Component Based Recognition of Objects in an Office Environment

Christian Morgenstern and Bernd Heisele

AI Memo 2003-024
CBCL Memo 232

November 2003

Abstract. We present a component-based approach for recognizing objects under large pose changes. From a set of training images of a given object we extract a large number of components which are clustered based on the similarity of their image features and their locations within the object image. The cluster centers build an initial set of component templates from which we select a subset for the final recognizer.

In experiments we evaluate different sizes and types of components and three standard techniques for component selection. The component classifiers are finally compared to global classifiers on a database of four objects.

1 Introduction

Component- or part-based approaches to object detection and recognition have become increasingly popular in the computer vision community over the last couple of years, see e.g. [1–5]. All these approaches have in common that they use some kind of local image features for classification. In the following we list the main reasons for using components of an object rather than the whole object pattern for recognition: (a) Small components usually vary less under pose changes than the image pattern belonging to the whole object. (b) Component-based approaches can handle partial occlusions better than global approaches. An example of robust component-based pedestrian detection in the presence of partial occlusion was presented in [5]. (c) A major problem in detection is the variation in the appearance of different objects belonging to the same class. For example, a car detector should be able to recognize SUVs as well as sports cars. Building a detector based on components which are visually similar across all objects of a class might solve this problem. In the case of cars, these components might be the wheels, the head- and the taillights.

The main question arising in component-based systems is how to choose the components given a set of training images. In [5] the components for detecting people were manually selected. A method for learning the size and shape of facial components by minimizing a bound on the classification error of the individual component classifiers was proposed in [2] for face detection. In [1] gray value patches of various sizes are cropped at random locations in the training images of an object. A subset of the cropped patches is selected based on mutual information. A popular strategy to automatically determine an initial set of components is to apply a generic interest operator to the training images and to pick components located in the vicinity of the detected points of interest [3, 4, 6]. In [4] this initial set was subsequently reduced by selecting components based on mutual information and likelihood ratio. Using interest operators as a preprocessing step has the advantage of being able to quickly and reliably locate component candidates in a given input image. However, interest operators have a tendency to locate points on the boundaries of objects, leading to components which include parts of the background. Furthermore, forcing the locations of the components to coincide with the points detected by the interest operator considerably restricts the choice of possible components–important components for classification might be lost.

We adopt the component-based classification architecture suggested in [2] for object detection. It consists of two levels of classifiers; component classifiers on the first level and a single combination classifier on the second level. The component classifiers are trained to locate the components and the combination classifier perform the final detection based on the maximum outputs of the component classifiers. In contrast to [2] where SVMs were used on both levels, we use component templates and normalized correlation for detecting the components and a linear classifier to combine the results of the normalized correlation. In a first step we extract a large number of components from the object images and cluster them based on the similarity of their image features and their locations within the image. The resulting cluster centers build an initial set of component templates from which we select a subset using Adaboost, stepwise forward selection, and a selection method based on the individual performance of the component templates. In experiments we analyze how the size of the components, the number of clusters, the features computed on the components, and the selection method affect the performance of the classifier. The systems were evaluated on four objects which had to be recognized under varying pose and illumination.

The outline of the paper is as follows: Section 2 briefly describes the architecture of the classification system. In Section 3 we explain how we determined the set of component templates. Section 4 contains experimental results and a comparison between the global and component-based approaches. Section 5 concludes the paper.

2 System Architecture

An overview of our two-level component-based classifier is shown in Fig. 1. On the first level, component classifiers independently detect components of the object. Each component classifier consists of a single component template which is matched against the image within a given search region using normalized correlation. For each component we propagate the maximum value of the normalized correlation to the combination classifier. The combination classifier is linear and produces a binary recognition result which classifies the input image as either belonging to the background or to the object class.

3 Selecting Components

Our training data consisted of images of four different objects which were manually extracted from larger images. All training images belonging to the same object were identical in width and height. We treated the recognition problem for each of the four objects separately, i.e. we trained four recognition systems independently of each other. In the following we describe the method for training the recognizer on a single object.

