A Neural 3-D Object Recognition Architecture Using Optimized Gabor Filters

Gunther Heidemann and Helge Ritter
Institut für Neuroinformatik
Universität Bielefeld
33501 Bielefeld, Germany
email: gheidema@techfak.uni-bielefeld.de

Abstract

We present an object recognition architecture based on feature extraction by Gabor filter kernels and feature classification by an artificial neural network. The parameters of the Gabor filters are optimized to the specific problem by minimizing an energy function. Such Gabor filters extract features that can be more easily classified by the neural network. Moreover the feature space is low-dimensional, so feature extraction doesn’t require much computational effort. The object recognition system is implemented on a Datacube and works in real-time.

1. Introduction

In the field of object recognition, many current applications are strongly based on explicit knowledge about the specific recognition task like the shapes of the objects or illumination conditions. However, the use of explicit knowledge should be reduced as much as possible for mainly two reasons. First, the acquisition of explicit knowledge is still mainly done by the human programmer and soon becomes unacceptably high if complex objects are to be modelled. Second, explicit knowledge like geometric relations is often not of much use in object recognition (though of course in other fields like 3D-reconstruction), since the appearance of an object is strongly determined by its surface reflectance properties, especially under natural illumination conditions. Therefore, an object recognition architecture seems reasonable that is able to acquire at least part of the knowledge needed for recognition by the presentation of examples.

Artificial neural networks (ANN) have the ability to learn classification tasks from examples. However, the use of ANNs in the domain of object recognition depends crucially on the “quality” of the feature vector extracted from the images to be classified. Feature extraction should deliver a low-dimensional and “easy to classify” vector, which means robustness (if not invariance) against affine transformations, changes in illumination or noise. Otherwise, the network required for the classification task became too large and needed too much computational resources, and, even worse, required a huge set of training examples.

Though it would be extremely hard to find a feature set that meets the requirements of low dimensionality, high specificity and robustness in general, it may be possible to generate feature detectors that do the job at least for a given problem. However, here again arises the danger of involving explicit knowledge about the specific domain by tuning the feature detectors. Therefore it would be desirable to train not only the classifying ANN but also the feature extraction itself by examples.

In this paper we present an architecture for object classification based on features extracted by Gabor filter kernels and their classification by an ANN of the Local Linear Map – type (LLM). The architecture was motivated by the works on hand posture recognition by Drees and Ritter [3, 1]. As a first step towards the required generation of feature detectors from examples, we define an energy function on the parameters of the Gabor filter kernels that measures the “classificability” of the feature vector. So part of the classification task is already done by the feature detectors and a relatively small LLM is sufficient. Therefore, only a relatively small set of training images is required. Because of the low computational effort the system is suitable for real-time applications. It is implemented on a Datacube system.

2. The Vision Architecture

Our system performs object recognition in three steps:

1. Search for regions of interest
2. Feature extraction
3. Classification

The architecture is shown schematically in Fig. 1. From the camera we get a HSV-image. The intensity channel is
processed separately from the two colour channels. The colour channels are used for a simple colour segmentation for colours of interest. The colour segmentation is a local transformation performed by a lookup table. Its result is a binary image (colour of interest – no colour of interest). After smoothing the binary image, a connectivity analysis yields the centers of the blobs which meet certain connectivity and size conditions.

