# Object Recognition with Informative Features and Linear Classification

Michel Vidal-Naquet    Shimon Ullman

*Faculty of Mathematics and Computer Science*
*The Weizmann Institute of Science*
*Rehovot 76100, Israel*
*e-mail: michel.vidal-naquet@weizmann.ac.il*

## Abstract

*In this paper we show that efficient object recognition can be obtained by combining informative features with linear classification. The results demonstrate the superiority of informative class-specific features, as compared with generic type features such as wavelets, for the task of object recognition. We show that information rich features can reach optimal performance with simple linear separation rules, while generic feature based classifiers require more complex classification schemes. This is significant because efficient and optimal methods have been developed for spaces that allow linear separation. To compare different strategies for feature extraction, we trained and compared classifiers working in feature spaces of the same low dimensionality, using two feature types (image fragments vs. wavelets) and two classification rules (linear hyperplane and a Bayesian Network). The results show that by maximizing the individual information of the features, it is possible to obtain efficient classification by a simple linear separating rule, as well as more efficient learning.*

## 1. Introduction

Schemes for visual classification usually proceed in two stages. First, features are extracted from the image, and the object to be classified is represented using these features. Second, a classifier is applied to the measured features to reach a decision regarding the represented class. Powerful methods have been developed for performing visual classification by linear separation, that is, when the representation of class and non-class examples can be separated by a hyperplane in feature space. Early algorithms for separating class from non-class images in this manner include the Perceptron [12] and Winnow [10] algorithms. A more recent method is the Support Vector Machine [21], which is computationally efficient, and, under some general assumptions, can determine the optimal separating hyperplane.

Unfortunately, in many cases, the representation of class and non-class examples in feature space does not allow simple separation. For example, when the image intensity values are used as the basic features, the separating surface between class and non-class images is usually highly nonlinear and therefore difficult to learn or to approximate. One approach to obtain better classification, used by Support Vector Machines, has been to map the data from the original feature space to a much higher dimensional space in which the classes become separable, as in [14]. However, there is no simple method for obtaining a successful mapping. In practice, one can try a number of different mappings (e.g., different kernels in the case of SVMs) and test the performance of the resulting classification. Another general approach has been to develop more complex classification methods, for example by multi-layer neural-network models [9, 16] that do not require linear separation between the classes. Unlike linear separation, there is no general method for obtaining optimal classification in this case. In practice, one only obtains a local optimum and can test the performance to determine the adequacy of the classification.

Regarding the issue of selecting the features and a classification scheme for object recognition, past approaches suggest a trade-off between the complexity of features and the complexity of the classification scheme. A first group of methods uses simple generic features in very high dimensional spaces, usually combined with elaborate classification schemes. If the features themselves are simple and not informative, then a large number of features is required and the classification function must extract the relevant information from the feature distributions. Such approaches are proposed in [2, 9, 11, 14, 16, 18, 22, 24]. Representative of these methods are [2] that uses edge type features combined with a decision tree for character recognition and face detection, and [9, 16] that plug the raw gray level intensities into a multi-layer neural-network for face and character recognition as well. Conversely, for methods using richer, class specific features, the separation becomes easier, the dimensionality of feature space is reduced and linear type classification functions or simple probability distribution models can be used. Such schemes are found in [1, 3, 13, 19, 20, 23].

[1, 20] use image patches as features and combine them with a naive-Bayes scheme or with a SNoW classifier; [19] generates a low dimensional subspace from an eigen-image basis and uses a simple nearest neighbor classifier.

This trend raises the possibility that by explicitly maximizing the information content of the features with respect to the class, it may be possible to obtain feature spaces where simple linear classification is sufficient. The current study tests and confirms this possibility. This is significant because efficient and optimal methods have been developed for spaces that allow linear separation. To test our approach, we compared classification schemes using two types of features and two types of classification rules. The features could be either object fragments selected by maximizing an information criterion, or simpler and more traditional wavelet-like features. The classification rules could be either a simple linear separating function, or a more complex model of the feature distribution, that takes into account higher order statistical dependencies between features.

