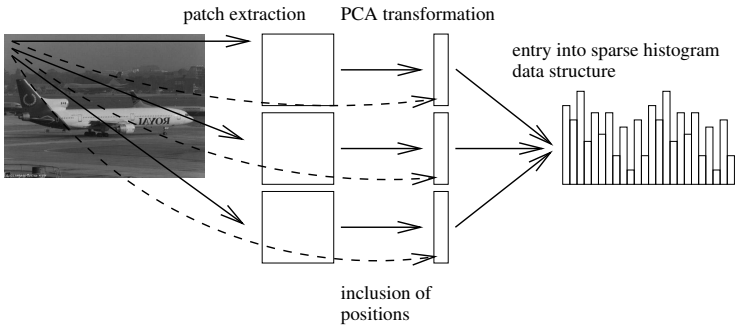

Fig. 1. Creation of sparse histograms. Solid arrows denote appearance information of the patches, dotted arrows denote spatial information of the patches

Spatial Information. One serious issue with many part-based models is the incorporation of spatial information. To incorporate spatial information in our approach, we simply concatenate the extraction position to the PCA reduced feature vectors and thus simply add two further components to the histograms. These additional components can easily be handled by the histograms. As the range of values for each component is calculated individually and independently of the other components, no special processing ot these additional components is required. One issue with the inclusion of the absolute patch extraction positions is that translation invariance, normally one of the major advantages of part-based models, is partly lost. Still, currently it is unclear how to incorporate relative position information into the model presented here. It will be shown later that for the tasks considered here, either the translation invariance is not required, or translations are sufficiently represented in the training data.

3 Classification of Sparse Patch Histograms

Given the sparse histograms that represent the images, any classifier that is able to handle the sparse representation can be used. We have tested three different classifiers: the nearest neighbor classifier in which we use two different distance functions, a classifier based on log-linear models trained using the maximum entropy criterion, and support vector machines.

3.1 Nearest Neighbor Classification

Nearest neighbor classification is often used as a baseline for classification. Immediate advantages are that no expensive training process is necessary, implementation can be done easily, and different distance functions can be used to compare the data used. In accordance with [12] we use Jeffrey Divergence to compare histograms. To classify the histogram h representing the image X the

following decision rule $r(x)$ is used:

$$h \mapsto r(h) = \arg\min_k \left\{ \min_{n=1\ldots N_k} d(h, h_n) \right\}, \tag{1}$$

where h_n is the histogram representing the nth training image from class k. The Jeffrey Divergence $d(h, h')$ between two histograms h and h' is defined as

$$d(h, h') = \sum_{c=1}^{C} h_c \log \frac{2h_c}{h_c + h'_c} + h'_c \log \frac{2h'_c}{h'_c + h_c}. \tag{2}$$

Here, h_c and h'_c are the cth bins of the histograms h and h', respectively.

One problem with the Jeffrey Divergence is that similarities between neighboring bins are completely neglected. Other distance measures that take into account inter-bin-similarities, for example the earth mover's distance [12], are too computationally expensive to be used for histograms with several thousand bins. We propose to use a much simpler way of taking into account neighboring bins that is inspired by an image matching algorithm [13]. This method is called *Histogram Distortion Model* (HDM) and it can be implemented for any bin-by-bin histogram comparison measure straightforwardly, as long as neighborhoods are defined for the underlying histograms. Given a bin at position $c = (c_1, \ldots c_D)$, we use the bin from position γ out of the neighborhood $U(c)$ of c that minimizes the resulting distance. Here, we use it as an extension to the Jeffrey Divergence, i.e., we replace the distance function $d(h, h')$ by $d_{\mathrm{HDM}}(h, h')$ with

$$d_{\mathrm{HDM}}(h, h') = \sum_{c=1}^{C} \min_{\gamma \in U(c)} h_c \log \frac{2h_c}{h_c + h'_\gamma} + h'_\gamma \log \frac{2h'_\gamma}{h'_\gamma + h_c}. \tag{3}$$

A related but computationally more expensive way to account for neighboring bins in the comparison of histograms would be to smooth the histograms. Here, the smoothing would lead to non-sparse histograms and thus it would lead to greatly increased computational requirements.

