Adaptive Patch Features for Object Class Recognition with Learned Hierarchical Models

Fabien Scalzo[†]
Computer Vision Laboratory
University of Nevada, Reno
scalzo@cse.unr.edu

Justus H. Piater
Montefiore Institute
University of Liège, Belgium
Justus.Piater@ulg.ac.be

Abstract

We present a hierarchical generative model for object recognition that is constructed by weakly-supervised learning. A key component is a novel, adaptive patch feature whose width and height are automatically determined. The optimality criterion is based on minimum-variance analysis, which first computes the variance of the appearance model for various patch deformations, and then selects the patch dimensions that yield the minimum variance over the training data. They are integrated into each level of our hierarchical representation that is learned in an iterative, bottom-up fashion. At each level of the hierarchy, pairs of features are identified that tend to occur at stable positions relative to each other, by clustering the configurational distributions of observed feature co-occurrences using Expectation-Maximization. For recognition, evidence is propagated using Nonparametric Belief Propagation. Discriminative models are learned on the basis of our feature hierarchy by combining a SVM classifier with feature selection based on the Fisher score. Experiments on two very different, challenging image databases demonstrate the effectiveness of this framework for object class recognition, as well as the contribution of the adaptive patch features towards attaining highly competitive results.

1. Introduction

Over the past years, part-based methods emerged as a promising family of methods for recognizing object classes [10]. Classically, these methods represent an object class by a collection of local parts that are consistent in both shape and appearance. However, the range of variations in these spaces is often difficult to represent by a model that consists of a single level of abstraction.

Recently, hierarchical approaches based on statistical models [1, 2, 3, 4, 8, 11] have received increasing attention. These are well suited to representing shape variability at different scales and granularities. However, they often tend to reduce the object to a sparse collection of local parts and do not include large-scale appearance representations at higher levels of the hierarchy.

The current paper addresses this problem by presenting a statistical method for learning a hierarchical model (Section 2) that exploits a new type of *adaptive patch features*. The hierarchical feature structure is naturally integrated into a graphical model formalism. This allows the formulation of detection (Section 4) as probabilistic inference that can be efficiently implemented e.g. by Nonparametric Belief Propagation (NBP). The proposed learning framework (Section 3) focuses on modeling spatial relations and high-level appearance between correlated features. Unlike many existing frameworks, the topology of the model itself is initially unknown; the graph is automatically and incrementally constructed by combining correlated features into higher-level abstractions. Discriminative models (Section 5) are learned on the basis of our feature hierarchy by combining a SVM classifier with feature selection based on the Fisher score. We present an empirical performance evaluation on two standard object recognition tasks in Section 6.

2. A Hierarchy of Visual Features

Our object model takes the form of a compositional hierarchy of visual feature classes [8, 11]. Climbing up this hierarchy, features correspond to increasingly complex object parts defined in terms of constellations of lower-level parts. At some level, parts become representative and specific enough to abstract the whole object class. The particularity of this scheme lies in the fact that each feature class $f = \{\mathcal{A}, \mathcal{S}, \mathcal{X}\}$ is described by an appearance model $\mathcal{A}$ and spatial relations $\mathcal{S}$ with respect to higher- and/or lower-level features. No relations occur between features within the same level. Pairwise spatial relations are defined in terms of relative positions between two feature classes. To compute the appearance model, the spatial extent of each visual class is normalized into a specific canonical shape $\mathcal{X}$. This shape is estimated using a novel, adaptive method (Section 3.2).

[†]Fabien Scalzo is supported by a Post-doctoral Fellowship of the Belgian American Educational Foundation (BAEF).

During detection, a visual feature class can be instantiated several times in the image. An instance of the visual feature f is defined as a triple $\mathcal{I}_f = \{l, p, w\}$, where l stands for a location in the image, p is the local affine pose of the feature instance, and w is the weight corresponding to the confidence of observing this particular instance. In the following, we will often refer to *visual feature classes* and their *instances* in a given image by using the terms "feature" and "instance", respectively.

The proposed recognition system employs an undirected graphical model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a Pairwise Markov Random Field, to implement an object model. Nodes in $\mathcal{V}$ correspond to feature classes characterized by their specific appearance, and edges in $\mathcal{E}$ represent spatial relations by pairwise potentials (Fig. 1). Both are learned from training data (Sec. 3). Together, they represent the structure of a given object class.

2.1. The Vertex Set

Our model distinguishes between hidden nodes x and observable nodes y, $V = x \cup y$. An observable node $y_i \in y$ corresponds to image measurements extracted by feature detectors at local image regions. Each hidden node $x_i \in x$ represents a feature class f_i. The instantiation of the feature class in a given image is defined by a spatial probability density represented by the hidden node.

A **hidden node** $x_i \in x$ represents a feature class of the model and is associated with an appearance model $\mathcal{A}_i$, represented by a mean appearance vector $\mu_i^{\mathcal{A}} \in \mathbb{R}^n$ and corresponding covariance matrix $\Sigma_i^{\mathcal{A}}$.

The belief associated with a hidden node is a spatial probability density that represents the plausible presence of instances of the corresponding feature class in a given image. During detection, these beliefs are inferred from image observations (Sec. 4).

