

Object Recognition Using Segmentation for Feature Detection

Michael Fussenegger^{1,2}, Andreas Opelt^{1,2}, Axel Pinz² and Peter Auer¹

¹ Institute of Computer Science, University of Leoben, Austria

{auer, andreas.opelt}@unileoben.ac.at

² Institute of Electrical Measurement and Measurement Signal Processing, TU Graz, Austria

{fussenegger, opelt, pinz}@emt.tugraz.at

Abstract:

A new method is presented to learn object categories from unlabeled and unsegmented images for generic object recognition. We assume that each object can be characterized by a set of typical regions, and use a new segmentation method - "Similarity-Measure Segmentation" - to split the images into regions of interest. This approach may also deliver segments, which are split into several disconnected parts, which turns out to be a powerful description of local similarities. Several textural features are calculated for each region, which are used to learn object categories with Boosting. We demonstrate the flexibility and power of our method by excellent results on various datasets. In comparison, our recognition results are significantly higher than results published in related work.

1 Introduction

Generic object recognition requires a number of ingredients. Several approaches have been reported which differ in certain aspects, but share a common outline of system design and sequence of processes. First, features like points or regions have to be found, which can characterize an object category. These features have to be flexible enough to accommodate to a wide variety of object categories and to a certain object variability like changing scale, orientation, lighting and viewpoint. Next, these features have to be normalized and represented appropriately, so they can be compared and learned. Finally, object categories have to be learned by finding those features, which are well suited to characterize a certain category.

Generic object detection and recognition has recently gained a lot of attention in computer vision. Consequently, there is an extensive body of literature that deals with this topic (e.g. [13], [4], [9], [10]). Most of them like Fergus et al. [4], Opelt et al. [13] and Lowe [9] use a kind of scale invariant features, to learn object categories. Fergus et al. [4], for example, use the detector of Kadir and Brady [8], which finds regions that are salient in both location and scale.

Figure 1 shows our overall framework, which can handle various kinds of region detectors and local

descriptors. This paper presents a method based on segmentation to get regions (similar to Barnard et al. [2]), which are described by a vector of texture moments ([7]). Based on this representation of image regions, we use AdaBoost [5] to learn object categories. The corresponding boxes in fig. 1 are shaded in gray.

2 Method and Algorithms

Our approach to generic object recognition assumes that each object can be described by a set of typical regions which are either detected as discontinuities or due to their homogeneity. Discontinuities can be found at various scales and can be represented by a location and a support region in the image [13]. There is a variety of work dealing with the detection of these interest points (e.g. [11]). Homogeneous regions are found by region-based segmentation algorithms (e.g. [2]). Figure 1 shows that our framework can use both, discontinuities and homogeneities.

In each learning step, we train for a certain category. This is achieved by dividing the image dataset into two piles of images, one containing examples of the object category we want to learn and one not. In the first step discontinuous/homogeneous regions are detected by various methods. After that, we calculate a vector of local descriptors for these features, as a preparation for the learning module. The result of the training procedure - the classifier, is saved in the final hypothesis.

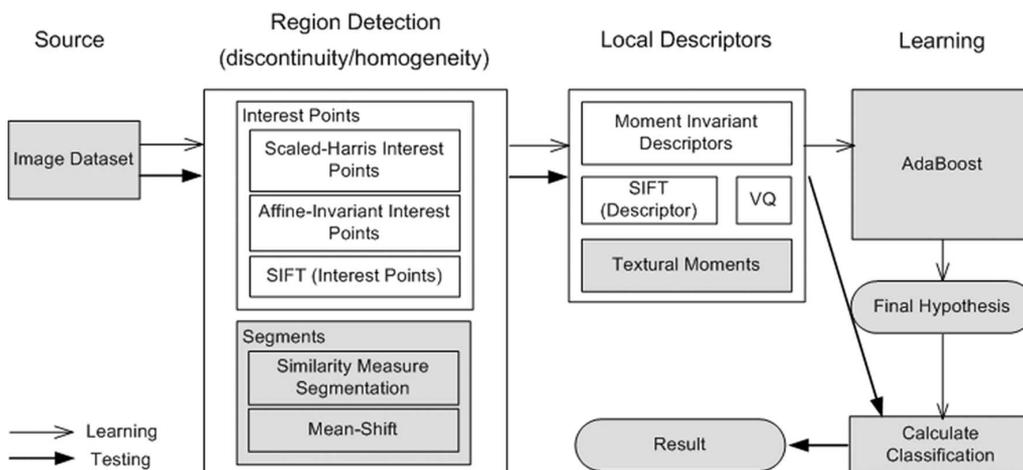


Figure 1: A sketch of our object recognition framework. We can use scaled and affine invariant interest points [11], SIFT interest points [9] and segmentation methods [3] for feature detection. Local descriptors are built using moment invariants [6], SIFT descriptors [9], or textural moments [7]. AdaBoost [5] is used to learn object categories. The gray boxes highlight the components described in this paper. For the discontinuity-based approach, see [13].

