Weak Hypotheses and Boosting for Generic Object Detection and Recognition

A. Opelt[1,2], M. Fussenegger[1,2], A. Pinz[2], and P. Auer[1]

[1] Institute of Computer Science,
8700 Leoben, Austria
{auer,andreas.opelt}@unileoben.ac.at
[2] Institute of Electrical Measurement and Measurement Signal Processing,
8010 Graz, Austria
{fussenegger,opelt,pinz}@emt.tugraz.at

Abstract. In this paper we describe the first stage of a new learning system for object detection and recognition. For our system we propose Boosting [5] as the underlying learning technique. This allows the use of very diverse sets of visual features in the learning process within a common framework: Boosting — together with a weak hypotheses finder — may choose very inhomogeneous features as most relevant for combination into a final hypothesis. As another advantage the weak hypotheses finder may search the weak hypotheses space without explicit calculation of all available hypotheses, reducing computation time. This contrasts the related work of Agarwal and Roth [1] where Winnow was used as learning algorithm and all weak hypotheses were calculated explicitly. In our first empirical evaluation we use four types of local descriptors: two basic ones consisting of a set of grayvalues and intensity moments and two high level descriptors: moment invariants [8] and SIFTs [12]. The descriptors are calculated from local patches detected by an interest point operator. The weak hypotheses finder selects one of the local patches and one type of local descriptor and efficiently searches for the most discriminative similarity threshold. This differs from other work on Boosting for object recognition where simple rectangular hypotheses [22] or complex classifiers [20] have been used. In relatively simple images, where the objects are prominent, our approach yields results comparable to the state-of-the-art [3]. But we also obtain very good results on more complex images, where the objects are located in arbitrary positions, poses, and scales in the images. These results indicate that our flexible approach, which also allows the inclusion of features from segmented regions and even spatial relationships, leads us a significant step towards generic object recognition.

1 Introduction

We believe that a learning component is a necessary part of any generic object recognition system. In this paper we investigate a principle approach for learning objects in still images which allows the use of flexible and extendible

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 71–84, 2004.

sets of features for describing objects and object categories. Objects should be recognized even if they occur at abitrary scale, shown from different perspective views on highly textured backgrounds. Our main learning technique relies on Boosting [5]. Boosting is a technique for combining several weak classifiers into a final strong classifier. The weak classifiers are calculated on different weightings of the training examples to emphasize different portions of the training set. Since any classification function can potentially serve as a weak classifier we can use classifiers based on arbitrary and inhomogeneous sets of image features. A further advantage of Boosting is that weak classifiers may be calculated when needed instead of calculating unnecessary hypotheses a priori.

In our learning setting, the learning algorithm needs to learn an object category. It is provided with a set of labeled training images, where a positive label indicates that a relevant object appears in the image. The objects are not segmented and pose and location are unknown. As output, the learning algorithm delivers a final classifier which predicts if a relevant object is present in a new image. Having such a classifier, the localization of the object in the image is straightforward. The image analysis transforms images to greyvalues and extracts normalised regions around interest (salient) points to obtain reduced representations of images. As an appropriate representation for the learning procedure we calculate local descriptors of these patches. The result of the training procedure is saved as the final hypothesis which is later used for testing (see figure 1).

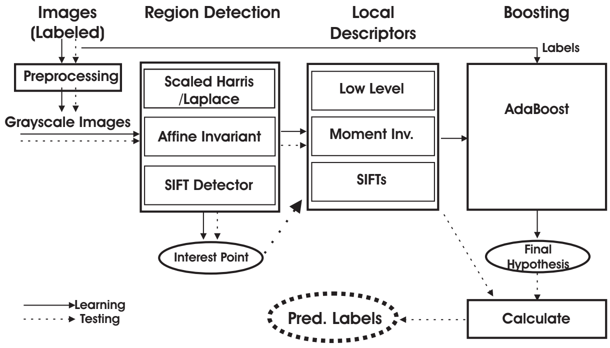


Fig. 1. Overview showing the framework for our approach for generic object recognition. The solid arrows show the training cycle, the dotted ones the testing procedure.

We describe our general learning approach in detail in section 2. In section 3, we discuss the image analysis steps, including illumination and size normalisation, interest point detection, and the extraction of the local descriptors. An explicit explanation of how we calculate the weak hypotheses used by the Boosting algorithm, is given in section 4. Section 5 contains a description of the setup we used for our experiments. The results are presented and compared with other approaches for object recognition. We regard the present system as a first step and further work is outlined in section 6.