The rest of the paper is organized as follows. In Section 2, we present the class-based informative features that we used, together with the algorithm for their extraction. In Section 3 we describe generic features that were used successfully in other studies, and used in the current comparison. In Section 4 and 5, we describe the two classification schemes used in the comparison, which were the Linear SVM and the Tree-Augmented Network. In Section 6, we present experimental results for the problem of detecting side-views of cars in low-resolution images, using the different classification approaches. Finally, Section 7 contains a discussion of the results and conclusions.

In summary, we compared two feature types, generic (wavelets) and informative (fragments), and two classification schemes, a simple linear separator and a more complex Tree-Augmented Network. The results show that simple features require a more complex classification function that relies on higher order aspects of the features distribution. In contrast, with informative features, learning becomes easier and a simple linear separator can reach optimal classification performance. The advantages of using informative features with linear classification rules are discussed in the conclusion.

# 2. Informative features and their extraction

Many of the more recent recognition systems use class specific image patches as visual features, such as patches extracted from images based on local image properties as in [1], eigen-patches of similar parts as in [23], or parts defined by the user as in [13]. A particularly useful set of features are intermediate size image fragments, that arise naturally when searching for a set of features that maximizes the information content of the set with respect to the class [20]. In extracting informative features, we follow the scheme introduced in [20]. The feature selection process is described in detail in Subsections 2.1-2.3 below.

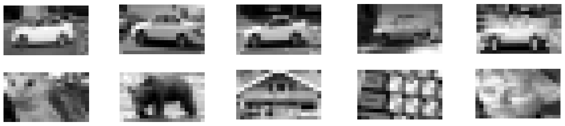

Figure 1: Examples of low resolution, 14x21 pixel, car and non-car images used to train and test the system.

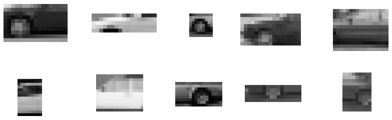

Figure 2: Examples of informative, low-resolution car fragments extracted automatically by the system (shown at higher resolution for clarity).

## 2.1. The selection of informative features

We outline here the selection process of informative image fragments as features for classification, performed during the training stage. The goal is to select class-specific image fragments that convey the maximal amount of information about the class. Informative features are selected from a large pool of parts, typically several tens of thousands, cropped from images containing the class of interest, rectangular in shape, of different sizes and from different locations. We use a greedy-search algorithm along with an information measure to select a set of features that, together, convey the maximal amount of information about the class. Feature selection is the computationally heavy stage of the fragment-based scheme.

Here is the summary of the main steps for finding a set of informative fragments:

- Generate a large set of candidate fragments $\{F_i\}$
- Compute, for each fragment, the optimal threshold that determines the minimum visual similarity for it to be detected in an image (Subsection 2.2).
- Select a set of maximally informative features (Subsection 2.3).

Figure 1 shows low resolution (14×21 pixels) images that we used in the car side-views detection experiments, and Figure 2 shows informative features extracted automatically from the image training set.

### 2.2. Similarity measure and detection threshold

The presence of the fragments in an image is determined by the combined use of a similarity measure and a detection threshold. Using a sliding window over the image, we measure the presence of the fragment in the window with normalized cross-correlation, a common method used in computer vision to measure visual similarity, and compare the score to a threshold.

We treat a given fragment $X_i$ as a binary random variable expressing its presence or not in the image ($X_i = 1$ if the fragment is present, 0 otherwise). This requires a threshold $\theta_i$ that represents the minimal detection similarity. The value of $X_i$ depends on whether the maximal similarity found in the image is larger than $\theta_i$ or not.

The threshold $\theta_i$ is set automatically by maximizing the mutual information [6], $I(X_i; C)$, between the fragment $X_i$ and the binary class variable $C$. The conditional probabilities $P(X_i(\theta_i) = 0|C)$ and $P(X_i(\theta_i) = 1|C)$ required in the calculation of the information are computed from the training data. The class priors, $P(C = 0)$ and $P(C = 1)$, are chosen a priori.

