Fast Object and Pose Recognition Through Minimum Entropy Coding

Günter Westphal
Institut für Neuroinformatik
Ruhr-Universität Bochum
D-44780 Bochum, Germany
guenter.westphal@neuroinformatik.rub.de

Rolf P. Würtz
Institut für Neuroinformatik
Ruhr-Universität Bochum
D-44780 Bochum, Germany
rolf.wuertz@neuroinformatik.rub.de

Abstract

We present a pattern recognizer to classify a variety of objects and their pose on a table from real world images. Learning of weights in a linear discriminant is based on estimating the relative information contributed by a set of features to the final decision. Evaluation of the discriminant is very fast, allowing for about three decisions per second on datasets without segmentation difficulties like the COIL-100 database. Experiments on that database yield high recognition rates and good generalisation over pose.

1. Introduction

Analysis of a visual scene relies on (bottom up) feature calculation as well as (top-down) object knowledge. While there are many methods for feature extraction the knowledge-driven aspect in machine vision suffers from the difficulty of applying knowledge about thousands of objects more or less instantaneously to a given scene. Neural networks are good candidates because they usually combine a slow (offline) training phase with very rapid evaluation. They have difficulties with the many invariances to be dealt with like movements in the image plane, 3D-movement, deformation, partial occlusion, and changes in lighting. Some of these are typically normalized in the feature extraction phase, although feature extraction can theoretically also be formulated as additional network layers [2].

Attempts to use the appearance rather than the shape of objects for classification go back to [3], their COIL-100 database has become a benchmark for combined object/pose classification for sufficiently segmented objects. It consists of 100 objects on a turntable with rotation angles 4° apart. It has been observed that a multilayer neural network with winner-take-all nonlinearity does a good job at recognizing objects [4].

In this paper we study the capability of a single layer perceptron to classify objects and measure their pose simultaneously. Weights are adjusted according to the contribution of the features' values to the final decision. Techniques of this sort have been used in linguistic analysis but to our knowledge they have not previously been applied to vision, yet.

2. Method

Our approach is based on information theory [6], which has once been developed for the transmission of encoded signals via a communication channel using as few resources as possible by exploiting the statistical structure of the messages sent. However, for object recognition purposes it is not the information transmitted by each symbol (or feature) that is of importance, but what the patterns in the observations convey about the nature of the information source. As more and more of the feature sequence is observed, more and more information about the data generation process is (hopefully) gained, or, conversely, we may interpret the task of object recognition as a quest for minimum entropy [7].

2.1. Unsupervised Feature Learning

Let $\mathbb{D}$ be a set of D images I_d, where $\mathbb{D}$ is a well-defined subset of some universe of images $\mathbb{I}$ of objects in varying poses. In the following $\mathbb{D}$ serves as a *learning set* and the images I_d serve as *learning examples*. The label d is used to differentiate between the images and ranges from 1 to D throughout.

$$\mathbb{D} = \{\, I_d |\, 1 \le d \le D \} \subseteq \mathbb{I} \tag{1}$$

We now define R not necessarily disjoint sets of features $\mathbb{F}^r$, each containing the results of a function $f^r : \mathbb{I} \mapsto \Omega^r$, called *feature calculator*, applied to all images $I_d \in \mathbb{D}$ resulting in T^r different *features* $f_t^r \in \Omega^r$ where Ω^r is the set of all possible features with respect to feature calculator f^r. A feature calculator shall be an implicitly parameterized function capable of extracting some feature, e.g., the average color, out of the parameterized image. Label r specifies

the feature calculator or the feature set, respectively, and ranges from 1 to R throughout. Label t is used to differentiate between the features in one feature set $\mathbb{F}^r$ and ranges from 1 to T^r throughout.

$$\mathbb{F}^r = \{\, f_t^r = f^r\left(I_d\right) | \, 1 \leq t \leq T^r \wedge I_d \in \mathbb{D}\} \subseteq \Omega^r \quad (2)$$

We are interested in the (subjective) probability that the object to be recognized is the one present in image I_d given that feature f_t^r has been observed. Hence, we are interested in the (posterior) probability

$$\mathcal{P}_t^r\left[I_d\right] := \mathcal{P}\left[\mathcal{I} = I_d | \mathcal{F}^r = f_t^r\right] \quad (3)$$

where $\mathcal{I}$ and $\mathcal{F}^r$ are random variables that the image is I_d or that feature f_t^r has been observed, respectively. To avoid notational clutter, we will omit the random variables $\mathcal{I}$ and $\mathcal{F}^r$ whenever possible. Applying Bayes' theorem and claiming that all prior probabilities are equal yields

$$\mathcal{P}_t^r\left[I_d\right] = \frac{\mathcal{P}\left[f_t^r | I_d\right]\mathcal{P}\left[I_d\right]}{\mathcal{P}\left[f_t^r\right]} = \frac{n_{t,d}^r}{\sum\limits_{I_{\tilde{d}} \in \mathbb{D}} n_{t,\tilde{d}}^r} \quad (4)$$

where $n_{t,d}^r$ is the number of occurrences of feature f_t^r in image I_d.