1.1 Related Work

Clearly there is an extensive body of literature on object recognition (e.g. [3], [2], [22], [24], [6], [14]). In general, these approaches use image databases which show the object of interest at prominent scales and with only little variation in pose. We discuss only some of the most relevant and most recent results related to our approach.

Boosting was successfully used by Viola and Jones [22] as the ingredient for a fast face detector. The weak hypotheses were the thresholded average brightness of collections of up to four rectangular regions. In our approach we experiment with much larger sets of features to be able to perform recognition of a wider class of objects.

Schneiderman and Kanade [20] used Boosting to improve an already complex classifier. In contrast, we are using Boosting to combine rather simple classifiers by selecting the most discriminative features.

Agarwal and Roth [1] used Winnow as the underlying learning algorithm for the recognition of cars from side views. For this purpose images were represented as binary feature vectors. The bits of such a feature vector can be seen as the outcomes of weak classifiers, one weak classifier for each position in the binary vector. Thus for learning it is required that the outcomes of all weak classifiers are calculated a priori. In contrast, Boosting only needs to find the few weak classifiers which actually appear in the final classifier. This substantially speeds up learning, if the space of weak classifiers carries a structure which allows the efficient search for discriminative weak classifiers. A simple example is a weak classifier which compares a real valued feature against a threshold. For Winnow, one weak classifier needs to be calculated for each possible threshold a priori[1], whereas for Boosting the optimal threshold can be determined efficiently when needed.

A different approach to object class recognition was presented by Fergus, Perona, and Zisserman [3]. They used a generative probabilistic model for objects built as constellations of parts. Using an EM-type learning algorithm they achieved very good recognition performance. In our work we have chosen a model-free approach for flexibility. If at all, the sets of weak classifiers we use can be seen as model classes, but with much less structure than in [3]. Furthermore, we propose Boosting as a very different learning algorithm from EM.

Dorko and Schmid [2] introduced an approach for constructing and selecting scale-invariant object parts. These parts are subsequently used to learn a classifier. They show a robust detection under scale changes and variations in viewing conditions, but in contrast to our approach, the objects of interest are manually pre-segmented. This dramatically reduces the complexity of distinguishing between relevant patches on the objects and background clutter.

[1] More efficient techniques for Winnow like using virtual threshold gates [13] do not improve the situation much.

2 Our Learning Model for Object Recognition

In our setup, a learning algorithm has to recognize objects from a certain category in still images. For this purpose, the learning algorithm delivers a classifier that predicts whether a given image contains an object from this category or not. As training data, labeled images $(I_1, \ell_1), \ldots, (I_m, \ell_m)$ are provided for the learning algorithm where $\ell_k = +1$ if I_k contains a relevant object and $\ell_k = -1$ if I_k contains no relevant object. Now the learning algorithm delivers a function $H : I \mapsto \hat{\ell}$ which predicts the label of image I. To calculate this classification function H we use the classical AdaBoost algorithm [5]. AdaBoost puts weights w_k on the training images and requires the construction of a weak hypothesis h which has some discriminative power relative to these weights, i.e.

$$\sum_{k:h(I_k)=\ell_k} w_k > \sum_{k:h(I_k)\neq\ell_k} w_k \ , \tag{1}$$

such that more images are correctly classified than misclassified, relative to the weights w_k. (Such a hypothesis is called weak since it needs to satisfy only a very weak requirement.) The process of putting weights and constructing a weak hypothesis is iterated for several rounds $t = 1, \ldots, T$, and the weak hypotheses h_t of each round are combined into the final hypothesis H.

In each round t the weight w_k is decreased if the prediction for I_k was correct $(h_t(I_k) = \ell_k)$, and increased if the prediction was incorrect. Different to the standard AdaBoost algorithm we vary the factor β_t to trade off precision and recall. We set

$$\beta_t = \begin{cases} \sqrt{\frac{1-\varepsilon}{\varepsilon}} * \eta & \text{if } \ell_k = +1 \text{ and } \ell_k \neq h_t(I_k). \\ \sqrt{\frac{1-\varepsilon}{\varepsilon}} & \text{else} \end{cases}$$

with ε being the error of the weak hypothesis in this round and η as an additional weight factor to control the update of wrongly classified positive examples.