As shown in Fig. 1 we divided the image into twenty non-overlapping search regions by a 5×4 grid. For each of the object images in the training set and for each search region we extracted 100 squared patches of fixed size whose centers were randomly placed within the corresponding search region. We then performed a data reduction step by applying k-means clustering to all components belonging to the same search region. The resulting cluster centers built our initial set of component templates. The search regions define a simple geometrical model of the object, in which the prior of finding a component within its corresponding search region is uniform, and the prior of finding it outside is 0. The intuition behind this model is that under small pose changes a given part of the object is projected into the same search region. In the experiment

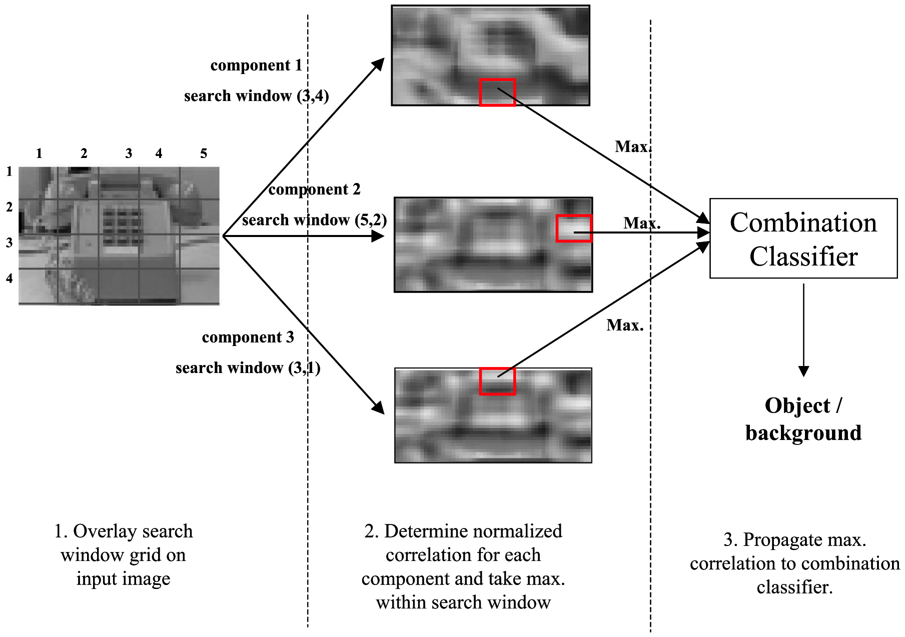
component 1
search window (3,4)
component 2
search window (5,2)
component 3
search window (3,1)
1 2 3 4 5
1
2
3
4
Max.
Max.
Max.
Combination Classifier
Object / background
1. Overlay search window grid on input image
2. Determine normalized correlation for each component and take max. within search window
3. Propagate max. correlation to combination classifier.

Fig. 1. The component-based architecture: On the first level, the component templates are matched against the input image within predefined search regions using normalized correlation. Each component's maximum correlation value is propagated to the combination classifier which performs the final recognition.

section we will indeed show that searching for components over the whole image is worse than limiting the correlation to search regions.

For each component template we build a corresponding component classifier which returns a single output value for every training image. This value is computed as the maximum of the normalized correlation between the component template and the input image within the search region. The next step is to pick a subset from the large number of available component templates (classifiers). To do this we added a negative training set containing of non-object images. The negative training images had the same width and height as the object images.

We evaluated the following techniques for selecting a given number of M component templates: (a) We generated ROC curves for each component classifier on the training set and ranked the components according to decreasing ROC area. The combination classifier, which was a linear SVM, was trained on the output values of the first M component classifiers according to above ranking. (b) We selected the components by M iterations of Adaboost for decision stumps [7]. (c) We applied M iterations of forward stepwise regression [8] to the set of component classifiers. We assigned labels $+1$ to object images and -1 to negative images to build a label vector $\mathbf{y}_1$. Each component classifier j provided us with a vector of output values $\mathbf{x}_j$, where output value was within $[-1, 1]$. For each component classifier we computed a linear regression of the labels, i.e. approximating $\mathbf{y_1}$ by $a_j\mathbf{x}_j + b_j$. The component k which gives the smallest quadratic error is chosen. We then computed the residual $\mathbf{y_2} = \mathbf{y_1} - a_k\mathbf{x}_k + b_k$ and used it as our new label vector for the second iteration. This was continued M times and yielded a linear regression function $f(\mathbf{x}) = \sum a_n\mathbf{x}_n + \sum b_n$, where the sum is over the M selected components. Note, that the same component can be picked multiple times in this procedure.