We would like to construct models for the real but unknown (objective) probabilities $\tilde{\mathcal{P}}_t^r$ by using the knowledge distilled out of the learning set. This results in $M = \sum_{r=1}^{R} T^r$ not necessarily different distributions. For this, we calculate the empirical risks $\mathcal{R}_t^r\left[\mathcal{I}\right]$ for each feature f_t^r which becomes the cross entropy between the subjective and objective distributions if $-\ln \tilde{\mathcal{P}}_t^r\left[I_d\right]$ is chosen as the classification loss function [5].

$$\mathcal{R}_t^r\left[\mathcal{I}\right] = -\sum_{\substack{I_d \in \mathbb{D} \\ \tilde{\mathcal{P}}_t^r[I_d] \neq 0}} \mathcal{P}_t^r\left[I_d\right] \ln \tilde{\mathcal{P}}_t^r\left[I_d\right] \quad (5)$$

In this case, the empirical risk becomes minimal, if the Kullback-Leibler distance between the subjective and objective probabilities becomes zero, i.e., $\mathcal{P}_t^r\left[\mathcal{I}\right] = \tilde{\mathcal{P}}_t^r\left[\mathcal{I}\right]$. This yields the Shannon entropy $\mathcal{H}_t^r\left[\mathcal{I}\right]$ with respect to the distribution of feature f_t^r in $\mathbb{D}$, which is a measure of disorder (or uncertainty) the feature exerts on the learning set.

$$\mathcal{H}_t^r\left[\mathcal{I}\right] = -\sum_{\substack{I_d \in \mathbb{D} \\ \mathcal{P}_t^r[I_d] \neq 0}} \mathcal{P}_t^r\left[I_d\right] \ln \mathcal{P}_t^r\left[I_d\right] \quad (6)$$

Based on those entropies we define a measure of information i_t^r for each feature f_t^r.

$$i_t^r = \ln D - \mathcal{H}_t^r\left[\mathcal{I}\right] \quad (7)$$

Those measures of information have the desirable property that they scale proportionally with the importance (or relevance) of the features, i.e., i_t^r attains the maximal value of $\ln D$, if there is exactly one image with feature f_t^r, whereas i_t^r becomes minimal, i.e., 0, if each image in the learning set contains feature f_t^r.

Further, we define R vectorial functions $\underline{\tau}^r$ that map an image $I \in \mathbb{I}$ onto binary vectors of length T^r each. Each vector element is the result of a function $\tau_t^r : \mathbb{I} \mapsto \{0,1\}$ applied to I where τ_t^r returns 1, if feature f_t^r is observed in I and zero otherwise.

$$\underline{\tau}^r : \mathbb{I} \rightarrow \{0,1\}^{T^r} ; \underline{\tau}^r\left(I\right) = \left(\tau_t^r\left(I\right)\right)^\top \Big|_{1 \leq t \leq T^r} \quad (8)$$

Finally, we define R matrices $\underline{\underline{\mathbf{W}}}^r$ of dimensions $(T^r \times D)$ whose elements $w_{t,d}^r$ are equal to i_t^r if image I_d contains feature f_t^r and zero otherwise.

$$\begin{aligned} \underline{\underline{\mathbf{W}}}^r &= \left(\tau_t^r\left(I_d\right) i_t^r\right)|_{1 \leq d \leq D, 1 \leq t \leq T^r} \\ &= \left(w_{t,d}^r\right)\big|_{1 \leq d \leq D, 1 \leq t \leq T^r} \end{aligned} \quad (9)$$

2.2. Simultaneous Object and Pose Recognition

For the object and pose recognition task we use the learning set as a gallery. For this purpose, a vectorial function $\underline{a}$ maps an image I of the object to be recognized onto a vector of *activations* of length D, i.e., we assign one activation a_d to each learning example I_d. Each activation is the sum of all measures of information i_t^r for which in both the presented image I and in the learning example I_d feature f_t^r is present. Note that $\underline{a}$ has the form of R linear equations.

$$\begin{aligned} \underline{a} : \mathbb{I} \rightarrow \left(\mathbb{R}_0^+\right)^D ; \underline{a}\left(I\right) &= \sum_{r=1}^{R} \left(\underline{\underline{\mathbf{W}}}^r\right)^\top \underline{\tau}^r\left(I\right) \\ &= \left(a_d\left(I\right)\right)^\top \Big|_{1 \leq d \leq D} \end{aligned} \quad (10)$$