3.2 Maximum Entropy Classification

Maximum entropy classification and log-linear models are a well-known way to model probability distributions in natural language processing and in image recognition [14].

The maximum entropy approach directly optimizes the class posterior probability $p(k|X)$. Thus, it is a discriminatively trained model. Here, we want to model the posterior probability $p(k|h)$ where h is the sparse histogram representing image X. Thus, the model for $p(k|h)$ is

$$p(k|h) = \frac{1}{Z(h)} \exp\left(\alpha_k + \sum_{c=1}^{C} \lambda_{kc} h_c\right) \tag{4}$$

where h_c is the cth bin of the histogram h and $Z(h)$ a normalization factor.

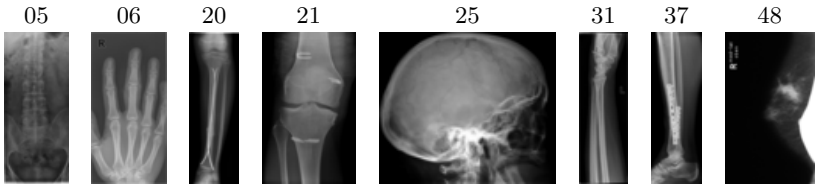

Fig. 2. Example images of the IRMA 10000 database together with their class.

Efficient algorithms to determine the parameters $\{\alpha_k, \lambda_{kc}\}$ exist. We use a modified version of generalized iterative scaling [15] to decrease the necessary computational effort. For classification, Bayes' decision rule is used:

$$h \mapsto r(h) = \arg\max_k \{p(k|h)\}\,. \tag{5}$$

3.3 Support Vector Machines

Support-vector-machines (SVM) are often used as a classification method that provides reasonable performance across various tasks. In the experiments we tried linear, polynomial and radial basis function kernels and optimized all parameters in cross-validation experiments on the training data.

4 Databases and Experimental Results

This section briefly presents the two databases used to evaluate our method, the IRMA 10000 database of medical radiographs and three of the Caltech object databases.

4.1 IRMA 10000.

The IRMA 10000 database[1] was used in the automatic annotation task of the 2005 ImageCLEF evaluation [17]. It consists of 10,000 fully classified radiographs taken randomly from medical routine at a large hospital. The images are split into 9,000 training and 1,000 test images and are subdivided into 57 classes. Example images for some of the classes are given in Figure 2. In the ImageCLEF 2005 automatic annotation task a total of 40 runs were submitted by 12 groups. In Table 1 we give the best results from the evaluation and compare our results to these. To keep the computing requirements low, we scaled all images such that the longest edge was 128 pixels while preserving the aspect ratio.

4.2 Caltech databases.

To compare the performance of our method to object recognition algorithms from other groups, we use some of the Caltech databases that were introduced

[1] http://irma-project.org

Fig. 3. Example images from the Caltech data sets airplanes, faces, and motorbikes, and a background image.

by Fergus et al. [8]. The task is to determine whether an object is present in an image or not. For this purpose, several sets of images containing certain objects (airplanes, faces, and motorbikes) and a set of background images not containing any of these objects [2] are given. The images are of various sizes and for the experiments they were converted to gray images. The airplanes and the motorbikes task consist of 800 training and 800 test images each, the faces task consists of 436 training and 434 test images. For each of these tasks, half of the images contain the object of interest and the other half does not. An example image of each set is shown in Figure 3. Many different groups have published results for these data. In Table 2 we summarize the best results we are aware of for each of the tasks to compare our results to. Here, we scaled the images to a common height of 128 pixels to keep the computing requirements low and to avoid the known issue that it is possible to classify some of the images just by image size [5].

5 Experimental Results

In this section, we present the results we obtained using sparse histograms of image patches for the IRMA and the Caltech tasks.