A weighted kernel density estimation (KDE) is used to model the spatial density at each hidden node $x_i \in x$. The multivariate kernel estimator is defined as $\hat{f}(x; \Theta) = \frac{1}{n} \sum_{i=1}^{n} w_i \mathcal{G}(x; \mu_i, \Sigma_i)$, where μ_i is a point in $\mathbb{R}^2$, w_i is a weight and Σ_i is a smoothing matrix. Since an instance description includes feature pose, a parameter corresponding to the local affine deformation ϑ_i is associated separately to each kernel component,

$$\forall i \in n,\ \Theta \leftarrow \{\mu_i, w_i, \Sigma_i\} \cup \{\vartheta_i\} \tag{1}$$

An **observable node** $y_i \in y$ represents a set of local affine regions extracted by feature detectors. A *local region* is defined as a triple $(\alpha, \vartheta, \mathcal{D})$, where $\alpha \in \mathbb{R}^2$ stands for an image location, ϑ is an affine deformation matrix, and $\mathcal{D} \in \mathbb{R}^{N_d}$ is a local descriptor that represents the appearance around the point α. To each node $y_i \in y$ is associated a set of region detectors $F_{l=1\ldots L} \in F$ (Section 4.1). The use of different detectors generally offers more robustness.

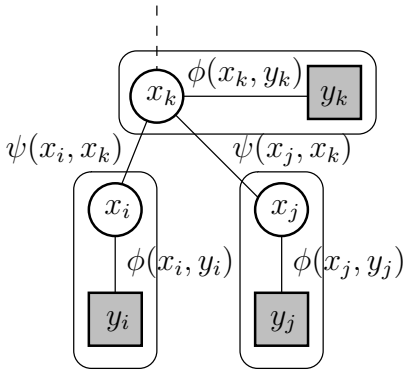


Figure 1. In our representation each feature is associated with an observable node y and a hidden node x linked through a local observation potential $\phi(x, y)$. Each pairwise potential $\psi(x_i, x_j)$ encodes the spatial relation between two feature classes i, j.

The union $\mathcal{O}_i(I)$ of the local affine regions obtained from each detector F_l associated to the current node y_i defines its instantiation for a given image I, where

$$\mathcal{O}_i(I) \leftarrow \bigcup_{F_{l=1\ldots L} \in F_{y_i}} \{\alpha, \vartheta, \mathcal{D}\}_{F_l(I)} \tag{2}$$

2.2. The Edge Set

We distinguish two types of edges $e \in \mathcal{E}$ in the graphical model that stand for pairwise and observation potentials.

A **pairwise potential** $\psi(x_i, x_j)$ represents the spatial relation between two neighboring hidden nodes (x_i, x_j). Intuitively, such a potential can be seen as a mapping function that encodes the relative positions of one feature with respect to another feature. Similarly to other work [13, 11], in practice we use two conditional functions $\psi(x_i|x_j), \psi(x_j|x_i)$ instead of a joint potential $\psi(x_i, x_j)$. The conditional density is approximated by a Gaussian mixture

$$\psi(x_j|x_i) = \sum_{k=1}^{N_{ij}} w_i^k\ \mathcal{G}(x_j; \gamma_{ijk}(x_i), \Sigma_i^k) \tag{3}$$

where w_i^k is the relative weight of an individual component and γ_{ijk} is a mapping function that computes the feature positions for the k-th component. Specifically, it displaces the feature instances of x_i using the k-th relative position $\mu_{ijk} \in \mathcal{S}_{ij}$ estimated between x_i and x_j (Section 3.1).

The **observation potentials** $\phi(x_i, y_i)$ correspond to the likelihood functions in the standard Bayesian formulation of an inference problem. This allows to represent the compatibility between a hidden node x_i and its corresponding image evidence y_i. Given a set of observed local regions $\mathcal{O}_k(I) = (\alpha_k, \vartheta_k, \mathcal{D}_k)$ at node y_i, the likelihood that a certain observed descriptor $\mathcal{D}_k$ detected at location α_k is an instance of the feature appearance model $\mathcal{A}_i$ can be formulated by creating a spatial Gaussian F_k at α_k weighted by a

similarity measure w. The likelihood $\mathcal{L}$ for a given point p in the image corresponds to the maximum response among all weighted Gaussian F_k (Eq. 4) at point p, that is

$$\begin{aligned} F_k &= w\,\mathcal{G}(\alpha_k, \Sigma) \text{ where } w = \exp(-\lambda(\mathcal{D}_k, \mathcal{A}_i)) \\ \mathcal{L}(p) &= \text{argmax}_k F_k(p) \end{aligned} \quad (4)$$

where $\lambda(\mathcal{D}_k, \mathcal{A}_i)$ is the Mahalanobis distance between an observation $\mathcal{D}_k$ and the appearance model $\mathcal{A}_i$.

3. Learning a Hierarchy of Feature Classes

The basic concept behind the learning algorithm is to accumulate statistics of the relative positions of observed features in order to find frequently-occurring feature co-occurrences. The structure of the model is built incrementally by combining spatially correlated feature classes into new feature abstractions.