4 Experiments

4.1 Training and test sets

To generate the positive training and test data we first manually cropped the objects from 640 × 480 gray images. The aspect ratio of the cropping window was kept constant for each object but varied between the four different objects. Then we scaled the cropped images to a fixed size, preserving the aspect ratio. The negative training and test sets consisted of image parts cropped at random locations of non-object images. Some examples of training and test images of the four objects are shown in Fig. 2. Note the large variations in pose and illumination.

The phone image set included 34 object training images and 114 object test images of size 69 × 40 pixels. For the coffee machine we used 54 training and 87 test images of size 51 × 40; for the bird we had 32 training images and 131 test images of size 49 × 40. Finally, 44 training images and 157 test images were of size 89 × 40 were used for the fax machine. For all objects we used randomly selected 4000 non-object training images and 9000 non-object test images. We

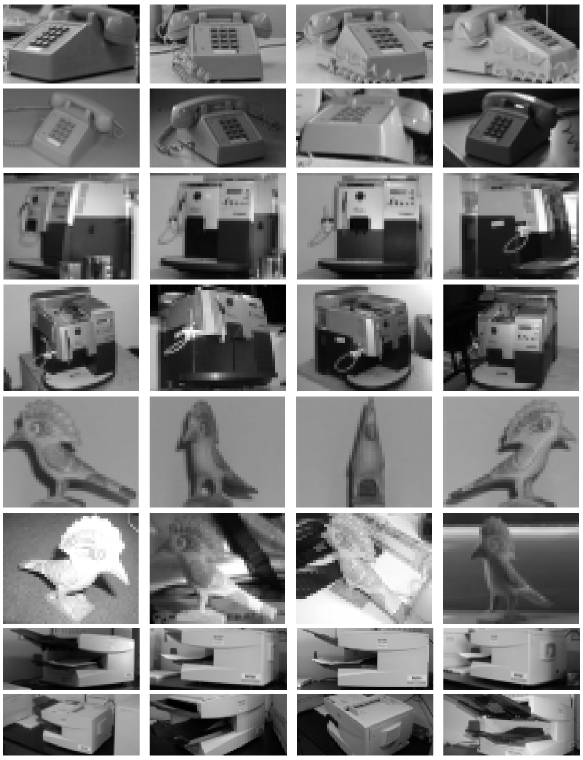

Fig. 2. Examples of training and test images for the four objects. The top row for each object shows the training images, the bottom row the test images.

computed two different types of features in our component templates: raw gray values and the magnitude of the gradient, computed by convolving the image with the derivative of a 2D-Gaussian ($\sigma = 1$); two examples of which are shown in Fig. 3.

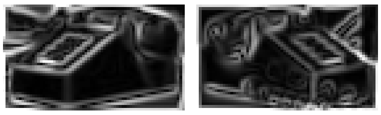

Fig. 3. Magnitude of the gradient computed on two example images of the training set.

The following results, with exception of the experiments at the end of the section, were first computed on the telephone data. The same experiments were later run again on the coffee-machine data.

4.2 Search regions

As mentioned before, we divided the images into twenty non-overlapping search regions using a 5×4 grid. We generated ROC curves for two systems, one using search regions and one in which the maximum output of the component classifiers was computed over the full image. As we can see in Fig. 4 the system with search regions is clearly better.

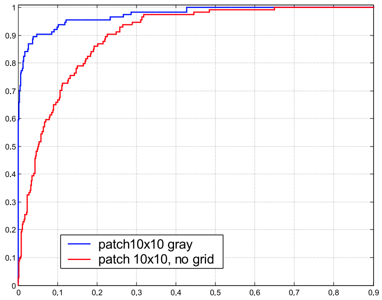

Fig. 4. ROC curves for systems with and without search regions.

4.3 Number of components per search region

One of the first parameters to choose, is the number of clusters per search region, which determines the initial number of components. This number should be large enough to capture the variations of the image pattern within a given search region across the training data. Since we recorded the objects over a large range of views, different parts of the object get projected into the same search region. If the cluster number is too small, these parts will be grouped into the same cluster. We experimented with $k = 5$, 10, 15, 20, 30 and 50 clusters per search window.

As described in the previous section we used normalized correlation to compute an ROC curve on the training data for every component. For each of the 20 search regions we then determined the best single component based on its ROC area and then computed the average ROC area over the 20 best components. The final average ROC value area steadily increased for our values of k up to $k = 30$, but then stayed approximately the same for $k = 50$. In all following experiments we therefor used 30 clusters per search region, resulting in an initial set of 600 components.