2.1 Similarity-Measure Segmentation

We have developed a new algorithm – “Similarity-Measure Segmentation” – which is described in detail, and compare its performance for object categorization with the well known mean shift

algorithm [3].

Stock and Pinz use a similarity measure (see equation 1) in [14] for the redetection of corners in an image sequence. We adapted this measure to describe pixel similarity for segmentation purpose. This similarity is used to split images into regions. SC_i defines an element of the Similarity-Criteria vector SC , in other words the distance of two pixels corresponding to a defined pixel feature. We are extracting two kinds of features. On one hand color, intensity, brightness and the position of a pixel, which consider only a single pixel. On the other hand texture measures ([7]), high-pass, Local Binary Patterns (see [12]) and Wavelets (see [1]), which consider a certain neighbourhood of a pixel. In our experiments described in section 3), we have used the combination of intensity, position and high-pass (3x3). Equation 2 shows the definition for one Similarity-Criteria element SC_i of the vector SC , where P_1 and P_2 are the two pixels and i is the index that defines the used feature. We use the Euclidean distance to calculate the elements of SC .

$$SM = \frac{\sum_i a_i e^{-\frac{SC_i}{2\pi\sigma_i}}}{\sum_i a_i} \quad 0 < SM \leq 1. \quad (1)$$

$$SC_i = f(P_1^i, P_2^i) \quad (2)$$

As Stock and Pinz show in [14] the parameters a_i can be used to weigh the different Similarity-Criteria. In our experiments (section 3), we don't use this feature, but we modify the sensitivity of each Similarity-Criterion SC_i by changing σ_i of the exponential part of equation 1. We introduce σ_c for the intensity-, σ_x for the position- and σ_t for the texture Similarity-Criterion. σ_c for the intensity depends on the contrast of the image. In other words σ_c is proportional to the variance σ_I^2 of the image (eq. 3). σ_x and σ_t are constant.

$$\sigma_c = \frac{\sigma_I^2}{128} * 3 \quad (3)$$

Our Similarity-Measure grouping algorithm consists of the following steps:

1. Take any unlabeled pixel in an image, define a new region R_j and label this pixel with RL_j .
2. Calculate the Similarity Measure to all other unlabeled pixels in the neighborhood, defined by a radius r .

3. Each pixel that has a similarity above a threshold t is also labeled with RL_j . Go back to step two for each newly labeled pixel.
4. If there aren't any newly labeled pixels, take the next unlabeled pixel and start again with step one, until all pixels have a region number RL_j .
5. Search all regions smaller than a minimum value reg_{min} , and merge each region with the nearest region larger than reg_{min} (equal to Mean-Shift segmentation [3]).

The radius r can be varied between 1 (to force connected regions) and r_{max} . The maximum radius r_{max} depends on the σ for the position σ_x and the threshold t . In other words, the smaller the threshold t the larger is the maximum radius r_{max} .

Depending on the radius r , it can happen, that some of our “regions” R_j are not connected. While this is in contradiction to the classical definition of segmentation, treating these R_j as entities for the subsequent learning process has shown recognition results, which are superior to results based on connected regions. We consider this new way of looking at disconnected segments a possibility to aggregate larger entities which are well suited to describe local homogeneities. These descriptions maintain salient local information and suppress spurious information which would lead to oversegmentation in other segmentation algorithms.

2.2 Local Descriptors

Segmentation leads to an image which is split into (potentially disconnected) regions of interest R_j . For each R_j we calculate several textural moments (see [7] for details):

- Mean μ
- Variance σ^2
- Coefficient of variation cv
- Smoothness R
- Skewness γ_1
- Kurtosis γ_2
- Gray level energy E

2.3 The Learning Model

For the learning of our object models, we need a learning technique, that allows us to choose freely any type of features. AdaBoost [5] as a general learning technique for obtaining classification functions, provides this functionality. Our learning algorithm delivers a classifier that predicts whether a given image contains objects of a certain category or not. As training data, labeled images $(I_1, l_1), \dots, (I_m, l_m)$ are provided where

$$l_k = \begin{cases} +1 & \text{if } I_k \text{ contains a relevant object} \\ -1 & \text{if } I_k \text{ contains no relevant object.} \end{cases}$$

The AdaBoost learning algorithm leads to a final hypothesis in form of a function $H : I \mapsto l_{pred}$ which predicts the label of image I . To calculate this function H AdaBoost puts weights ω_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.

$$\sum_{k:h(I_k)=l_k} \omega_k > \sum_{k:h(I_k) \neq l_k} \omega_k, \quad (4)$$

in a way that more images are correctly classified than misclassified, relative to the weights ω_k . This 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 several times $i = 1, \dots, T$, in our case $T = 100$, and all weak hypotheses h_i of each iteration are combined into a final hypothesis H (for details see [5] or [13]). Based on H , the system decides whether the test image contains the learned object category or not.

3 Experiments and Results

Experiments were carried out in two steps. First the whole approach was tested on two datasets, with Mean-Shift and Similarity-Measure segmentation and varying values for reg_{min} (see Table 1). We used the category cars(rear) trained versus the background images from the database used by Fergus et al. [4] and the category bikes trained versus the category persons, which is a more difficult dataset from our database described in [13]. Figure 2 shows examples of our images. Our training sets contained 60 positive and 60 negative images. The tests were carried out on 60 new images half belonging to the learned class and half not.