First, a clustering algorithm (K-means) is applied to the set of descriptors $\mathcal{D}$ of local regions previously extracted from the training set. The number of classes is selected according to the BIC criterion [12]. This yields a visual codebook that is used to create the first level of the graph $\mathcal{G}$. Each feature is associated to a visual word of the codebook to create an appearance class $\mathcal{A}_p$. Figure 2 shows the 94 descriptor classes learned from the color pixel intensity patches for a given object class.

After clustering, the training procedure accumulates information on the relative positions Λ of features and their image locations Φ, and extracts those feature pairs $\mathcal{C} \leftarrow [f_i, f_j]$ that tend to be located in the same neighborhood. Then it estimates the parameters of their geometric relations $\mathcal{S}_{ij}$ using Expectation-Maximization (Section 3.1). It selects the closest relations and estimates their shape $\mathcal{X}$ and their appearance model $\mathcal{A}$ using a new adaptive method (Section 3.2). Finally, it generates new feature nodes in the graph (Section 3.3). The process is applied iteratively to each new level in the graph. In the following sections, we describe the main steps of this weakly supervised learning procedure, whose outline is given in Algorithm 1.

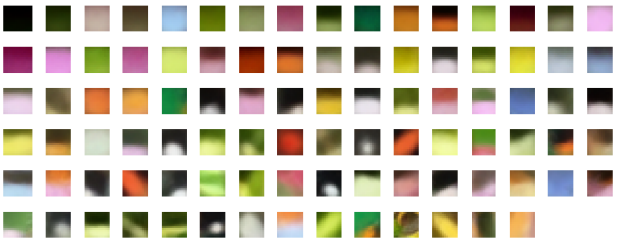


Figure 2. A visual codebook obtained from 27 training examples from one object class (Admiral Butterfly). The feature classes are sorted in descending order according to the number of assigned members.

Algorithm 1 Learning: learn()

1: $\{\alpha, \vartheta, \mathcal{D}\} \leftarrow$ regions extracted from the training set
2: $\{f_p, \mathcal{A}_p\} \leftarrow$ K-means($\mathcal{D}$) *// cluster the descriptors*
3: $\mathcal{G} \leftarrow$ create($f_p, \mathcal{A}_p$) *// first level of the graph*
4: **for each** level $<$ nLevels **do**
5: *// extract co-occurrence statistics: correlated features, their relative positions and image locations*
6: $\mathcal{C}, \Lambda, \Phi \leftarrow \text{extract}(\mathcal{G}, \text{level})$
7: **for each** correlated feature class pair $[f_i, f_j] \in \mathcal{C}$ **do**
8: $\mathcal{S}_{ij} \leftarrow \text{EM}(\Lambda_{i,j})$ *// estimate spatial relational model*
9: $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_{ij}$
10: **end for**
11: $\mathcal{S}' \leftarrow \text{closest}(\mathcal{S})$ *// keep the closest spatial relations*
12: $[\mathcal{X}, \mathcal{A}] \leftarrow \text{adaptivePatch}(\mathcal{S}', \Phi)$
13: $\mathcal{G} \leftarrow \text{generate}(\mathcal{X}, \mathcal{A}, \mathcal{S}', \mathcal{G})$ *// connect new nodes*
14: **end for**

3.1. Spatial Relations

Spatial relations are defined in terms of the relative position between two features. This is represented by a Gaussian mixture where each component represents a cluster of relative positions μ_k of one of the two features f_j with respect to the other, the reference feature f_i.

Expectation-Maximization (EM) is used to estimate the mixture parameters between each correlated feature pair $[f_i, f_j] \in \mathcal{C}$. EM maximizes the likelihood of the observed spatial relations over the model parameters $\Theta = (w_{1\ldots K}; \mu_{1\ldots K}; \Sigma_{1\ldots K})$. They are stored in a table $\mathcal{S}$ at the corresponding entry $\mathcal{S}_{ij}$ of the feature pair $[f_i, f_j]$.

3.2. Adaptive Patch Features

The feature combination from which a new feature will be created is not only defined by a spatial configuration $\mathcal{S}_{ij}$ of lower features, but also by an appearance $\mathcal{A}_{ij}$ over a region of shape $\mathcal{X}_{ij}$. We now propose an efficient method for estimating these parameters from a set of previously extracted positions $\Phi_{ij} \in \Phi$ of the feature pair $[f_i, f_j]$.

The scale at which the appearance should be extracted is a priori unknown. A naive approach would be to derive it from the distance between its parts. We consider this as an initial reference scale s_{init}; however, the optimal size of this region critically depends on the class and on the type of its neighborhood (region, edge, corner, ...). Too small or too large regions may result in information loss and inaccurate models. Therefore, it is desirable to estimate a specific spatial extent for each novel feature to compute its appearance. In this work, we use two scale factors s_x, s_y relative to the initial scale s_{init}, one for each dimension of the neighborhood, normalized with respect to the gradient orientation.