The detection threshold for a fragment is formally defined by

$$\begin{aligned} \theta_i &= \arg\max_{\theta} I(X_i(\theta); C) \qquad (1) \\ &= \arg\max_{\theta} \left(H(C) - H(C|X_i(\theta))\right) \ . \end{aligned}$$

$H(x)$ [1] and $H(x|y)$ [2] are Shannon's entropy and conditional entropy; here, $x$ and $y$ take their values in $\{0, 1\}$.

This procedure automatically assigns to each fragment in the pool a detection threshold that maximizes the information delivered by the fragment. We next describe the selection of an optimal subset of fragments from the pool.

### 2.3. Greedy-Search

The feature selection process is based on a greedy-search algorithm [17] that adds fragments iteratively to the set of informative features, in a greedy fashion, until adding more fragments no longer increases the estimated information content of the set.

Denote the initial fragment pool by the set $P$, from which the fragments are to be chosen. After an initial filtering that removes the least promising features, the algorithm is initialized by moving from $P$ the fragment with the highest mutual information, obtained by eq. (1), to the set of selected fragments, denoted by $S_1$. $P_1$ now represents the pool after the transfer of the first fragment. In the next step, we seek a second fragment $X_2$ from $P$ to be added to the set of selected features. At this stage, however, the selection criterion is not the mutual information of $X_2$ alone, but how much information $X_2$ can add with respect to the already existing $X_1$. Therefore, $X_2$ should maximize $I(X_i, X_1; C) - I(X_1; C)$. Following the same scheme, we iteratively add the fragment that brings the highest increase of information content contained in the set $S$. The next fragment $X_k$ to be added at iteration $n + 1$ is defined by:

$$X_k = \arg\max_{X_i \in P_n} \min_{X_j \in S_n} \left(I(X_i, X_j; C) - I(X_j; C)\right) \ . \quad (2)$$

The updates of the pool and the set of selected fragments are defined by:

$$P_{n+1} = P_n \setminus X_k \quad \text{and} \quad S_{n+1} = S_n \cup \{X_k\} \ . \quad (3)$$

With eq. (2), the fragment that we add to $S_n$ is the one, among those in $P_n$, that yields the maximal increase in the estimated information content of the set.

The informative fragments selected in this way are used together with one or the other combination schemes described later in Sections 4 and 5. We next describe the other type of features, simple and generic, used in our study.

## 3. Simple generic features

Many state of the art recognition systems are based on the use of generic, non class-specific visual features, e.g. [22, 14, 18]. In this section, we describe the simple features we used in our comparison, generally following [14, 18].

The generic features used for object classification are designed to capture local frequency and orientation information of the image. The individual features therefore convey limited information about the class on their own. It is the right combination of these features that enables the system to capture the visual properties that are specific to the different classes of objects.

### 3.1. Wavelet transform

A class of features commonly used for object recognition tasks is the wavelet family, applied for pedestrian, face and car detection [14, 24, 18]. The wavelet transform captures frequency and orientation properties at all locations in the image within an analysis window, at different scales. It is characterized by a kernel function, whose choice influences the type of visual features to which the transform is sensitive. Figure 3 shows some examples of wavelet features that can be used. The first line of features represents Gabor-wavelets as used in [24]. The second line shows a set of biorthogonal 5/3 wavelets as in [18]. The third line displays 1/1 biorthogonal wavelets, used in [14], that work as simple discrete differential operators. These degenerate wavelets are also similar to the rectangular features used in [22], in the framework of fast face detection. In our tests using low-resolution car images, they performed better than alternative

[1] $H(x) = -\sum_x p(x) \log(p(x))$

[2] $H(x|y) = -\sum_{x,y} p(x, y) \log(p(x|y))$

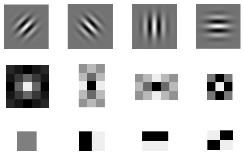

Figure 3: Typical wavelet features used in recent work on object recognition. Top: four oriented gabor-wavelet filters, used in [24]. Middle: biorthogonal 5/3 wavelets, used in [18]. Bottom: simplified wavelets, also used in [14], that we chose for our experiments for the generic feature type classifiers.

wavelet features and were therefore selected for our experiments. Note that the features generated from the wavelet transform are defined at every location in the image or in the analysis window, as opposed to the fragments whose presence or absence is defined in a given area.

