Abstract

We describe an algorithm for automatically learning discriminative components of objects with SVM classifiers. It is based on growing image parts by minimizing theoretical bounds on the error probability of an SVM. Component-based face classifiers are then combined in a second stage to yield a hierarchical SVM classifier. Experimental results in face classification show considerable robustness against rotations in depth and suggest performance at significantly better level than other face detection systems. Novel aspects of our approach are: a) an algorithm to learn component-based classification experts and their combination, b) the use of 3-D morphable models for training, and c) a maximum operation on the output of each component classifier which may be relevant for biological models of visual recognition.

1 Introduction

We study the problem of automatically synthesizing hierarchical classifiers by learning discriminative object parts in images. Our motivation is that most object classes (e.g. faces, cars) seem to be naturally described by a few characteristic parts or components and their geometrical relation. Greater invariance to viewpoint changes and robustness against partial occlusions are the two main potential advantages of component-based approaches compared to a global approach.

The first challenge in developing component-based systems is how to choose automatically a set of discriminative object components. Instead of manually selecting the components, it is desirable to learn the components from a set of examples based on their discriminative power and their robustness against pose and illumination changes. The second challenge is to combine the component-based experts to perform the final classification.
2 Background

Global approaches in which the whole pattern of an object is used as input to a single classifier were successfully applied to tasks where the pose of the object was fixed. In [6] Haar wavelet features are used to detect frontal and back views of pedestrians with an SVM classifier. Learning-based systems for detecting frontal faces based on a gray value features are described in [14, 13, 10, 2].

Component-based techniques promise to provide more invariance since the individual components vary less under pose changes than the whole object. Variations induced by pose changes occur mainly in the locations of the components. A component-based method for detecting faces based on the empirical probabilities of overlapping rectangular image parts is proposed in [11]. Another probabilistic approach which detects small parts of faces is proposed in [4]. It uses local feature extractors to detect the eyes, the corner of the mouth, and the tip of the nose. The geometrical configuration of these features is matched with a model configuration by conditional search. A related method using statistical models is published in [9]. Local features are extracted by applying multi-scale and multi-orientation filters to the input image. The responses of the filters on the training set are modeled as Gaussian distributions. In [5] pedestrian detection is performed by a set of SVM classifiers each of which was trained to detect a specific part of the human body.

In this paper we present a technique for learning relevant object components. The technique starts with a set of small seed regions which are gradually grown by minimizing a bound on the expected error probability of an SVM. Once the components have been determined, we train a system consisting of a two-level hierarchy of SVM classifiers. First, component classifiers independently detect facial components. Second, a combination classifier learns the geometrical relation between the components and performs the final detection of the object.

3 Learning Components with Support Vector Machines

3.1 Linear Support Vector Machines

Linear SVMs [15] perform pattern recognition for two-class problems by determining the separating hyperplane with maximum distance to the closest points in the training set. These points are called support vectors. The decision function of the SVM has the form:

\[ f(x) = \sum_{i=1}^{\ell} \alpha_i y_i < x_i \cdot x > + b, \quad (1) \]

where \( \ell \) is the number of data points and \( y_i \in \{-1, 1\} \) is the class label of the data point \( x_i \). The coefficients \( \alpha_i \) are the solution of a quadratic programming problem. The margin \( M \) is the distance of the support vectors to the hyperplane, it is given by:

\[ M = \frac{1}{\sqrt{\sum_{i} \alpha_i}} \quad (2) \]

The margin is an indicator of the separability of the data. In fact, the expected error probability of the SVM, \( EP_{err} \), satisfies the following bound [15]:

\[ EP_{err} \leq \frac{1}{\ell} E \left[ \frac{D^2}{M^2} \right], \quad (3) \]

where \( D \) is the diameter of the smallest sphere containing all data points in the training set.
3.2 Learning Components

Our method automatically determines rectangular components from a set of object images. The algorithm starts with a small rectangular component located around a pre-selected point in the object image (e.g. for faces this could be the center of the left eye). The component is extracted from each object image to build a training set of positive examples. We also generate a training set of background patterns that have the same rectangular shape as the component. After training an SVM on the component data we estimate the performance of the SVM based on the upper bound on the error probability. According to Eq. (3) we calculate:

\[ \rho = \frac{D^2}{M^2}. \] (4)

As shown in [15] this quantity can be computed by solving a quadratic programming problem. After determining \( \rho \) we enlarge the component by expanding the rectangle by one pixel into one of the four directions (up, down, left, right). Again, we generate training data, train an SVM and determine \( \rho \). We do this for expansions into all four directions and finally keep the expansion which decreases \( \rho \) the most. This process is continued until the expansions into all four directions lead to an increase of \( \rho \). In order to learn a set of components this process can be applied to different seed regions.

4 Learning Facial Components

Extracting face patterns is usually a tedious and time-consuming work that has to be done manually. Taking the component-based approach we would have to manually extract each single component from all images in the training set. This procedure would only be feasible for a small number of components. For this reason we used textured 3-D head models [16] to generate the training data. By rendering the 3-D head models we could automatically generate large numbers of faces in arbitrary poses and with arbitrary illumination. In addition to the 3-D information we also knew the 3-D correspondences for a set of reference points shown in Fig. 1a). These correspondences allowed us to automatically extract facial components located around the reference points. Originally we had seven textured head models acquired by a 3-D scanner. Additional head models were generated by 3-D morphing between all pairs of the original head models. The heads were rotated between \(-30^\circ\) and \(30^\circ\) in depth. The faces were illuminated by ambient light and a single directional light pointing towards the center of the face. The position of the light varied between \(-30^\circ\) and \(30^\circ\) in azimuth and between \(30^\circ\) and \(60^\circ\) in elevation. Overall, we generated 2,457 face images of size 58×58. Some examples of synthetic face images used for training are shown in Fig. 1b).

The negative training set initially consisted of 10,209 58×58 non-face patterns randomly extracted from 502 non-face images. We then applied bootstrapping to enlarge the training data by non-face patterns that look similar to faces. To do so we trained a single linear SVM classifier and applied it to the previously used set of 502 non-face images. The false positives (FPs) were added to the non-face training data to build the final training set of size 13,654.

We started with fourteen manually selected seed regions of size 5×5. The resulting components were located around the eyes (17×17 pixels), the nose (15×20 pixels), the mouth (31×15 pixels), the cheeks (21×20 pixels), the lip (13×16 pixels), the nostrils (22×12 pixels), the corners of the mouth (18×11 pixels), the eyebrows (19×15 pixels), and the bridge of the nose (18×16 pixels).