The optimal relative region size $\mathcal{X}_{ij} \leftarrow [s_x, s_y]$ is se-

lected by applying a minimum-variance analysis method. It starts by extracting appearance vectors at the detected locations Φ_{ij} of the combination in the training images for a set of N_s different pairs of scale factors $[s_x, s_y]_{N_s}$ where $s_x, s_y \in [0.2, 2.0[\times s_{\text{init}}$.

For each scale pair $[s_x, s_y]_j$, a trimmed mean $\mathcal{M} \in \mathbb{R}^N$ is computed from the extracted appearance vectors (Eq. 5). It is used to compute an N-dimensional vector of variances $\delta \in \mathbb{R}^N$. Then we select the scale factor pair $[s_x, s_y]_{\min}$ that minimizes the sum of variances over all N dimensions (Eq. 6).

$$\forall \mathcal{M}^i \in \mathcal{M}, \mathcal{M}^i = \frac{\sum_{(th_1 < a_i < th_2)} a_i}{\sum_{(th_1 < a_i < th_2)} 1} \tag{5}$$

$$[s_x, s_y]_{\min} = \text{argmin}_{[s_x, s_y]_{j \in N_s}} \sum_{i=0}^{N} \delta_i^{[s_x, s_y]_j} \tag{6}$$

Here, $\delta^{[s_x, s_y]_j}$ is the variance vector corresponding to a relative window size of $[s_x, s_y]_j$. This optimal scale selection procedure is illustrated in Figure 3. The shape model of the newly created compound feature class is then set to $\mathcal{X}_{ij} = [s_x, s_y]_{\min}$, and the appearance model $\mathcal{A}_{ij}$ to the mean appearance vector $\mu^{\mathcal{A}} = \mathcal{M}^{[s_x, s_y]_{\min}}$ and its corresponding variance $\Sigma^{\mathcal{A}} = \delta^{[s_x, s_y]_{\min}}$. For our experiments, appearance vectors are represented as color pixel values in the HSV colorspace. Note that any other description method (such as SIFT) can be used to represent them.

3.3. Feature generation

During the preceding steps, the learning process has identified reliable spatial relations $\mathcal{S}'$ between features, and has estimated the shape $\mathcal{X}$ and appearance model $\mathcal{A}$ that best characterize their neighborhood.

To incorporate these relations into the graphical model, the system generates a new pair of hidden and observable nodes (x_n, y_n) in the vertex set $\mathcal{V}$ for each pair of spatially related features (x_i, x_j) that appears in $\mathcal{S}'$. The new hidden node x_n corresponds to a higher-level feature and is linked to its subfeature nodes (x_i, x_j) by four conditional density functions $\psi(x_i|x_n)$, $\psi(x_n|x_i)$, $\psi(x_j|x_n)$, and $\psi(x_n|x_j)$.

A conditional $\psi(x_n|x_i)$ (Eq. 3) between two nodes is computed by means of a mapping function γ_{ink} (Eq. 7) that moves each component of x_i with the relative position μ_{ink} from x_i to x_n. To ensure symmetry, we set the position of the new feature x_n to the midpoint between its subfeatures, thus to half the distance of the relative position $\mu_{ijk} \in S'_{ij}$ of feature x_j from x_i and vice versa. The other conditionals $\psi(x_i|x_n)$, $\psi(x_j|x_n)$, $\psi(x_n|x_j)$ are defined similarly.

$$\gamma_{ink}(x_i) = x_i + \mu_{ink} = x_i + (\mu_{ijk}/2) \tag{7}$$

Each newly created hidden node x_n is associated to the shape $\mathcal{X}_{ij}$ and appearance $\mathcal{A}_{ij}$ of the feature combination.

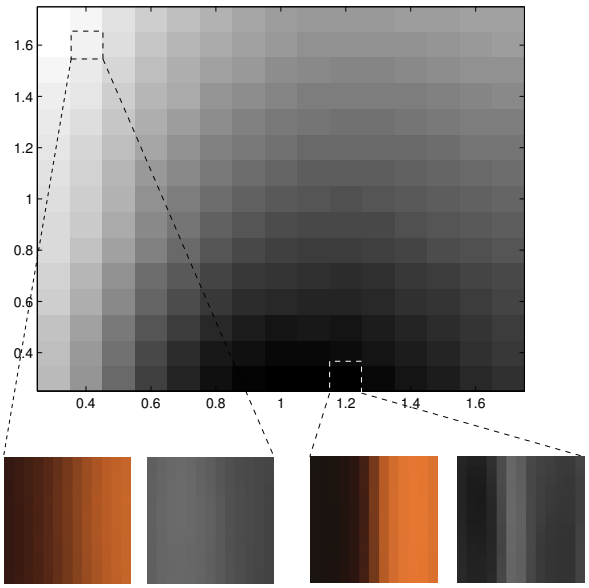


Figure 3. Illustration of the adaptive scale selection procedure. The gray value at each bin is proportional to the sum of dimension-wise variances for a pair of scale factors $[s_x, s_y]$. During the extraction process, each local patch is normalized to the local gradient direction computed at scale $s = (s_y + s_x)/2$ and resampled into a patch of 13×13 pixels. The trimmed-mean appearance and variance vectors corresponding to the optimal relative scale pair are shown on the bottom right.

Finally, the hidden node is linked to the observation by adding an observation potential $\phi(x_n, y_n)$.