Here two general comments are in place. First, it is intuitively quite clear that weak hypotheses with high discriminative power — with a large difference of the sums in (1) — are preferable, and indeed this is shown in the convergence proof of AdaBoost [5]. Second, the adaptation of the weights w_k in each round performs some sort of adaptive decorrelation of the weak hypotheses: if an image was correctly classified in round t, then its weight is decreased and less emphasis is put on this image in the next round, yielding quite different hypotheses h_t and h_{t+1}.[2] Thus it can be expected that the first few weak hypotheses characterize the object category under consideration quite well. This is particularly interesting when a sparse representation of the object category is needed.

Obviously AdaBoost is a very general learning technique for obtaining classification functions. To adapt it for a specific application, suitable weak hypotheses

[2] In fact AdaBoost sets the weights in such a way that h_t is *not* discriminative in respect to the *new* weights. Thus h_t is in some sense oblivious to the predictions of h_{t+1}.

have to be constructed. For the purpose of object recognition we need to extract suitable features from images and use these features to construct the weak hypotheses. Since AdaBoost is a general learning technique we are free to choose any type of features we like, as long as we are able to provide an effective weak hypotheses finder which returns discriminative weak hypotheses based on this set of features. The chosen features should be able to represent the content of images, at least in respect to the object category under consideration. Since we may choose several types of features, we represent an image I by a set of pairs $\mathcal{R}(I) = \{(\tau, v)\}$ where τ denotes the type of a feature and v denotes a value of this feature, typically a vector of reals. Then for AdaBoost a weak hypothesis is constructed from the representations $\mathcal{R}(I_k)$, labels ℓ_k, and weights w_k of the training images.

In the next section we describe the types of features we are currently using, although many other features could be used, too. In Section 4 we describe the effective construction of the weak hypotheses.

3 Image Analysis and Feature Construction

We extract features from raw images, ignoring the labels used for learning. To lower the number of the points in an image we have to attend to, we use an interest point detector to get salient points. We evaluate three different detectors, a scale invariant interest point detector, an affine invariant interest point detector, and the SIFT interest point detector ([15], [16], [12], see section 3.1). Using these salient points we can reduce the content of an image to a number of points (and their surroundings) while being robust against irrelevant variations in illumination and scale. Since the most salient points[3] may not belong to the relevant objects, we have to take a rather large number of points into account, which implies choosing a low threshold in the interest point detectors. The number of SIFTs is reduced by a vector quantization using k-means (similarly to Fergus et al. [3]). The pixels enclosing an interest point are refered to as a patch. Due to different illumination conditions we normalise each patch before the local descriptors are calculated. Representing patches through a local descriptor can be done in different ways. We use subsampled grayvalues, intensity moments, Moment Invariants and SIFTs here.

3.1 Interest Point Detection

There is a variety of work on interest point detection at fixed (e.g. [9,21,25,10]), and at varying scales (e.g. [11,15,16]). Based on the evaluation of interest point detectors by Schmid et al. [19], we decided to use the scale invariant Harris-Laplace detector [15] and the affine invariant interest point detector [16], both by Mikolajczyk and Schmid. In addition we use the interest point detector used by Lowe [12] because it is strongly interrelated with SIFTs as local descriptors.

[3] E.g. by measuring the entropy of the histogram in the surrounding [3] or doing a Principal Component Analysis.

The scale invariant detector finds interest points by calculating a scaled version of the second moment matrix M and localizing points where the Harris Measure $H = det(M) - \alpha trace^2(M)$ is above a certain threshold th. The characteristic scale for each of these points is found in scale-space by calculating the Laplacians $L(\mathbf{x}, \sigma) = |\sigma^2(L_{xx}(\mathbf{x}, \sigma) + L_{yy}(\mathbf{x}, \sigma))|$ for each desired scale σ and taking the one at which L has a maximum in an 8-neighbourhood of the point.

The affine invariant detector is also based on the second moment matrix computed at a point which can be used to normalise a region in an affine invariant way. The characteristic scale is again obtained by selecting the scale at which the Laplacian has a maximum. An iterative algorithm is then used which converges to affine invariant points by modifying the location, scale and neighbourhood of each point.