### 3.2. Quantization

In [18], the coefficients of the wavelet transform are quantized into 3 levels. For the purpose of comparison with fragment-based classification, we binarize the wavelet transform so as to interpret the resulting transform as expressing the presence or absence of the different wavelet features at different locations in the image. The binarization process is done by thresholding the coefficients of the wavelet transform by their measured average in the set of training images. In the experiments, the use of thresholds other than the average led to a decrease in the classification performance.

## 4. Classification by linear separation

During classification, the system generates a feature vector $X = [X_1, ..X_n]$ that represents the encoding of the image in feature space. For example, it can be obtained by measuring the presence of specific visual features in the image. Final decision about the class is performed by plugging the feature vector into a classification function $f(X)$ that returns 1 or 0, depending on whether the object is estimated to be present or not.

The simple feature combination rule we tested is a linear discriminant, learned with a Linear Support Vector Machine (LSVM). The linear discriminant has the following functional form:

$$f(X) = \begin{cases} 1 & \text{if} \quad \sum_i \alpha_i X_i \geq \theta, \\ 0 & \text{otherwise} \; . \end{cases} \qquad (4)$$

where the $X_i$ represent the measured value of the individual features. The $\alpha_i$ are the weights of the features and are obtained during the learning phase. $\theta$ is a bias term.

Linear SVM training is used to learn the optimal discriminant function. SVMs are classifiers that learn a linear decision surface in feature space, the Maximum Margin Surface (MMS), which is optimal in the sense that it lies as far as possible from the class and non-class data points in feature space. When the data is not linearly separable, the function to minimize is not just the margin but the margin combined with a cost depending on the number of misclassifications. Finding the MMS is a quadratic programming problem and is therefore attractive because computationally efficient and guaranteed to reach the optimal solution under general conditions. Detailed descriptions of SVMs can be found in [4, 21]. Linear SVM training yields a vector $\alpha = [\alpha_1, ..\alpha_n]$, normal to the decision surface, and used in eq. (4) during classification.

A more typical use of SVMs is to find non-linear decision surfaces in feature space. This is obtained by projecting non-linearly the feature space onto a very high dimensional projective space and finding a maximal margin hyperplane there. Here also, a cost can be used when the data is not separable in the projective space. We present some recognition results using a polynomial SVM, for comparison, in the experiments Section 6.

## 5. The Tree-Augmented Network

In this section, we present the more complex classification scheme we tested, the Tree-Augmented Network (TAN).

Unlike the LSVM scheme in which, during classification, features are used independently of each other, the TAN takes into account some pairwise statistical dependencies between features, thereby enabling a better approximation of their underlying distribution. The TAN is therefore a richer model than linear discriminants. It is a particular Bayesian network [15, 8], where the features, represented by the nodes of the graph, are connected to the class variable and are organized in a tree structure, as shown in figure 4. The edges in the tree express statistical correlation between connected features. In this probabilistic model, the probability for a feature $X_i$ to have a specific value depends not only on the value of the class variable $C$, but also on the value of its parent feature $X_{\Pi(i)}$. Imposing a tree structure on the network restricts the modelling power but enables straightforward computation of the probability of an input, given by eq. (5), which is not the case with loopy networks [15]. The structure of the tree is found during learning by searching for the maximum weighted spanning tree, where the weight of an edge connecting features $X_i$ and $X_j$ is the mutual information $I(X_i; X_j)$ between $X_i$ and $X_j$ [5].