4. Inferring High-Level Features

Computing the presence of features of an object representation in an image amounts to estimating $p(x|y)$, the posterior belief associated with the hidden nodes given all observations. Thus, detection of hierarchical features amounts to inference in our graphical model. One way to do this is to use Nonparametric Belief Propagation (NBP) [14]. NBP is an inference algorithm for graphical models that generalizes particle filtering and propagates information by a series of local message-passing operations. Following the notation of BP, a message m_{ij} from node i to j is written

$$m_{i,j}(x_j) \leftarrow \int \psi_{i,j}(x_i, x_j)\, \phi_i(x_i, y_i) \prod_{k \in \mathcal{N}_i \setminus j} m_{k,i}(x_i)\, \mathrm{d}x_i$$

where $\mathcal{N}_i$ is the set of neighbors of node i, $\psi_{i,j}(x_i, x_j)$ is the pairwise potential between nodes i, j, and $\phi_i(x_i, y_i)$ is the local observation potential. After any iteration, each node can compute an approximation $\hat{p}(x_i|y)$ to the marginal distribution $p(x_i|y)$ by combining the incoming messages with the local observation:

$$\hat{p}(x_i|y) \leftarrow \phi_i(x_i, y_i) \prod_{k \in \mathcal{N}_i} m_{k,i}(x_i)$$

In NBP, each message as well as each node distribution is represented through a kernel density estimate (Section 2.1). To better reflect the local distribution, we use a different bandwidth Σ_i for each sample point μ_i. A common way to compute the covariance is to use the k nearest neighbors, where an empirical choice for the integer k is $k \approx n^{1/2}$.

For tree-structured graphs, the beliefs will converge to the true marginals $p(x_i|y)$. On graphs with cycles, belief propagation is not guaranteed to converge. However, in practice the algorithm often exhibits satisfactory performance nevertheless.

4.1. Low-Level Feature Extraction

The purpose of feature extraction is to reduce the visual input space to a set of local regions. These regions are described by vectors of HSV color pixel values that are used to incorporate information into observable nodes y at the lowest level of the graph $\mathcal{G}$.

Ideally, our recognition framework should be able to learn different object classes without restrictions concerning their shape, appearance or texture. A common problem in object recognition is that different object classes might be described by different visual properties. Most existing approaches only use one type of detector. Opelt *et al.* [9] recently combined multiple methods to capture the main characteristics of various object categories. This improves the generality of their approach and therefore motives us to exploit a similar feature extraction scheme where MSER, Hessian-Laplace, Hessian-Affine and Random Patches are combined within the same framework.

5. Supervised Selection of Visual Features

Once a graphical model has been learned for each object class, they can be used for detection in new images. However, since each model has been constructed from co-occurrence statistics and without using discriminant information, some features in the graphical model and the probabilities they produce are not useful for differentiating objects. Discriminant features might be spread over different levels in the graph.

In this section, we aim at constructing a classifier from the proposed feature hierarchies for the purpose of object class recognition. The general idea is to construct a multi-class SVM classifier from the maximum activation of features obtained during detection (NBP).

The main issue is to convert our graphical models $\mathcal{G}_q \in \mathcal{G}$ to a single input vector $\mathcal{Z}$ for the classifier. To this end, we consider each node $x_i \in \mathcal{G}_q$ of the graphical model of an object q as an element e_i of the SVM input vector. The value of the element e_i will correspond to the maximum activation of the node x_i for the current image. The maximum value is obtained by evaluating the kernel density at each location in the image. This process is repeated for each object q. Finally, we concatenate the vectors $\mathcal{Z}_q = (e_1, \ldots, e_{N^1})^T$ of each object class into a single vector

$$\mathcal{Z} = (\mathcal{Z}_1^T, \mathcal{Z}_2^T, \ldots, \mathcal{Z}_q^T)^T, \tag{8}$$

where the dimensionality of the vector $\mathcal{Z}$ is $\sum_{j=1}^{q} N^j$ and each N^j corresponds to the number of nodes in the graphical model of object j. During recognition, the vectors $\mathcal{Z}_q$ are obtained by processing the input image in each graphical model $\mathcal{G}_q$, thus obtaining q activation vectors $(e_1, \ldots, e_{N^q})^T$.

It has been shown that SVM performance can degrade in high-dimensional spaces with many irrelevant features [16]. One way to bypass this problem is to perform feature selection to eliminate useless features. We employ a conventional feature selection procedure (Algorithm 2) based on the Fisher score. It computes the recognition rate (on the training set) for a set of Fisher score thresholds $\mathcal{T}_i \in \mathcal{T}$. Then it selects the threshold $\mathcal{T}_i$ with the best validation rate.

The Fisher score measures the discriminatory power between two sets of real numbers. Given training vectors $x_{k=1,\ldots,m}$, if the number of positive and negative instances are n_p and n_n, respectively, then the F-score of the ith feature is defined as

$$F(i) = \left| \frac{\mu_i^+ - \mu_i^-}{\sigma_i^+ + \sigma_i^-} \right| \tag{9}$$

where $\mu_i^{\pm}$ is the mean value for the i-th feature in the positive and negative classes, and $\sigma_i^{\pm}$ is the standard deviation.