Lowe introduced an interest point detector invariant to translation, scaling and rotation and minimally affected by small distortions and noise [12]. He also uses the scale-space but built with a difference of Gaussian (DoG). Additionally, a scale pyramid achieved by bilinear interpolation is employed. Calculating the image gradient magnitude and the orientation at each point of the scale pyramid, salient points with characteristic scales and orientations are achieved.

3.2 Region Normalisation

To normalise the patches we have to consider illumination, scale and affine transformations. For the size normalisation we have decided to use quadratic patches with a side of l pixels. The value of l is a variable we vary in our experiments. We extract a window of size $w = 6 * \sigma_I$ where σ_I is the characteristic scale of the interest point delivered by the interest point detector. Scale normalisation is done by smoothing and subsampling in cases of $l < w$ and by linear interpolation otherwise. In order to obtain affine invariant patches the values of the transformation matrix resulting from the affine invariant interest point detector are used to normalise the window to the shape of a square, before the size normalisation.

For illumination normalisation we use Homomorphic Filtering (see e.g. [7], chapter 4.5). The Homomorphic Filter is based on an image formation model where the image intensity $I(x, y) = i(x, y)r(x, y)$ is modeled as the product of illumination $i(x, y)$ and reflectance $r(x, y)$. Elimination of the illumination part leads to a normalisation. This is achieved by applying a Fast Fourier Transform to the logarithm image $ln(I)$. Now the reflectance component can be separated by a high pass filter. After a back transformation and an exponentiation we get the desired normalised patch.

3.3 Feature Extraction

To represent each patch we have to choose some local descriptors. Local descriptors have been researched quite well (e.g. [4], [12], [18], [8]). We selected four local descriptors for our patches. Our first descriptor is simply a vector of all pixels in a patch subsampled by two. The dimension of this vector is $\frac{l}{4}^2$ which is rather high and increases computational complexity. As a second descriptor we use intensity moments $M_{I_{pq}}^a = \int\int_\omega i(x, y)^a x^p y^q \, dx \, dy$ with a as the degree and

$p+q$ as the order, up to degree 2 and order 2. Without using the moments of degree 0 we get a feature vector with a dimension of 10. This reduces the computational costs dramatically. With respect to the performance evaluation of local descriptors done by Mikolajczyk and Schmid [17] we took SIFTs (see [12]) as a third and Moment Invariants (see [8]) as a fourth choice. In this evaluation the SIFTs outmatched the others in nearly all tests and the Moment Invariants were in the middle ground for all aspects considered.

According to [8] we selected first and second order Moment Invariants. We chose the first order affine Invariant and four first order affine and photometric Invariants. Additionally we took all five second order Invariants described in [8]. Since the Invariants require two contours, the whole square patch is taken as one contour and rectangles corresponding to one half of the patch are used as a second contour. All four possibilities of the second contour are calculated and used to obtain the Invariants. The dimenson of the Moment Invariants description vector is 10.

As shown in [12] the description of the patches with SIFTs is done by multiple representations in various orientation planes. These orientation planes are blurred and resampled to allow larger shifts in positions of the gradients. A local descriptor with a dimension of 128 is obtained here for a circular region around the point with a radius of 8 pixels, 8 orientation planes and sampling over a 4x4 and a 2x2 grid of locations.

4 Calculation of Weak Hypotheses

Using the features constructed in the previous section, an image is represented by a list of features (τ_f, v_f), $f = 1, \ldots, F$, where τ_f denotes the type of a feature, v_f denotes its value as real vector, and F is the number of extracted features in an image. The weak hypotheses for AdaBoost are calculated from these features. For object recognition we have chosen weak hypotheses which indicate if certain feature values appear in images. For this a weak hypothesis h has to select a feature type τ, its value v, and a similarity threshold θ. The threshold θ decides if an image contains a feature value v_f that is sufficiently similar to v. The similarity between v_f and v is calculated by the Mahalanobis distance for Moment Invariants and by the Euclidean distance for SIFTs. The weak hypotheses finder searches for the optimal weak hypothesis — given labeled representations of the training images $(\mathcal{R}(I_1), \ell_1), \ldots, (\mathcal{R}(I_m), \ell_m)$ and their weights $w_1, \ldots, w_m$ calculated by AdaBoost — among all possible feature values and corresponding thresholds.