Table 1 gives an overview of the best results obtained for the IRMA tasks from the ImageCLEF 2005 evaluation [17] along with the results we obtained using sparse patch histograms with and without position information. For all experiments, the patches were reduced to 6 components using PCA. For the experiments with position, two components representing position were concatenated to the data vector thus resulting in 8 dimensional data. For all experiments, each component was subdivided into four steps, thus resulting in 4,096 and 65,536 bin-histograms for the experiments without and with spatial information respectively. These parameters were determined in informal cross-validation experiments to perform best on the average: For dimensionality reduction we measured the performance for dimensionalities between 4 and 10. Furthermore, we tried 2 to 6 subdivisions per component.

The results we obtained for this task are better than all results that are published for these data so far. With and without positions, the error rate is greatly improved using the histogram distortion model in comparison to using

[2] http://www.robots.ox.ac.uk/~vgg/data

Table 1. Results for the IRMA data. The comparison results are taken from the ImageCLEF 2005 automatic annotation task [17].

method	rank	group	error rate [%]
image distortion model	1	RWTH Aachen	12.6
image distortion model & texture feature	2	IRMA Group	13.3
patch-based object classifier (maximum entropy)	3	RWTH Aachen	13.9
patch-based object classifier (boosting)	4	Uni Liège	14.1
image distortion model & texture feature	5	IRMA Group	14.6
patch-based object classifier (decision trees)	6	Uni Liège	14.7
GNU image finding tool	7	Uni Geneva	20.6
32×32 images, Euclidean distance, nearest neighbor	-	-	36.8
sparse histograms (w/o position)		this work	
+ nearest neighbor			13.0
+ histogram distortion model, nearest neighbor			12.5
+ maximum entropy classification			11.6
+ support vector machine			11.3
sparse histograms (w/ position)		this work	
+ nearest neighbor			10.1
+ histogram distortion model, nearest neighbor			9.8
+ maximum entropy classification			**9.3**
+ support vector machine			10.0

only the Jeffrey Divergence. This shows that the histogram distortion model is, at least partly, able to compensate for the sparseness of the histograms. As mentioned above, an alternative to the histogram distortion model would be to smooth the histograms, but informal experiments have shown that, apart from the problems of storage, the improvement is lower than using the deformation model. The result obtained using maximum entropy training is again clearly improved for the case without position information. For the case with position information, the maximum entropy training cannot improve on the results.

In Table 2, results for the experiments on the three Caltech tasks are given. The first part of the table gives the best results we know for each of these tasks, the second part gives the results we obtained. We highlighted the best results in total and the best results we obtained with our method. Here again, using the histogram distortion model usually gave an improvement over the normal Jeffrey Divergence, and a further improvement can be achieved using the discriminatively trained log-linear model. Although the model we present is clearly much simpler than the models presented in [1,2,4,8,11], we achieve very competitive error rates. Using SVMs, the results are in the same area as those using the maximum entropy training. For both maximum entropy and SVM classifiers the results are better than those obtained using the nearest neighbor classification rule. This clearly shows that discriminative modeling can improve the results.

Table 2. Results for the Caltech data

method		error rate airp.	faces	motb.
constellation model	[8]	9.8	3.6	7.5
improved constellation model	[2]	6.3	9.7	2.7
PCA SIFT features	[18]	2.1	0.3	5.0
patch-histograms, discriminative training	[11]	1.4	3.7	**1.1**
boosting weak hypotheses	[1]	2.5	**0.0**	5.7
texture features	[19]	**0.8**	1.6	7.4
sparse histograms (w/o position)				
+ nearest neighbor		4.9	12.7	6.1
+ histogram distortion model, nearest neighbor		4.8	13.6	7.0
+ maximum entropy classification		3.5	7.8	4.8
+ support vector machines		2.4	4.1	2.3
sparse histograms (w/ position)				
+ nearest neighbor		9.1	6.5	6.8
+ histogram distortion model, nearest neighbor		6.5	7.6	6.9
+ maximum entropy classification		1.9	**3.9**	1.8
+ support vector machines		**0.8**	4.4	**1.3**