Formally, the class-conditional distributions modelled by the TAN have the following form:

$$P(X_1, ..X_N|C) = \prod_{i=1}^{N} P(X_i|X_{\Pi(i)}, C) \; . \qquad (5)$$

The optimal Bayes decision rule [7] obtained with the TAN

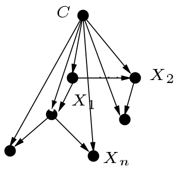

Figure 4: The Tree-Augmented Network classifier. The model assumes some pairwise dependencies between the features. The features are organized in a tree, i.e., connected to the class node and at most one other feature node.

model is:

$$f(X) = \begin{cases} 1 & \text{if} \quad \prod_{i=1}^{N} \frac{P(X_i|X_{\Pi(i)},C=1)}{P(X_i|X_{\Pi(i)},C=0)} \geq \theta, \\ 0 & \text{otherwise .} \end{cases} \qquad (6)$$

$X_{\Pi(i)}$ is the parent feature of $X_i$. The probabilities $P(X_i|X_\Pi(i),C)$ that parameterize the model are learned from the training data.

Note that for binary features, i.e., $X_i \in \{0,1\}$, it can be shown that the decision rule in eq. (6) defines a quadratic surface in feature space, in contrast to the linear surface obtained with the LSVM.

# 6. Experiments

We now describe the experiments comparing the different classifiers and features discussed. The first part of the section describes the experimental setup. The second part concerns the recognition performance of classifiers using the different possibilities between the LSVM or the TAN classification functions, and wavelet or fragment type features. For comparison, we also added recognition results using a non-linear SVM. The third part deals with the training effectiveness, for both fragments or wavelets.

## 6.1. Training and testing the classification schemes

We compared the performance of the two feature types, fragments and wavelets, and two combination schemes, LSVM and TAN. The classification task consisted of the detection of side-views of cars in 14x21 pixel images. The image database comprised a total of 573 car images and 461 non-car images. The cars occupied approximately a 10x15 pixel box inside the image. From this data, we trained and tested four classifiers corresponding to the different possibilities obtained by choosing fragments or wavelets as features, and the LSVM or the TAN for the classification function. The performance of the classifiers was estimated by a cross-validation method: we repeatedly trained and tested the classifier on independent data sets, that were reshuffled at each iteration. We performed 20 cross-validation iterations to generate the ROC curves presented in Section 6.2. The initial selection of the fragments, on a Pentium computer, took several hours, using Matlab. The bottleneck of the method is the measurement of the features on the training images and their selection. The computation time can be significantly reduced, however, to 10 or 30 minutes, if we restrict the search to the intermediate sized features, as in [23, 1], rather than features with sizes ranging from very small to full templates. Learning the TAN takes less than a minute, while learning the LSVM is virtually instantaneous, for 168 features.

The dimension of the binary feature vector representing the detection of features, was taken to be the same for the fragment-based and the wavelet-based classifiers, 168, for comparisons in spaces of same dimension.

### 6.1.1. Fragment-based classifiers

The initial pool of fragments $P$ contained 59200 fragments, extracted from the first 100 cars. Their sizes varied from $4\times4$ pixels to $10\times14$ pixels and were taken from all the possible locations in the 10x15 pixel region surrounding the car. Each fragment was labelled with the rough location from which it was extracted, enabling us to restrict the detection zone to a limited area. This gives the fragments a certain degree of translation invariance, while capturing rough spatial relations between the different fragments. In the experiments, the fragments were allowed to move in a 5x5 pixel area surrounding their original location.

We used the remaining 473 car images and the 461 non-car images for training and testing. At every iteration of the cross-validation process, we randomly selected 200 car and 200 non-car images for training. The complementary 273 cars and 261 non-cars were used to test the classifier. The training images served to select the useful fragments and for the learning of the classification functions.