The main computational burden is the calculation of the distances between v_f and v, since they both range over all feature values that appear in the training images.[4] Given these distances which can be calculated prior to Boosting, the remaining calculations are relatively inexpensive. Details for the weak hypotheses finder are given in Figure 2. After sorting the optimal threshold for feature $(\tau_{k,f}, v_{k,f})$ can now be calculated in time $O(m)$ by scanning through the weights

[4] We discuss possible improvements in Section 6.

Input: Labeled representations $(\mathcal{R}(I_k), \ell_k)$,
$k = 1, \dots, m$, $\mathcal{R}(I_k) = \{(\tau_{k,f}, v_{k,f}) : f = 1, \dots, F_k\}$.
Distance functions: Let $d_\tau(\cdot,\cdot)$ be the distance in respect to the feature values of type τ in the training images.
Minimal distance matrix: For all features $(\tau_{k,f}, v_{k,f})$ and all images I_j calculate the minimal distance between $v_{k,f}$ and features in I_j,

$$d_{k,f,j} = \min_{1 \le g \le F_j : \tau_{j,g} = \tau_{k,f}} d_{\tau_{k,f}}(v_{k,f}, v_{j,g}) \ .$$

Sorting: For each k, f let $\pi_{k,f}(1), \dots, \pi_{k,f}(m)$ be a permutation such that

$$d_{k,f,\pi_{k,f}(1)} \le \dots \le d_{k,f,\pi_{k,f}(m)} \ .$$

Select best weak hypothesis (Scanline): For all features $(\tau_{k,f}, v_{k,f})$ calculate over all images I_j

$$\max_s \sum_{i=1}^{s} w_{\pi_{k,f}(i)} * \ell_{\pi_{k,f}(i)} \ .$$

and select the feature $(\tau_{i,f}, v_{i,f})$ where the maximum is achieved. **Select threshold θ:** With the position s where the scanline reached a maxium sum the threshold θ is set to

$$\theta = \frac{d_{k,f,\pi_{k,f}(s)} - d_{k,f,\pi_{k,f}(s+1)}}{2} \ .$$

Fig. 2. Explanation of the weak hypotheses finder.

$w_1, \dots, w_m$ in the order of the distances $d_{k,f,j}$. Searching over all features, the calculation of the optimal weak hypothesis takes $O(Fm)$ time.

To give an example of the absolute computation times we used a dataset of 150 positive and 150 negative images. Each image has an average number of approximately 400 patches. Using SIFTs one iteration after preprocessing requires about one minute computation time on a P4, 2.4GHz PC.

5 Experimental Setup and Results

We carried out our experiments as follows: the whole approach was first tested on the database used by Fergus et al. [3]. After demonstrating a comparable performance, the approach was tested on a new, more difficult database[5], see figure 5. These images contain the objects at arbitrary scales and poses. The images also contain highly textured background. Testing on these images shows that our approach still performs well. We have used two categories of objects, persons (P) and bikes (B), and images containing none of these objects (N). Our database contains 450 images of category P, 350 of B and 250 of category N. The recognition was based on deciding presence or absence of a relevant object.

[5] Available at $http://www.emt.tugraz.at/\sim pinz/data/$

Preparing our data set we randomly chose a number of images, half belonging to the object category we want to learn and half not. From each of these two piles we take one third of the images as a set of images for testing the achieved model. The performance was measured with the receiver-operating characteristic (ROC) corresponding error rate. We tested the images containing the object (e.g. category B) against non-object images from the database (e.g. categories P and N). Our training set contains 100 positive and 100 negative images. The tests are carried out on 100 new images, half belonging to the learned class and half not. Each experiment was done using just one type of local descriptor.

Figure 3(a) shows the recall-precision curve (RPC) of our approach (obtained by varying η), the approach of Fergus et al. [3] and the one of Agarwal and Roth [1], trained on the dataset used by Fergus et al. [3][6]. Our approach performs better than the one of Agarwal and Roth but slightly worse than the approach of Fergus et al.

Table 1 shows the results of our approach (using the affine invariant interest point detection and Moment Invariants) compared with the ones of Fergus et al. and other methods [23], [24], [1]. While they use a kind of scale and viewing direction normalisation (see [3]), we work on the original images. Our results are almost as good as the results of Fergus et al. for the motorbikes dataset. For the other datasets our error rate is somewhat higher than the one of Fergus et al., but mostly lower than the error rate of the other methods.