Table 1 shows the results comparing Mean-Shift and Similarity-Measure segmentation. For comparison we show our own results using affine invariant interest-points as described in [13]. In these



Figure 2: Examples from our image database. 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 images contain objects at arbitrary scales and poses as well as highly textured backgrounds.

experiments, object categorization works best with Similarity-Measure. Mean-Shift is comparable to affine invariant interest-points, performing better on bikes and slightly worse on cars.

Cars(rear)		
Method	$reg_{min} = 50$	$reg_{min} = 250$
Mean-Shift	15	18.3
Similarity-M.	8.3	11.7
Aff. Interst Point	13.3	
Bikes		
Method	$reg_{min} = 50$	$reg_{min} = 250$
Mean-Shift	18.3	23.3
Similarity-M.	15	20
Aff. Interst Point	33.3	

Table 1: Relative error on dataset cars(rear) [4] and bikes [13]. For two tests, we used segmentation one with $reg_{min} = 50$ and one with $reg_{min} = 250$. For the third test we used affine invariant interest-points. In all cases the object categorization works best with Similarity-Measure

We performed several further tests, only using Similarity-Measure segmentation, on the datasets used by Fergus et al. [4] and by Opelt et al. [13].

In all experiments, we used the following parameters: $\sigma_x = 1.2$, $\sigma_t = 0.5$, $t = 0.83$ and $reg_{min} = 50$. The performance was measured with the receiver-operating characteristic (ROC) equal error rate (see table 2).

Dataset	Ours	Others	Ref
Airplanes	97.8	90.2	[4]
Faces	99.9	96.4	[4]
Bikes	89.6	76.5	[13]
Cars(rear)	99.9	90.3	[4]
Cars(side)	99.9	88.5	[4]

Table 2: The table gives ROC equal error rates on a number of datasets from the databases used by Fergus et al. [4] and by Opelt et al. [13]. Our method based on Similarity-Measure Segmentation provides better results for all datasets.

Table 2 shows the results of our approach compared with Fergus et al. [4] and Opelt et al. [13]. In all cases, the performance of the algorithm is superior to the other methods, without being tuned for a particular dataset.

4 Summary and Outlook

In conclusion, we have shown a novel approach for object categorization and presented a new segmentation method based on Similarity-Measure. The recognition results presented here demonstrate the power of combining Similarity-Measure segmentation and AdaBoost. With weak supervised learning, we get in four cases less than 3% error rate, and an error rate of approximately 10% on a much more difficult dataset.

Currently, we are investigating extensions of our approach in several directions. We experiment with new feature descriptors and test other feature combinations for the Similarity-Criteria vector. We also work towards an integrated approach which combines several region detection algorithms and local descriptors.

5 Acknowledgment

This research was supported by LAVA: Learning for Adaptable Visual Assistants (EU-IST Programme IST-2001-34405) and by the Joint Research Program ‘Cognitive Vision’ of the Austrian Science Funds (FWF-JRP S9103-N04 and S9104-N04).

References

- [1] J. C. Feaveau A. Cohen, I. Daubechies. Biorthogonal bases of compactly supported wavlets. In *Commun. Pure Appl. Math.*, pages 485–560, 1992.
- [2] K. Barnard, P. Duygulu, R. Guru, and D. Forsyth. The effect of segmentation and feature choice in a translation model of object recognition. In *Proceedings of Computer Vision and Pattern Recognition*, volume 2, pages 675–682, 2003.
- [3] D. Comaniciu and P. Meer. Mean shift: A robust approach towards feature space analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24(5), pages 603–619, 2002.
- [4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2003.
- [5] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning. In *Computer and System Science*, volume 55, pages 119–139, 1997.
- [6] L. Van Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. In *Proceedings of European Conference on Computer Vision*, volume 2, pages 642–651, 1996.
- [7] R. M. Haralick. Statistical and structural approaches to texture. In *Proceedings of ICPR*, 1979.
- [8] T. Kadir and M. Brady. Scale, saliency and image description. In *IJCV*, volume 45(2), pages 83–105, 2001.
- [9] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 2, pages 1150–1157, 1999.
- [10] S. Mahamud, M. Hebert, and J. Shi. Object recognition using boosted discriminants. In *Proceedings of Computer Vision and Pattern Recognition*, volume 1, pages 551–558, 2001.
- [11] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of European Conference on Computer Vision*, volume 1, pages 128–142, 2002.
- [12] T. Ojala and M. Pietikinen. Unsupervised texture segmentation using feature distributions. In *Journal of Pattern Recognition*, volume 32, pages 477–486, 1999.
- [13] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proceedings of ECCV 2004*.
- [14] Ch. Stock and A. Pinz. Similarity measure for corner redetection. In *Proceedings of Scandinavian Conference on Image Analysis*, volume 1, pages 133–139, 2003.