As a decision rule for the recognition tasks we apply the 'winner-take-all' nonlinearity [4], i.e., the image $\hat{I} \in \mathbb{D}$ with the largest activation, i.e., with the largest sum of measures of information, i.e., with the smallest sum of entropies [7], i.e., with the smallest uncertainty, is most likely the image of the object to be recognized.

$$\hat{I} = I_{\hat{d}} \in \mathbb{D}\big|_{\hat{d} = \arg\max\limits_{d} \{\, a_d(I) | 1 \leq d \leq D\}} \quad (11)$$

Note that the object and its pose are recognized simultaneously rather than in two consecutive tasks like in [3]. The object pose is recognized implicitly by assigning the pose of $\hat{I}$.

2.3. Network Representation

From equation (10) one may conclude that the presented object recognition system can be interpreted as a single layer perceptron with M input neurons, D output neurons and connections $w_{t,d}^r$. Figure 1 shows the resulting network.

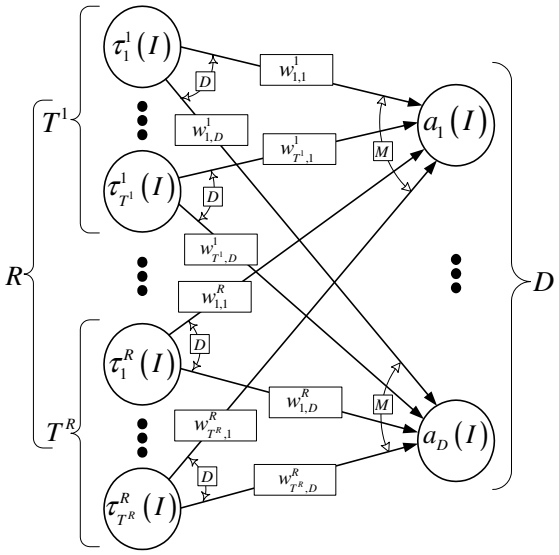


Figure 1. Resulting Single Layer Perceptron

2.4. Feature Calculators

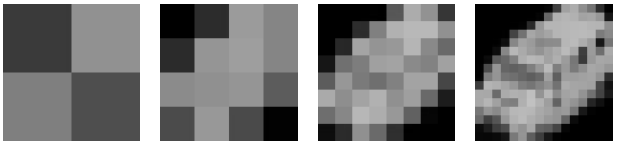
So far, the feature calculators have been defined in a rather abstract fashion. But intuitively, the choice of reasonable feature calculators along with a reasonable parameterization is crucial for the system's functional efficiency [1]. We use three feature calculators with various parameterizations introduced in the following.

2.4.1. Bounding Box Ratio and Orientation This feature calculator places a bounding box around the object and calculates the ratio of the boxes height and width. The bounding box may be unrotated or rotated, the ratio may have up to four decimal places. The orientation of the rotated bounding box is also a part of this feature. An example is shown in Figure 2.

(a) Unrotated (b) Rotated

Figure 2. Bounding Box Ratio and Orientation

2.4.2. Local Average of Grey Level This feature calculator evaluates the average grey level in rectangular image patches. For this, the object in the unrotated bounding box is enlarged to 128×128 pixels. Further, the number of grey levels and the maximal difference between two grey levels when comparing two vectors of grey levels is parameterized. The number of image patches is either 2^2, 4^2, 8^2 or, 16^2, the number of different grey levels is either 5, 12, 25 or, 50 and the maximum difference between two grey levels is either 0, 1, 2, 5 or, 10. An example is shown in Figure 3.



(a) 2^2 Patches (b) 4^2 Patches (c) 8^2 Patches (d) 16^2 Patches

Figure 3. Local Average of Grey Level

2.4.3. Local Average of Color This feature calculator is very similar to the 'Local Average of Grey Level' feature calculator but that it does not operate on the grey level but on the separate RGB channels.

3. Results

Table 1 summarizes the results derived by applying our object recognition system to the COIL-100 image database [3]. The experiments are subdivided into two series. In the first series the local average of grey level and the local average of color feature calculators are parameterized to consider up to 8^2 image patches, whereas in the second series also 16^2 image patches are considered. We constructed nine learning sets with increasing rotation angles (column 'Δ') between two consecutive learning examples resulting in decreasing numbers of learning examples per object (column '$Ex.$'). All remaining images were assigned to the respective testing sets. Note that smaller learning sets lead to larger testing sets, which is a twofold increase in difficulty.