### 6.1.2. Wavelet-based classifiers

We used the simplified wavelet operators shown on line 3 in Figure 3, at 2 different scales. We also tested the biorthogonal 5/3 wavelets that were used in [18], but they gave poorer results. They were probably too large for the images we used, and smeared the orientation and frequency information. The performance of the wavelet-based classifiers was assessed in the same way as the fragment-based classifiers. At each iteration of the cross-validation process, we used 200 car and 200 non-car images, limited to the $10\times15$ car area to learn the classification functions, LSVM or TAN. The rest of the images used for performance assessment were taken in their $14\times21$ format to impose a degree of translational invariance similar to the one tested in the fragment-based scheme. The decision about the class of an image was based on the maximal response of the classifier over each the $10\times15$ pixel windows in the image.

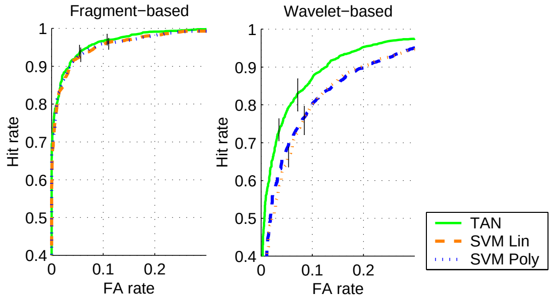

Figure 5: ROC curves for the different classifiers, using a TAN classifier, an LSVM classifier (Lin), or a Non-Linear SVM (Poly). The fragment-based scheme performs better. The complexity of the classification method influences substantially the performance for the wavelet-based classifier: the TAN curve is significantly higher than both SVM methods, that have overlapping curves. The classification scheme does not affect the fragment-based classifier, where the three curves virtually overlap.

## 6.2. Classification results

The classification results for each classifier are presented in the form of the Receiver Operating Characteristic (ROC) curves shown in Figure 5. ROC curves represent the ability of classifiers to combine the constraints of having a low false-positive rate and a high detection rate. The higher the curve, the better the classifier. The curves were obtained by averaging the results of the cross-validation iterations.

The graph shows that the fragment-based scheme performs better than the scheme using wavelets. For example, at a 5% false-alarm rate, detection rates for the fragment based classifier are over 92% when using the different classification functions. For the wavelet-based classifiers, the detection rate is around 70% with the LSVM combination, and reaches 80% with the TAN. More important to the current discussion is the influence of the classification functions. The use of the TAN scheme versus LSVM enhanced substantially the performance of the wavelet-based classifier, while the performance of the fragment-based classifier was virtually unaffected.

We also show ROC curves obtained with a Non-Linear SVM, using a polynomial kernel of degree 3, for comparison. The performance of the Polynomial SVM was similar to the LSVM and the TAN when using the fragments features. With wavelets, the Polynomial SVM had a performance equivalent to the LSVM, and both were outperformed by the TAN.

Figure 6 summarizes the recognition results in terms of information. The diagram shows the information content gain of the TAN combination scheme for wavelets (cross) and fragments (square), averaged over the cross-validation iterations. The information was computed at a 5% false-alarm rate. The information gain is defined as the difference between the information provided about the class by the TAN classifier and the information provided by the LSVM classifier:

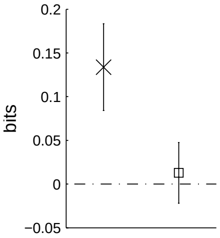

Figure 6: Information gain $\Delta I$ in bits using the complex combination scheme, for wavelets ($\times$) and fragments ($\square$). The gain is much higher for the wavelet-based classifier than for the fragment-based classifier.

$$\Delta I = I(C;\widehat{C}_{\text{TAN}}) - I(C;\widehat{C}_{\text{LSVM}}) \qquad (7)$$

where $I(C;\widehat{C})$ is the information provided by a classifier (simple or complex), defined as the mutual information between the final decision of the classifier $\widehat{C}$ and true class of the image, $C$:

$$I(C;\widehat{C}) = \sum_{C,\widehat{C}\in\{0,1\}} p(C,\widehat{C}) \log \frac{p(C,\widehat{C})}{p(C)p(\widehat{C})} \qquad (8)$$

For perfect classification, with $P(C=0) = P(C=1) = 0.5$, $I(C;\widehat{C}) = 1$. For random decision, $I(C;\widehat{C}) = 0$.