4.4 Size of Components

We trained four classifiers for components of size 3 × 3, 5 × 5, 10 × 10 and 15 × 15 pixels. The ROC curves are shown in Fig. 5. All four classifiers used 30 components selected by Adaboost. As can be seen in in Fig. 5, the 3 × 3 components seem to be too small to capture object specific information and will therefor not be further considered in the following experiments. The components larger than 5 × 5 perform similarly.

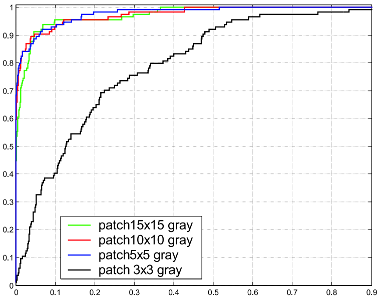

Fig. 5. ROC curves for a raw gray-value classifier with different component sizes

4.5 Selection methods

As a baseline measure for the following experiments on component selection we randomly selected components from the original set of 600 components, trained a linear SVM on their outputs and generated ROC curves on the test set (see

Fig. 6). We then implemented three selection methods: Adaboost, stepwise forward selection and a selection technique based on the ROC area of the component classifiers. For both Adaboost and forward selection, a linear combination classifier is computed during the selection process. For the third method we had to train a linear SVM on the continuous outputs of the selected component classifiers. The ROC curves are shown in Fig. 6. Each diagram shows the comparison between the three selection methods for components of size 5 × 5, 10 × 10 and 15 × 15. All three methods clearly outperform random selection, as was expected. Adaboost is marginally better than the selection based on ROC area. Forward selection was in two of the three experiments clearly inferior to both other techniques. In the remaining experiments we only used Adaboost.

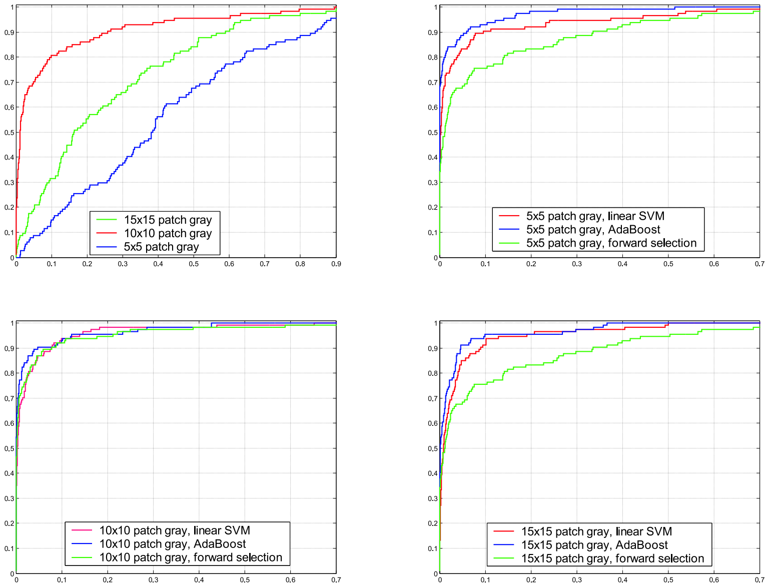

Fig. 6. The ROC curves for randomly selected components is shown in the top left diagram; The other diagrams show the ROC curves for the three selection algorithms for different component sizes.

4.6 Image Features

So far we have only used gray value components. As mentioned at the beginning of the section, we also computed components from gradient images. The comparison between using gray value components, gradient components and the combination of both is shown in Fig. 7. We selected 100 components using Adaboost and computed the ROC curves for the telephone data set.

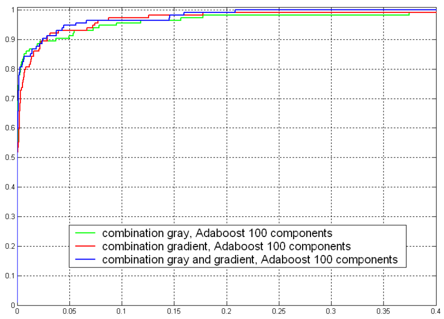

Fig. 7. Comparison between three systems using gray value components, gradient components and the combination of both.

The best performance was thus achieved by combining both gray and gradient templates into a big pool of 3600 components and selecting the best templates out of it.