Algorithm 2 Fisher score for feature selection

```
Calculate F-score of every feature Z_i ∈ Z
T_0 ... T_N ← thresholds on F-scores
for each threshold T_j ∈ T do
    for each Z_i ∈ Z do
        if F-score(Z_i) < T_j then
            remove feature Z_i
        end if
    end for
    {Train, Test} ← randomly split(trainingSet)
    Train a SVM classifier on Train
    R_j ← Calculate the prediction rate on Test
end for
T_s ← Select threshold T_j ∈ T with best rate R_j.
for each F-score(f_i) < T_s do
    remove features f_i
end for
```

6. Experiments

In this section, we apply the proposed method to two very different and challenging applications of object recognition, the *Soccer* [15] and *Butterfly* [5] image databases. The *Soccer* dataset contains 315 images, including 140 for training; the task is to recognize the team membership of soccer players. The *Butterfly* dataset is composed of 619 images, 182 of them for training, acquired from the Internet; the objective is to identify the species. Both datasets comprise seven classes. A wide variety of artifacts (blur, lack of focus, resampling, compression) is noticeably present in both image sets. Moreover, clutter, multiple occurrences, occlusions, large scale variations, viewpoint changes and arbitrary rotations are common in these images (Fig. 4).

In our experiments, a graphical model is learned on the training images of each object class. The hierarchical model contains up to five levels of features and 200 samples are used to approximate each marginal distribution. To specifically evaluate our adaptive patch features (Fig. 5), we also implemented a bag-of-features recognition system similar to Nowak *et al.* [7]. For each image, we count the number of occurrences of each visual word previously extracted with K-Means. An optimal threshold is selected using the mutual information criterion. The classical bag-of-feature framework $\mathbf{B}^-$ is improved by incorporating our adaptive patch features (learned from our hierarchies) in the codebook $\mathbf{B}^+$. To count the occurrences of a given adaptive patch feature, we extract patches at random locations, scales and orientations but using the shape information of the adaptive patch $\mathcal{X}$.

The results obtained by our framework are compared to three, published, state-of-the-art methods [5, 6, 15] in addition to the two bag-of-features systems. On the *Soccer* database, our hierarchical system $\mathbf{H}^+$ clearly outperforms existing approaches thanks to the use of adaptive patch features (Table 1). On the *Butterflies*, our results (Table 2) are comparable to the local affine frames [5].

Soccer Class	$\mathbf{H}^+$	$\mathbf{H}^-$	$\mathbf{B}^+$	$\mathbf{B}^-$	**[6]**	**[15]**
AC Milan	80%	80%	73%	67%	73%	-
Barcelona	93%	93%	93%	87%	93%	-
Chelsea	67%	67%	53%	73%	87%	-
Juventus	93%	87%	93%	80%	67%	-
Liverpool	87%	80%	87%	73%	87%	-
Madrid	87%	87%	87%	80%	93%	-
PSV	67%	60%	60%	60%	47%	-
Total	82%	79%	78%	74%	78%	73%

Table 1. Classification results for the Soccer dataset. We report the results for our feature hierarchy (H) and Bag-of-Features (B) systems with (H^+, B^+) or without (H^-, B^-) adaptive patch features. These are compared to methods based on random subwindows [6] and efficient color features [15].

Butterfly Class	$\mathbf{H}^+$	$\mathbf{H}^-$	$\mathbf{B}^+$	$\mathbf{B}^-$	**[5]**
Admiral	91%	81%	59%	73%	87%
Swallowtail	81%	75%	81%	94%	75%
Machaon	95%	84%	72%	67%	96%
Monarch 1	67%	65%	73%	65%	73%
Monarch 2	84%	79%	85%	69%	91%
Peacock	98%	94%	76%	68%	100%
Zebra	92%	83%	63%	55%	89%
Total	89.4%	83%	71%	68%	90.3%

Table 2. Classification results for the Butterflies dataset. Results are compared to the Local Affine Frames [5].

7. Conclusions and Further work

We described a hierarchical object appearance representation that represents spatial relationships between features by spatial, pairwise potentials in a graphical model (Pairwise Markov Random Field). At each level of the hierarchy, features have an associated appearance. Interestingly, the graphical model formalism allows to pose feature detection as probabilistic inference that we implemented efficiently by NBP [14]. We also showed how to use the feature activations as input to a SVM classifier for object class recognition.

In summary, this paper extends current hierarchical models [1, 2, 3, 4, 8, 11] in two ways;

- Typically, the learning of hierarchical object models often has to deal with a high cost in complexity. A key contribution of our approach is the combination of unsupervised (based on the analysis of co-occurences) and supervised learning methods (SVM). This strategy allows the system to bridge the gap between low-level visual features and object classes more easily.

- Another problem that commonly arises from visual feature hierarchies is the learning of appearance models associated with high-level features. To tackle this difficulty, we introduced *adaptive patch features* that select feature neighborhood dimensions to minimize variance over the training data. Such features present interesting rotational and scale invariance properties and are generic enough to be computed with most current local descriptors.