The graph shows that the more complex combination scheme contributes significantly to the information delivered by the wavelet-based classifier, while for the fragment-based classifier, the complex combination scheme adds little or no information, and may even reduce it, as can be seen by the error bar falling under $0$ in Figure 6. The occasional loss of information when using the more complex scheme stems from over-fitting the classifier parameters, that are also harder to learn because they involve second order statistics and require more training data to be accurate, thereby affecting its generalization capacity. In the fragment-based classifier, the useful information for classification is already contained in the features themselves, and consequently, the scheme relies less on higher-order interactions.

We considered the possibility that the poor performance of the wavelet-based classifier may be caused by the loss of information due to the binarization process, rather than the expressiveness of the features. We therefore tested linear and non-linear SVMs with the full wavelet coefficients, rather than their binarized values, but this actually led to a decrease in classification performance, in our low-resolution application.

Note also that the performance of the wavelet-based classifier could eventually be increased by using yet higher-order statistics in the feature distribution model. However, this would require heavier computations and more training data to learn the higher-order interactions correctly.

The experiments reported above were supported by similar additional experiments, using different object classes and different simple features. We performed the same feature extraction procedure and classification to face rather than car images. Classification of the face images (face vs. non-face images) based on informative fragments was performed with linear classification, and the improvement using non-linear classification was not significant. In addition, we trained a back-propagation neural network to extract face features and classify face vs. non-face images. The information content of the features extracted by the network was low on average, less than 10% of the information obtained by fragments. We then tested the features extracted by the backpropagation network, but with linear classification. This resulted in a severe decrease in recognition performance. We conclude that the extraction of class-specific informative fragments is a practical method to obtain feature spaces in which linear separation is effective. In contrast, for simpler and more generic features of the type used by many current classifiers, the use of a simple separating hyperplane is far from optimal.

### 6.3. Feature type and the difficulty of training

We measured how the amount of training images influences the generalization capacity of the wavelet-based and the fragment-based classifiers. For this purpose, we measured $I(C;\widehat{C})$ (eq. 8), at a false-alarm rate of 5%, for the wavelet-based and the fragment-based classifiers on a set of unseen images, as a function of the number of training images. As in Subsection 6.1, the measurements were performed with 20 cross validation loops, using part of the database for training and the complement for testing and displaying the results, presented in Figure 7. The classification rule for the wavelet-based classifier was the TAN, while the classification rule for the fragment-based classifier was the LSVM.

The increase in information between using 50 and 250 training images per class is more substantial for the wavelet based-classifier than for the fragment-based classifier, with an increase of 0.14 bit and 0.065 bit on average respectively. Also, the fragment based scheme with 50 training images per class still performs significantly better than the wavelet-based classifier with 250 training images per class.

From these results, it appears that the learning strategy using fragments is more efficient than the wavelet-based strategy, in that it learns faster, i.e., from fewer examples, the common structure of images that discriminates between the class.

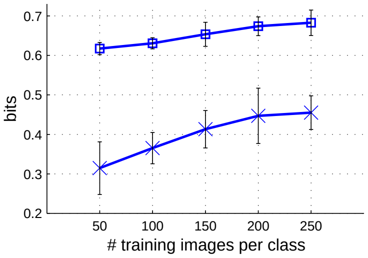

Figure 7: Information versus the number of training images per class, for wavelets (×) and fragments (□). The fragment-based scheme performs better than the wavelet-based scheme, even when learning is done using less data. The increase in information between 50 and 250 training images per class is more significant for the wavelet scheme than for the fragment scheme.