Table 1. The table gives the ROC equal error rates on a number of datasets from the database used by Fergus et al. [3]. Our results (using the affine invariant interest point detection and Moment Invariants) are compared with the results of the approach of Fergus et al. and other methods [23], [24], [1]. The error rates of our algorithm are between the other approaches and the ones of Fergus et al. in all cases except for the faces where the algorithm of Weber et al. [24] is also slightly better.

Dataset	Ours	Fergus et al. [3]	Others	Ref.
Motorbikes	92.2	92.5	84	[23]
Airplanes	88.9	90.2	68	[23]
Faces	93.5	96.4	94	[24]
Cars(Side)	83.0	88.5	79	[1]

This comparison shows that our approach performs well on the Fergus et al. database. We proceed with experiments on our own dataset and show some effects of parameter tuning[7]. Figure 3(b) shows the influence of the additional weighting of right positive examples in the Boosting algorithm (η). We can see that with a factor η smaller than 1.8, the recall increases faster than the precision

[6] Available at *http://www.robots.ox.ac.uk/ ∼ vgg/data/*

[7] Parametes not given in these tests are set to $\eta = 1.8$, $T = 50$, $l = 16px$, $th = 30000$, smallest scale is skipped. Depending on textured/homogenous background, the number of interest points detected in an image varies between 50 and 1000.

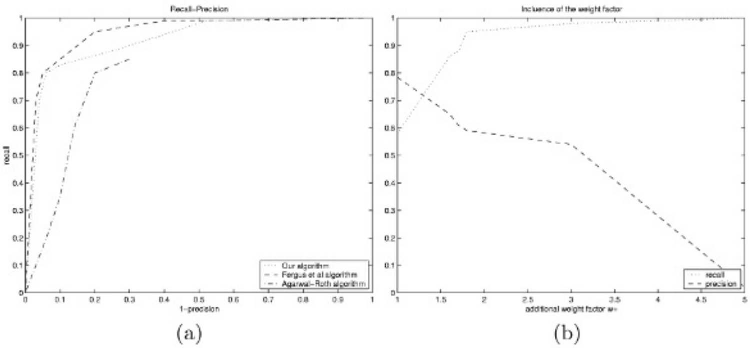


Fig. 3. The curves in (a) and (b) are obtained by varying the factor η. In (a) the diagram shows the recall-precision curve for [3], [1] and our approach on the cars (side) dataset. Our approach is superior to the one of Agarwal and Roth but slightly worse than the one of Fergus et al. The diagram (b) shows the influence of an additional factor η for the weights of correctly positive classified examples. The recall increases faster than the precision drops until a factor of 1.8.

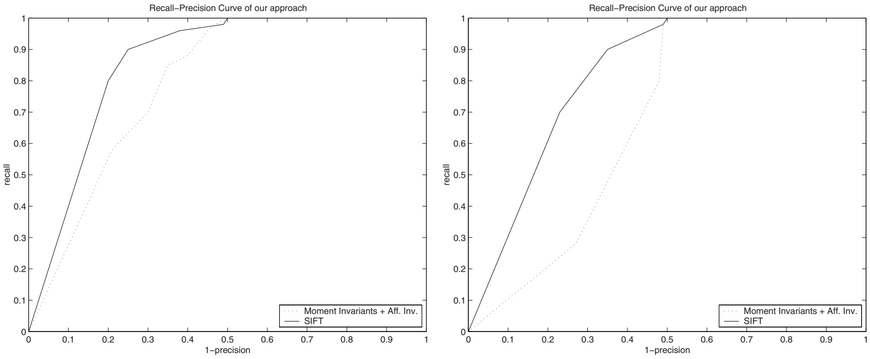
drops. Then both curves have nearly the same (but inverse) gradient up to a factor of 3. For $\eta > 3$ the precision decreases rapidly with no relevant gain of recall.

Table 2 presents the performance of the Moment Invarants as local descriptor, compared with our low level descriptors (using the affine invariant interest point detector). Moment Invariants delivered the best results but the other low level descriptors did not perform badly, either. This behaviour might be explained by the fact that the extracted regions are already normalised against the same set of transformations as the Moment Invariants.

Table 2. The table shows the results we reached with the three different kinds of local descriptors. We used an additional weight factor $\eta = 1.7$ here. Moment Invariants delivered the best results.