Our experimental results are on par with or exceed the best published results, and highlight the contribution of our adaptive patch features. These offer rotation and scale-invariance, but affine-invariance should also be possible (by using relative affine shapes). This might enable more accurate recognition from images with large viewpoint variation.

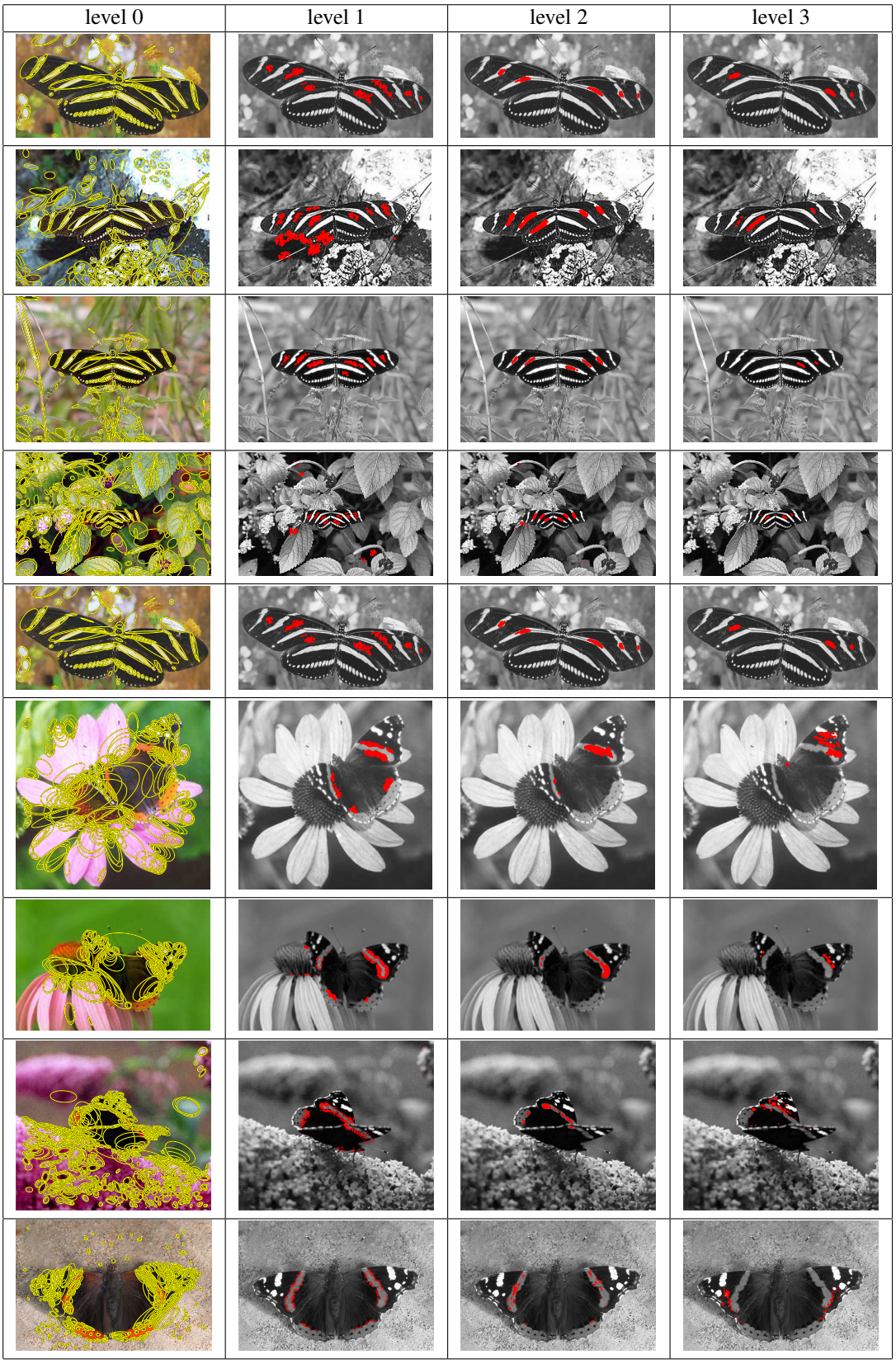
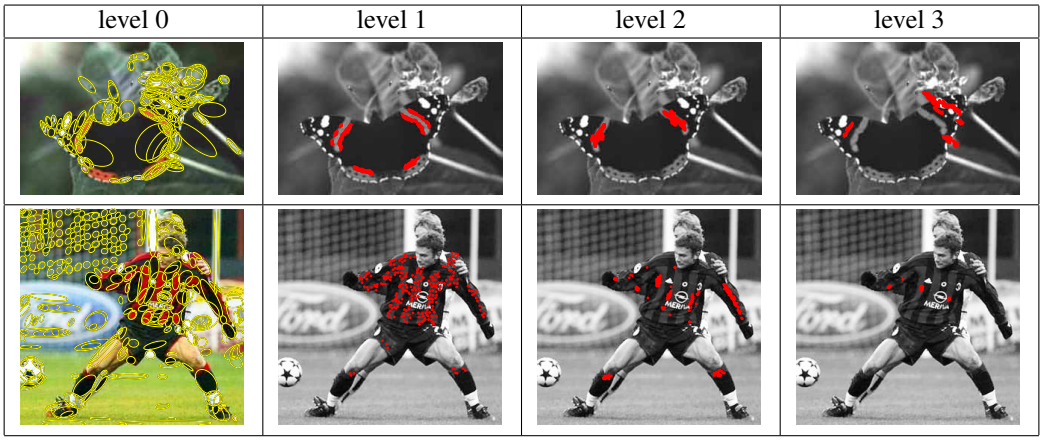