## 7. Discussion and Conclusion

We can compare our approach to two main strategies that use simple features for object recognition. Since simple generic features that are not selected specifically for the class of images at hand usually do not allow effective linear classification, one general approach is to develop more complex classification stages, such as multi-layer neural networks. There is no general optimal method for this task, but a variety of techniques can be developed and tested for a given application. A second approach, which led to the Support Vector Machine and the different kernel based techniques, has been to use a mapping to a higher dimensional space where the linear separation becomes more effective. There is no straightforward method for finding a good mapping, and different mappings must usually be applied and evaluated. A third approach, supported by the comparisons in this study, is to first extract during learning a set of information rich features, selected for the specific class to be recognized, followed by the use of a simple classifier, constructed for example by a linear SVM.

Our comparative study shows that linear separation can be obtained in low dimensional feature space if the features are chosen to be highly informative. If the individual features themselves have a low information content, it can be expected that the required number of features will be large. This is also supported by the following consideration. For features that are conditionally independent (the fragments and other features used for classification are often selected to reduce conditional dependence), it can be shown that

$$I(X_1,..X_N;C) \leq \sum_i^N I(X_i;C) = N\bar{I} \qquad (9)$$

where $\bar{I}$ is the average mutual information of the fragments and $N$ is the number of fragments. To obtain perfect classification, $I(X_1,..X_N;C)$ must be equal to $H(C)$, the entropy of the class variable. From this we conclude that

$$N \geq \frac{H(C)}{\bar{I}} \tag{10}$$

For correlated features, the required number will usually be higher. This supports the conclusion that the number of features used for classification is related to the information content of the individual features. In addition, our comparisons show that for simple generic features the classifier had to use higher-order properties of their distribution. Conversely, when the individual features were by themselves informative, the relative contribution of the higher-order interactions was reduced and a linear decision rule was enough for efficient classification.

We showed how informative features can be automatically extracted. This requires an extensive search but the procedure is straightforward, and it is performed as an offline stage. Recognition schemes using such features can then take advantage of known techniques that are guaranteed to find an optimal separating hyperplane. Taken together, the results show that a practical method to obtain efficient recognition is to combine the extraction of informative features with linear classification.

Finally, it would be interesting to examine in future studies the useful combination of both simple and complex features in multi-stage classification schemes. The informative features allow reliable classification with simple decision rules, but their extraction over the entire image may be more demanding than the extraction of some families of features designed for fast extraction, such as integral features [22]. A combined scheme could use the simpler features for initial filtering and the identification of sub-regions in the images that may contain an object of interest, followed by the application of the reliable and informative features to the selected regions.

# References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proceedings of ECCV 2002*, volume 4, pages 113–130, 2002.

[2] Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.

[3] M. S. Bartlett and T. J. Sejnowski. Viewpoint invariant face recognition using independent component analysis and attractor networks. In *Advances in Neural Information Processing Systems*, volume 9, page 817. The MIT Press, 1997.

[4] C. J. C. Burgess. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[5] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT14(3):462–467, May 1968.

[6] T. M. Cover and J. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley and Sons, New-York, NY, USA, 1991.

[7] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc., 1973.

[8] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, Winter 1989.

[10] N. Littlestone. Learning quickly when irrelevant attributes abound:a new linear-threshold algorithm. *Machine Learning*, 2, 1988.

[11] B. Mel. Seemore: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Computation*, 9:777–804, 1997.

[12] M. L. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, Massachussets, 1988.

[13] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. PAMI*, 23, 4, 2001.

[14] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of International Conference on Computer Vision*, 1998.

[15] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of Plausible Inference*. Morgan Kaufmann, California., 1988.

[16] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.

[17] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence, Upper Saddle River, New Jersey, 1995.

[18] H. Schneiderman and T. Kanade. A statistical approcah to 3d object detection applied to faces and cars. In *Proceedings of the Eighth IEEE International Conference on Computer Vision (2000)*, June 2000.

[19] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[20] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. In *Proc. 4th IWVF 2001*, May 2001.

[21] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[22] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[23] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. 6 th Europ. Conf. Comput. Vision*, June 2000.

[24] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.