For feature classification we chose a Local Linear Map – network, which is a computationally efficient means for nonlinear function approximation. It is related to the self-organizing map [2] and the GRBF [4, 5, 7] approach. It approximates the nonlinear function by a set of locally valid linear mappings, for details see e.g. [6]. Here we use a “winner takes all” network. In this case for a given input only one node, the best match or “winner” node, contributes to the output vector.
For the classification task, we need a mapping from an \( n_G \) -dimensional input space to an \( n_G + 1 \) -dimensional output space, \( \mathbb{R}^{n_G} \rightarrow \mathbb{R}^{n_G+1} \). Let \( \vec{x} \) be the input vector, then the output vector \( \vec{y} \) of the network is given by
\[
\vec{y} = \vec{w}^{\text{out}}_k + g \cdot A_k (\vec{x} - \vec{w}^{\text{in}}_k),
\] (5)
where \( k \) is the winner node, \( k \in \{1 \ldots n_N\} \), \( n_N \) is the number of nodes, \( \vec{w}^{\text{in}}_k \) and \( \vec{w}^{\text{out}}_k \) are the input and output weight vectors of node \( k \), respectively. \( A_k \) is a \( (n_O + 1) \times n_G \) matrix associated with node \( k \), \( g \geq 0 \) determines the ratio of \( \vec{w}^{\text{out}}_k \) and \( A_k \) contributing to the output. The best match node \( k \) is determined by
\[
k = \arg \min_{k'} (||\vec{x} - \vec{w}^{\text{in}}_{k'}||), \quad k' = 1 \ldots n_N, \quad (6)
\]
If there are \( n_T \) correct input-output pairs, \( (\vec{x}^{(\alpha)}, \vec{y}^{(\alpha)}), \alpha = 1, 2, \ldots, n_T \), the network is trained supervised, using the following adaptation equations
\[
\Delta \vec{w}^{\text{in}}_k = \epsilon_\text{in} \cdot (\vec{x}^{(\alpha)} - \vec{w}^{\text{in}}_k), \quad (7)
\]
\[
\Delta \vec{w}^{\text{out}}_k = \epsilon_\text{out} \cdot (\vec{y}^{(\alpha)} - \vec{y}(\vec{x}^{(\alpha)})) + g \cdot A_k \Delta \vec{w}^{\text{in}}_k, \quad (7)
\]
\[
\Delta A_k = \epsilon_A \cdot (\vec{y}^{(\alpha)} - \vec{y}(\vec{x}^{(\alpha)})) (\vec{x}^{(\alpha)} - \vec{w}^{\text{in}}_k)^2,
\]
with the adaptation stepsizes \( \epsilon_\text{in}, \epsilon_\text{out}, \epsilon_A \in [0, 1] \). For training the classification task the output vectors have the form
\[
y_l^{(i)}(t) = \delta_{tl}, \quad \text{with} \quad i, l = 1 \ldots n_O + 1, \quad (8)
\]
where \( l \) is the class of the object to be trained.

4. Optimizing feature extraction by adaptation of Gabor filters

The “quality” of the feature vector extracted by the Gabor filter kernels is crucial for the performance of the system. As outlined above, the feature vector should on the one hand be low-dimensional but on the other hand each object class should be represented in a highly specific way. So even a small LLM should be able to yield a good classification performance. By the method outlined here, the parameters of a given set of feature detectors can be optimized such that classification performance improves. We assume that the following properties of the feature vectors lead to high specificity:

- Feature vectors of the same object class cluster closely in feature space
- Clusters of feature vectors of different object classes are separated from each other as far as possible

From this requirements an energy function can be constructed, the minimization of which should yield the desired transformation of the feature space (shown idealized in Fig. 2). Its value is the smaller the better the desired clustering in feature space is achieved:
\[
E = c_\text{cluster} E_\text{cluster} - c_\text{inter cluster} E_\text{inter cluster} \quad (9)
\]
with
\[
E_\text{cluster} = \sum_{s=1}^{n_O+1} \sum_{i=1}^{n_I(o)} \text{dist} (\vec{x}^G(o,i), S^G(o)) \quad (10)
\]
\[
E_\text{inter cluster} = \sum_{o_1=1}^{n_O+1} \sum_{o_2=n_O+1}^{n_O+1} \text{dist}_2 (S^G(o_1), S^G(o_2)) \quad (11)
\]
\[
S^G(o) = \frac{1}{n_I(o)} \sum_{i=1}^{n_I(o)} \vec{x}^G(o,i). \quad (12)
\]
Here \( n_I(o) \) is the number of training images of object \( o \). \( \vec{x}^G(o,i) \) is the feature vector for training image \( i \) of object \( o \). \( S^G(o) \) is the “center of mass” of the cluster of feature vectors belonging to object \( o \). Superscript \( G \) is meant to indicate the dependance of all feature vectors and centers of mass on the Gabor parameters (cf. Eqs. 1, 2). \( \text{dist} \) and \( \text{dist}_2 \) are distance measures in the \( n_G \) -dimensional feature space. In the simplest case, which was adopted here, the euclidean distance can be used. Then the first term of Eq. 9 corresponds to the requirement of good clustering of the feature vectors of one object class, whereas the second term is aimed to separate the cluster centers as far as possible. The results reported below were obtained with the choice \( c_\text{cluster} = c_\text{inter cluster} = 1 \).