Figure 4. Detection using our hierarchical model ($\mathbf{H}^+$). The first column shows the local regions obtained from a feature detector and available at an observable node y_i of the first level. Each subsequent column to the right shows the final belief (marginal probability) of a higher-level node as a kernel density estimate. Each of them depicts different visual aspects of the object class. The four features shown were chosen such that there exists a path linking them in the hierarchy, i.e., each level-i feature is a child of the level-$i+1$ feature shown in the column to its right.

Figure 4. Continued.



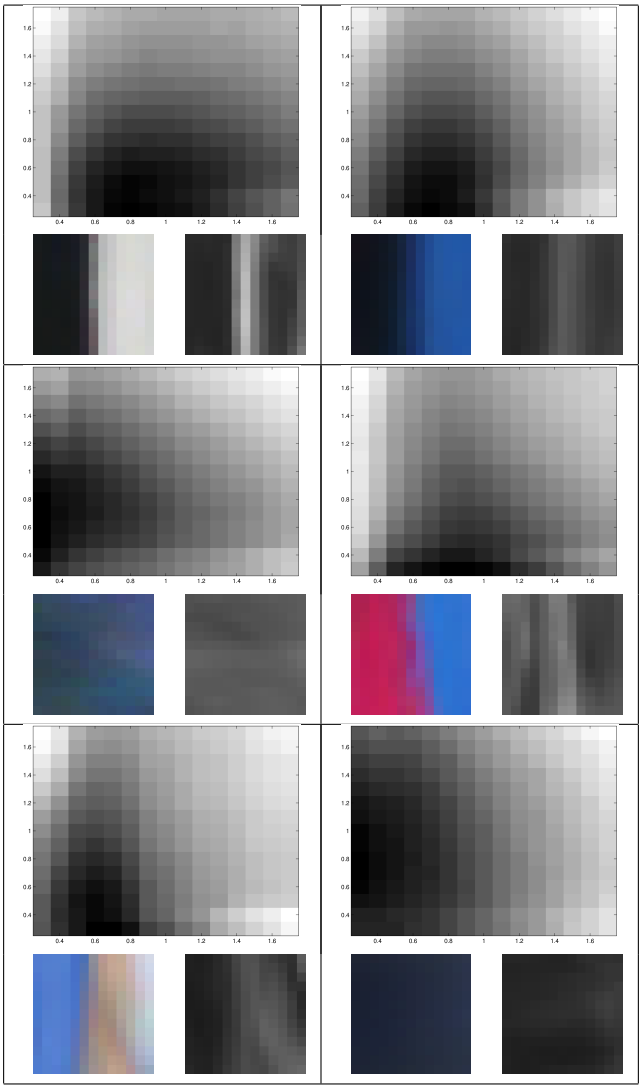


Figure 5. Adaptive Patch Features for different spatial relations. A variance map (over the training set) is shown for each adaptive patch as a function of its shape. The maximum is selected to produce means and variances that are shown on the bottom panels.

References

[1] A. Agarwal and B. Triggs. Hyperfeatures - Multilevel local coding for visual recognition. In *ECCV*, volume 3951 of *Lecture Notes in Computer Science*, pages 30–43. Springer, 2006.

[2] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *CVPR*, pages 710–715, 2005.

[3] B. Epshtein and S. Ullman. Feature hierarchies for object classification. In *ICCV*, pages 220–227, 2005.

[4] S. Fidler, G. Berginc, and A. Leonardis. Hierarchical statistical learning of generic parts of object structure. In *CVPR*, volume 1, pages 182–189, 2006.

[5] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *BMVC*, pages 959–968, 2004.

[6] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *CVPR*, volume 1, pages 34–40, 2005.

[7] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, pages 490–503, 2006.

[8] B. Ommer and J. M. Buhmann. Learning compositional categorization models. *ECCV*, pages 316–329, 2006.

[9] A. Opelt, A. Pinz, M. Fussenegger, and P.Auer. Generic object recognition with boosting. In *PAMI*, volume 28, page 3, March 2006.

[10] P. Perona, R. Fergus, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, page 264, June 2003.

[11] F. Scalzo and J. Piater. Unsupervised learning of dense hierarchical appearance representations. In *ICPR*, volume 2, pages 395–398, August 2006.

[12] G. Schwarz. Estimating the dimension of a model. *Ann. Stat.*, 6(2):461–464, 1978.

[13] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *CVPR*, pages 421–428, 2004.

[14] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *CVPR*, pages 605–612, 2003.

[15] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *ECCV*, volume 2, pages 334–348, 2006.

[16] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. In *NIPS*, 2000.