Local Descriptor	recall	precision
Moment Invariants	0.88	0.61
Intensity Moments	0.70	0.57
Subsampled Grayvalues	0.82	0.62

In table 3 the results of our approach using the scale invariant interest point detector compared with the use of the affine invariant interest point detector are shown. We also vary the additional weight for right positive classified examples η. The affine invariant interest point detector achieves better results for the recall but precision is higher when we use the scale invariant version of the interest



(a) (b)

Fig. 4. In (a) the recall-precision curve of our approach with Moment Invariants and the affine invariant interest point detection, and the recall-precision curve of our approach using SIFTs for category bike are shown. (b) shows the recall-precision curves with the same methods for the category person.

point detector. This is to be expected since the affine invariant detector allows for more variation in the image, which implies higher recall but less precision.

Table 3. The table shows the results of our approach using the scale invariant interest point detector compared with using the affine invariant interest point detector varying the additional weight for right positive classified examples η.

η	recall (scale inv.)	precision (scale inv.)	recall (affine inv.)	precision (affine inv.)
1.7	0.78	0.70	0.88	0.61
1.9	0.79	0.64	0.92	0.59
2.1	0.82	0.62	0.94	0.57

We skipped the smallest scale in our experiments because experiments show that this reduction of number of points does not have relevant influence to the error rates. Again, using the parameters that performed best, figure 4(a) shows an example of a recall-precision curve (RPC) of our approach trained on the bike dataset from our image database with Moment Invariants and the affine invariant interest point detection compared with our approach using SIFTs. Using the same methods we obtain the recall-precision curves (RPC) shown in figure 4(b) for the category person.

For directly comparing the results reached using the Moment Invariants with the affine invariant interest point detector or using SIFTs, the ROC equal error rates on various datasets are shown in table 4. As seen here the SIFTs perform better on our database. Tested on a category of the database from Fergus et al. one can see that the Moment Invariants perform better in that case.

Table 4. This table shows a comparison of the ROC equal error rates reached with the two high level features. On our database the SIFTs perform better, but on the database of Fergus et al. the Moment Invariants reach the better error rate.

Dataset	Moment Invariants	SIFTs
Airplanes	88.9	80.5
Bikes	76.5	86.5
Persons	68.7	80.8

6 Discussion and Outlook

In conclusion, we have presented a novel approach for the detection and recognition of object categories in still images. Our system uses several steps of image analysis and feature extraction, which have been previously described, but succeeds on rather complex images with a lot of background structure. Objects are shown in substantially different poses and scales, and in many of the images the objects (bikes or persons) cover only a small portion of the whole image. The main contribution of the paper, however, lies in the new concept of learning. We use Boosting as the underlying learning technique and combine it with a weak hypothesis finder. In addition to several other advantages of this approach, which have already been mentioned, we want to emphasize that this approach allows the formation of very diverse visual features into a final hypothesis. We think that this capability is the main reason for the good experimental results on our complex database. Furthermore, experimental comparison on the database used by Fergus et al. [3] shows that our approach performs similarly well to state-of-the-art object categorization on simpler images.

We are currently investigating extensions of our approach in several directions. Maybe the most obvious is the addition of more features to our image analysis. This includes not only other local descriptors like differential invariants [12], but also regional features[8] and geometric features[9]. To reduce the complexity of our approach we are considering a reduction of the number of features by clustering methods.

As the next step we will use spatial relations between features to improve the accuracy of our object detector. To handle the complexity of many possible relations between features, we will use the features constructed in our current approach (with parameters set for high recall) as starting points. Boosting will again be the underlying method for learning object representations as spatial combinations of features. This will allow the construction of weak hypotheses for discriminative spatial relations.

Acknowledgements. This work was supported by the European project LAVA (IST-2001-34405) and by the Austrian Science Foundation (FWF, project S9103-

[8] Regional features describe regions found by appearance based clustering.

[9] A geometric feature describes the appearance of geometric shapes, e.g. ellipses, in images.