Then we measured the recognition rate (column 'RR') with respect to the testing set and the average time needed for one recognition (column '$\bar{T}$'). For our experiments we used a Pentium IV 2.4 GHz. Further, we analyzed the pose angle of the correctly classified objects with increasing tolerance (columns 'PR_α'), where PR_α means that the object's pose is considered to be correct, if

$$\min\left\{\left|\varphi_{I_d} - \varphi_I\right|, 360^\circ - \left|\varphi_{I_d} - \varphi_I\right|\right\} \leq \alpha\Delta \qquad (12)$$

holds true and φ_I is the true pose angle and φ_{I_d} is the classified one.

	Δ [°]	$Ex.$	RR [%]	$\bar{T}$ [s]	$PR_{0.5}$ [%]	$PR_{1.0}$ [%]	$PR_{1.5}$ [%]	$PR_{2.0}$ [%]
I. Up to 8^2 Patches	10	36	99.22	0.45	91.55	91.55	95.77	95.77
	20	18	98.02	0.38	89.78	93.33	95.09	95.79
	30	12	96.25	0.30	88.10	92.81	94.16	94.49
	40	9	95.11	0.27	82.24	87.10	88.67	89.47
	50	8	93.28	0.27	78.27	84.99	86.52	87.97
	60	6	89.52	0.24	81.60	88.44	90.78	92.11
	70	6	89.38	0.25	74.40	81.12	83.61	85.68
	80	5	86.61	0.24	76.08	83.41	86.51	91.83
	90	4	74.63	0.22	78.50	86.98	92.43	100.00
II. Up to 16^2 Patches	10	36	99.28	1.67	93.90	93.90	97.26	97.26
	20	18	97.98	1.15	91.78	94.69	96.13	96.67
	30	12	96.30	0.94	89.82	93.96	94.95	95.33
	40	9	95.27	0.83	83.39	87.92	89.09	89.70
	50	8	93.02	0.80	79.84	85.69	87.03	88.12
	60	6	89.59	0.68	83.09	89.60	91.75	92.88
	70	6	89.61	0.70	75.09	81.38	83.83	85.78
	80	5	86.79	0.63	77.61	84.13	86.88	91.76
	90	4	75.01	0.55	79.73	87.79	92.59	100.00

Table 1. Results

4. Discussion and Outlook

Although our classification system has a single layer structure performance compares very favorably with other classifiers. In [8] the performance of the original system [3] and a support vector machine is compared with their method of setting up the feature extraction layers in an evolutionary fashion. Their results are about the same as for our system, which has a small advantage between 5 and 12 training views per image and is still better than the SVM for 4 training views per image. The pose accuracy of our system appears rather high, other authors do not usually measure them for the whole object set. The pose errors contain all errors due to pose ambiguity, which are negligible in practice. For robot grasping, the number of misclassified poses is more relevant than the mean pose error. Our method is fast enough to allow for real time recognition and pose estimation.

As single layer perceptrons cannot solve all classification problems these results mean that either the combined object/pose estimation consists of linearly separable classes or that the number of objects is still small. The latter is certainly true compared to the number of objects needed for realistic scene analysis. The scaling of object recognition algorithms to large numbers of objects is thus a very important research question.

It is clear that appearance-based classification schemes rely heavily on prior segmentation. They also have difficulties when objects are deformable. Thus, our recognition system can only be a part of larger system, which takes care of segmentation and handling of other object classes.

Due to the hierarchical structure of the organization of features and objects additional heuristics can be incorporated. The features we have used are not independent, e.g., the average grey value clearly depends on the averaged color channels, and the lower resolution features depend on the higher ones. These dependencies are handled automatically by the statistical weighting of the single features. Therefore, we expect good scaling of the system when more features are added, whose dependencies may be less clear (e.g., shape descriptors and color histograms). As an example for heuristics, after identification of a moving object, dynamic pose estimation can be restricted to that object type. Furthermore, the method is very well suited to pick a set of object/pose combinations with high activations and hand them over to more sophisticated correspondence-based methods, e.g., [9]. The resulting correspondence fields open the possibility of improved pose calculation. First experiments in this direction have been successfully conducted.

References

[1] S. Becker and M. Plumbley. Unsupervised Neural Network Learning Procedures For Feature Extraction and Classification. *Journal of Applied Intelligence*, 6:185–203, 1996.

[2] C. M. Bishop. *Neural networks for pattern recognition*. Clarendon Press, Oxford, England, 1995.

[3] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[4] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[5] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, Cambridge, Massachusetts, London, England, 2002.

[6] C. E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[7] S. Watanabe. Pattern Recognition as a Quest for Minimum Entropy. *Pattern Recognition*, 13(5):381–387, 1981.

[8] H. Wersing and E. Körner. Learning optimized features for hierarchical models of invariant object recognition. *Neural Computation*, 15(7):1559–1588, 2003.

[9] R. P. Würtz. Object recognition robust under translations, deformations and changes in background. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):769–775, 1997.