4.7 Number of selected components

We combined the gray and gradient components of size 5×5, 10×10 and 15×15 to build a set of 3600 components, from which we selected between 10 and 400 components with Adaboost. Above around 400 components the performance of the classifier did not improve significantly, adding further components seemed did not always justify the increase in computational complexity.

4.8 Final experiments

To have a baseline system to compare to, we trained linear and Gaussian SVMs with different values of σ, on the whole, image of every object. Preprocessing the images with histogram equalization had negative effects on the performance of the global SVMs. Thus we used for the comparison the best performing SVM without using histogram equalization. Fig. 8 shows the ROC curves for the four different objects for the component-based system and the global classifier. Except for the fax machine, where both systems were practically on par, the component based system is clearly better.

Both systems had problems recognizing the bird. This can be explained by the strong changes in the silhouette of the figure under rotation. Since we extracted the object images with a fixed aspect ratio, some of the training images of the bird contained a significant amount of background. Adding the fact that the background was the same on all training images but was different on the test images, the relatively poor performance is not surprising.

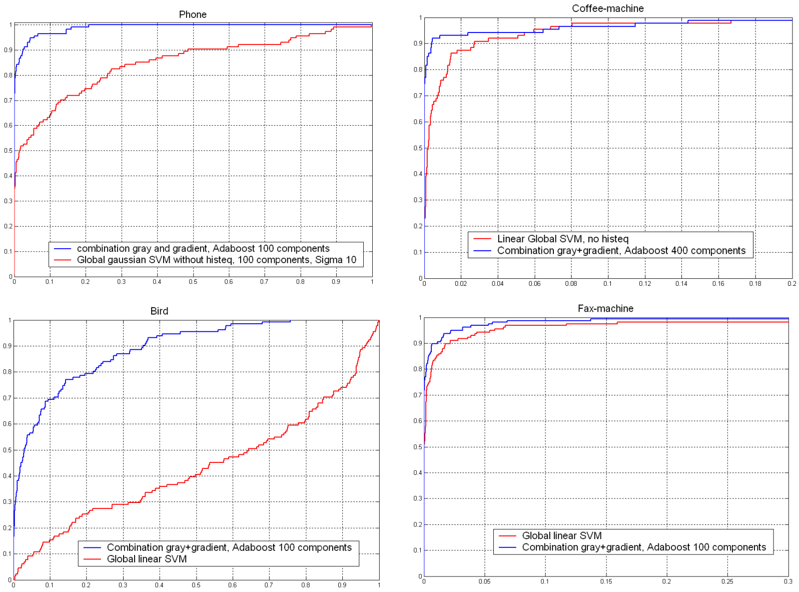

Fig. 8. Final identification results for the 4 different objects in comparison to the performance of a global classifier.

5 Conclusion

We presented a component-based approach for recognizing objects under large pose changes. From a set of training images of a given object we extracted a large number of components which are clustered based on the similarity of their image features and their locations within the object image. The cluster centers built an initial set of component templates which were matched against the image within predefined search regions. From the initial set of components we selected a subset using Adaboost, stepwise forward selection and a method based on the ROC area of the individual component classifiers.

We conducted experiments with different sizes, numbers and types of components. In conclusion, between 100 and 400 gray value and gradient filtered components selected from an initial set of 3600 with sizes between 5×5 and 15×15 performed best. Amongst the selection methods, Adaboost was marginally better than the selection based on ROC area and consistently outperformed stepwise forward selection. We compared the component-based classifier to a global, nonlinear classifier on a database of four objects. For three objects the component system was significantly better and it was about on par with the global classifier for the fourth object.

References

1. Ullman, S., Sali, E.: Object classification using a fragment-based representation. In: Biologically Motivated Computer Vision (eds. S.-W. Lee, H. Bulthoff and T. Poggio). (2000) 73–87 (Springer, New York)
2. Heisele, B., Serre, T., Pontil, M., Vetter, T., Poggio, T.: Categorization by learning and combining object parts. In: Neural Information Processing Systems (NIPS), Vancouver (2001)
3. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2003)
4. Dorko, G., Schmid, C.: Selection of scale invariant neighborhoods for object class recognition. In: International Conference on Computer Vision. (2003)
5. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume 23. (April 2001) 349–361
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (submitted 2003)
7. Schapire, R., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: A new explanation of effectiveness of voting methods. The Annals of Statistics **26** (1998) 1651–1686
8. Weisberg, S.: Applied Linear Regression. Wiley, New York (1980)