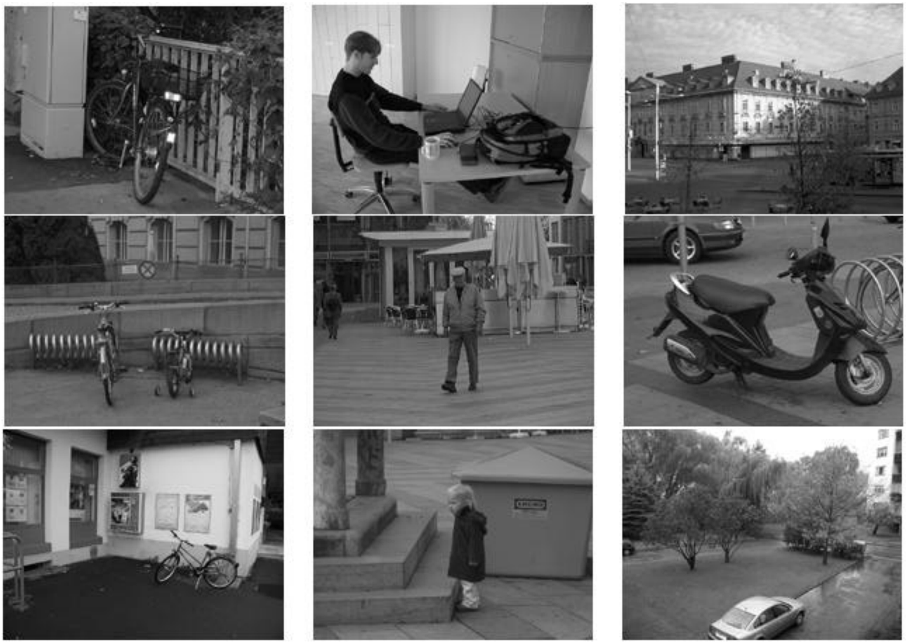


Fig. 5. Examples from our image data base. The first column shows three images from the object class bike, the second column contains objects from the class person and the images in the last column belong to none of the classes (called nobikenoperson). The second example in the last column shows a moped as a very difficult counter-example to the category of bikes.

N04). We are grateful to David Lowe and Cordelia Schmid for providing the code for their detectors/descriptors.

References

1. S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proc. ECCV*, pages 113–130, 2002.
2. Gy. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *Proc. International Conference on Computer Vision*, 2003.
3. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.
4. W. Freeman and E. Adelson. The design and use of steerable filters. In *PAMI*, pages 891 – 906, 1991.
5. Y. Freund and R. E. Schapire. A decision-theoretic generalisation of on-line learning. *Computer and System Sciences*, 55(1), 1997.
6. A. Garg, S. Agarwal, and T. S. Huang. Fusion of global and local information for object detection. In *Proc. CVPR*, volume 2, pages 723–726, 2002.
7. R. C. Gonzalez and R.E. Woods. *Digital Image Processing.* Addison-Wesley, 2001.

8. L. Van Gool, T. Moons, and D. Ungureanu. Affine / photometric invariants for planar intensity patterns. In *Proc. ECCV*, pages 642 – 651, 1996.
9. C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of the 4th ALVEY vision conference*, pages 147–151, 1988.
10. R. Laganiere. A morphological operator for corner detection. *Pattern Recognition*, 31(11):1643 – 1652, 1998.
11. T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 1996.
12. D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, 1999.
13. W. Maass and M. Warmuth. Efficient learning with virtual threshold gates. *Information and Computation*, 141(1):66–83, 1998.
14. S. Mahamud, M. Hebert, and J. Shi. Object recognition using boosted discriminants. In *Proc. CVPR*, volume 1, pages 551–558, 2001.
15. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. ICCV*, pages 525–531, 2001.
16. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*, pages 128–142, 2002.
17. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proc. CVPR*, 2003.
18. C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. In *PAMI*, volume 19, pages 530–534, 1997.
19. C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, pages 151–172, 2000.
20. H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, to appear.
21. E. Shilat, M. Werman, and Y. Gdalyahu. Ridge's corner detection and correspondence. In *Computer Vision and Pattern Recognition*, pages 976 – 981, 1997.
22. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, 2001.
23. M. Weber. *Unsupervised Learning of Models for Object Recognition.* PhD thesis, California Institute of Technology, Pasadena, CA, 2000.
24. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. ECCV*, 2000.
25. R. P. Wuertz and T. Lourens. Corner detection in color images by multiscale combination of end-stopped cortical cells. In *International Conference on Artificial Neuronal Networks*, pages 901 – 906, 1997.