6 Conclusion

In this work we presented a part-based approach to object recognition that was evaluated on a database of medical radiographs and on three object recognition tasks. An advantage of this novel approach over other approaches is that it does not require large parts of the data to be disregarded, but instead almost arbitrary numbers of image patches can be handled by using a sparse histogram representation. Possible problems resulting from data sparseness are effectively counteracted by using a histogram distortion model which also improves the recognition results. Furthermore, the approach does not require an expensive training process, as the code-book is determined independently from the training data. The results obtained are the best published results for the task of radiograph recognition and are very competitive for the Caltech object recognition tasks. It was also shown that spatial information can easily be incorporated into the approach and that this information, although to the cost of loosing translation invariance, can improve the results notably for the restricted domain task of radiograph recognition and in most cases for the Caltech tasks.

In the future we plan to extend the presented model to incorporate relative patch positions.

References

1. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. **28**(3) (2006) 416–431

2. Fergus, R., Perona, P., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05). (2005) 380–389
3. Ulusoy, I., Bishop, C.M.: Generative versus discriminative methods for object recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05). (2005) 258–265
4. Marée, R., Geurts, P., Piater, J., Wehenkel, L.: Random subwindows for robust image classification. In: IEEE Conference on Computer Vision and Pattern Recognition. (2005) 34–40
5. Deselaers, T., Keysers, D., Ney, H.: Discriminative training for object recognition using image patches. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2., San Diego, CA (2005) 157–162
6. Bosch, A., Zissermann, A., Muñoz, X.: Scene classification via plsa. In: ECCV 2006. Volume 3954 of LNCS., Graz, Austria (2006) 517–530
7. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: ECCV 2006. Volume 3951 of LNCS., Graz, Austria (2006) 1–15
8. Fergus, R., Perona, P., Zissermann, A.: Object class recognition by unsupervised scale-invariant learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 03), Blacksburg, VG (2003) 264–271
9. Leibe, B., Schiele, B.: Scale invariant object categorization using a scale-adaptive mean-shift search. In: DAGM. Number 3175 in LNCS (2004) 145–153
10. Linde, O., Lindberg, T.: Object recognition using composed recpetive field histograms of higher dimensionality. In: International Conference on Pattern Recognition, Cambridge, UK (2004)
11. Deselaers, T., Keysers, D., Ney, H.: Improving a discriminative approach to object recognition using image patches. In: DAGM 2005, Pattern Recognition, 26th DAGM Symposium. Number 3663 in Lecture Notes in Computer Science, Vienna, Austria (2005) 326–333
12. Puzicha, J., Rubner, Y., Tomasi, C., Buhmann, J.: Empirical evaluation of dissimilarity measures for color and texture. In: International Conference on Computer Vision. Volume 2., Corfu, Greece (1999) 1165–1173
13. Keysers, D., Gollan, C., Ney, H.: Local context in non-linear deformation models for handwritten character recognition. In: International Conference on Pattern Recognition. Volume 4., Cambridge, UK (2004) 511–514
14. Jeon, J., Manmatha, R.: Using maximum entropy for automatic image annotation. In: Proceedings of the 3rd International Conference on Image and Video Retrieval. (2004) 24–32
15. Darroch, J.N., Ratcliff, D.: Generalized iterative scaling for log-linear models. Annals of Mathematical Statistics **43**(5) (1972) 1470–1480
16. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd edn. Morgan Kaufmann, San Francisco (2005)
17. Clough, P., Mueller, H., Deselaers, T., Grubinger, M., Lehmann, T., Jensen, J., Hersh, W.: The clef 2005 cross-language image retrieval track. In: Working Notes of the CLEF Workshop, Vienna, Austria (2005)
18. Zhang, W., Yu, B., Zelinsky, G.J., Samaras, D.: Object class recognition using multiple layer boosting with heterogeneous features. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05). (2005) 323–330
19. Deselaers, T., Keysers, D., Ney, H.: Classification error rate for quantitative evaluation of content-based image retrieval systems. In: International Conference on Pattern Recognition 2004. Volume 2., Cambridge, UK (2004) 505–508