\( E \) is a function of the set of all Gabor parameters \( \vec{\alpha}_i, \vec{\sigma}_i, \vec{k}_i, \phi_i, i = 1 \ldots n_G \). Therefore, even for moderate numbers of \( n_G \), Eqs. 9-12 lead to a minimization problem
in a rather high-dimensional space. Due to the nonlinear dependence of each Gabor function on its parameters, \( E \) becomes a rather complex function. Minimization of \( E \) with respect to the parameters of the Gabor kernels approves to be difficult because of the high complexity of the energy surface. It turns out that \( E \) has numerous local minima. Standard methods as conjugate gradient descent and simulated annealing as a rule don’t find the global minimum. Fortunately, it turns out that also local minima may lead to satisfactory results.

5. Results

The proposed architecture has been investigated so far in the context of the task of the recognition of a set of wooden toy pieces ("Baufix"), which are a "wheel", a "cube", a "rhomb-nut" and a "ring". Further, there are two types of "screws", each in four different lengths. These eight different screws were considered as one object class, so there are in total five object classes. The parts are presented to the camera lying on a table, see Fig. 3. The objects may be freely arranged within the range of the table from where the training images were taken as long as there is no occlusion. As a training set 50 images of each part were used, for the screws 200 images were used. On the training images the parts are arranged in different views and distances from the camera. By this way rotational invariance and scaling up to 30% were trained.

20% of the training set was used for the optimization of the Gabor filter kernels. As a starting configuration we chose 16 Gabor filters with vanishing wave-vector, arranged on a \( 4 \times 4 \) grid. Each mask is a square of \( 9 \times 9 \) pixels. Local optimization of the associated function \( E \) (Eq. 9) by conjugate gradient descent transformed the configuration to one with much better classification properties (see Fig. 4). The misclassification rate was reduced from 35% to 20% for all blobs found by the colour segmentation. The current implementation on a Datacube MaxVideo200 with DIGICOLOR reaches a frame rate of about 4 frames per second for a \( 250 \times 250 \) pixel image containing 15 objects.

For training the LLM the whole training set was used. A number of 20 nodes approved to be the optimum. Though a higher number of nodes gives a still better approximation of the training data, the higher specialization of the network decreases its generalization properties and leads to worse classification on the test data.

The rejection class used so far is not universal, i.e. only objects that were trained to be rejected are classified correctly. Completely unknown objects, however, may only by chance fall into the rejection class but as well be classified as one of the known objects. Therefore the terminal is misclassified in Fig. 3. Misclassifications can also occur if two objects lie too close together as the screw and the rhomb-nut in the upper half of Fig. 3. In this case the colour segmentation delivers only one blob for the two objects which can’t
be classified correctly. These problems remain to be solved.

6. Conclusion

We have presented an easy to implement architecture for object recognition. Computational efficiency and simple acquisition of (implicit) knowledge from examples could be achieved by using an adaptive system not only in the classification part (ANN) but also for the feature extraction. First results are promising, the classification rate can be significantly improved by optimizing the feature detectors.

Our method differs from the approach of training networks on recognition tasks directly on the pixel data, because we train the feature extraction and the classification part separately. Though e.g. a multilayer perceptron using the backpropagation algorithm can in principle solve the same task, our approach is more efficient. In contrast to a multilayer perceptron, which forms its “feature detectors” in the first layer, we can influence this process directly by the choice of the energy function, therefore feature extraction becomes efficient and the net for classification may be small.

Further investigation will concentrate on testing other energy functions and their linear combination with appropriate weighting. Also suitable starting parameters for the Gabor functions have to be found. Moreover, using other types of functions seems to be promising, a first step could be the combination of several Gabor functions to one feature detector.

References


