Optimization framework for learning a hierarchical shape vocabulary for object class detection

Sanja Fidler
http://vicos.fri.uni-lj.si/sanja
Marko Boben
http://vicos.fri.uni-lj.si/markobob
Aleš Leonardis
http://vicos.fri.uni-lj.si/alesl

University of Ljubljana
Faculty of Computer and Information Science
Slovenia

Abstract

This paper proposes a stochastic optimization framework for unsupervised learning of a hierarchical vocabulary of object shape intended for object class detection. We build on the approach by [6], which has two drawbacks: 1.) learning is performed strictly bottom-up; and 2.) the selection of vocabulary shapes is done solely on their frequency of appearance. This makes the method prone to overfitting of certain parts of object shape while losing the more discriminative shape information. The idea of this paper is to cast the vocabulary learning into an optimization framework that iteratively improves the hierarchy as a whole. Optimization is two-fold: one that learns and selects the vocabulary of shapes at each layer in a bottom-up phase and the other that extends/improves it by top-down feedback from the higher layers. The algorithm then loops between the two learning stages several times. We have evaluated the proposed learning approach for object class detection on 11 diverse object classes taken from the standard recognition data sets. Compared to the original approach [6], we obtain a 3 times more compact vocabulary, a 2.5 times faster inference, and a 10% higher detection performance at the expense of 5 times longer training time (25min vs 5min). The approach attains a competitive detection performance with respect to the current state-of-the-art at both, faster inference as well as shorter training times.

1 Introduction

Approaches that learn visual codebooks of appearance [3, 12, 25] and/or shape [4, 19, 23], and combine them with simple object geometry have been shown to give the most successful performance for object class detection to date. Most of these works, however, use flat visual vocabularies where each object is represented as an immediate aggregate of intermediately complex codebook features. Recently, hierarchical approaches have demonstrated appealing computational and qualitative advantages [2, 6, 18, 26]. Hierarchical vocabularies incorporate structural dependencies among the codebook entries at multiple levels: objects are defined in terms of a collection of parts, which are further composed from a set of simpler

BMVC 2009 doi:10.5244/C.23.93

constituents, etc. This 1.) increases the reliability of detections, and 2.) reduces inference time because the features are not only shared among the distinct classes, but are also shared among the features themselves — at multiple layers of the representation.

Learning a hierarchical visual vocabulary in a *bottom-up* and *unsupervised* manner faces several difficult issues. In the bottom-up approach, the codebook features at each layer are formed by combining the features from the layer below. Since no supervision is assumed to point to the object relevant combinations, we must induce the structure from the data itself. In the process of learning we thus deal with a very large number of potential feature aggregations, which can quickly result in a combinatorial explosion already at the early stages of the hierarchical vocabulary construction [22]. Furthermore, the set of features the algorithm learns at each layer directly influences the expressiveness of the layers that work on top of them. If a more discriminative, less frequent, feature is missed in the learning process, we may get poor models for the object classes.

This paper builds upon the approach by Fidler and Leonardis [6] which recursively learns a hierarchical vocabulary of object shape. The representation is *compositional*, i.e., each shape in the vocabulary is composed out of simpler ones by means of spatial relations. The main drawback is, however, that learning is performed strictly bottom-up — once a layer is learned it accepts no further revisions. Performed in this way, object relevant information gets either lost or a very large, possibly redundant set of features must be chosen at each layer in order to compensate for the potential loss. This, however, may result in overfitting certain more generic and less articulated parts of the objects yet may also lose the more discriminative shape information. For object class detection these properties might have negative implications on the performance.

Here we seek for a *compact* hierarchical shape vocabulary while also ensuring it contains *all* object relevant information. The novel idea of this paper is to cast layer learning into a *stochastic optimization framework* that iteratively improves the performance of the hierarchy as a whole. We learn each layer with a two-fold optimization. The *bottom-up phase* learns and selects the layer's vocabulary by maximizing its expressiveness while keeping its complexity low. In the *top-down* phase, the potential lack of expressiveness of the layer above dictates how to extend the lower layer's vocabulary in order to improve the higher layer performance. The algorithm then iterates between the two learning stages several times to learn each layer sufficiently well. While the optimization algorithm is general and could be applied over several layers simultaneously, the optimization over each two consecutive layers has yielded satisfactory results. To learn the representation for an object class, we assume supervision in terms of a positive and validation set of class images — however, we learn the hierarchical shape vocabulary for the classes in a completely *unsupervised* way (no labels on object parts and smaller constituents are assumed).

We have evaluated the proposed learning approach for object class detection on 11 object classes. Compared to the original approach [6], we obtain an about 3 times more compact vocabulary, a 2.5 times faster inference, and a 10% higher detection performance at the expense of 5 times longer training time. The approach attains a competitive detection performance with respect to the current state-of-the-art at faster inference as well as shorter training times.

2 Related work

Our main contribution is *unsupervised learning* of a hierarchical shape vocabulary, thus we briefly review the related literature on this topic.

Unsupervised hierarchical learning of object structure. Work on unsupervised learning of compositional hierarchies has been relatively scarce. Most unsupervised approaches fall under the domain of neural networks [20], which are conceptually very different from compositional representations [10]. Slightly more related, the HMAX approach [16, 22] builds only 2 layers by choosing the combinations randomly as they appear (no statistical learning is employed). Epshtein and Ullman [26] learn the hierarchical vocabulary by recursively *decomposing* the class-specific patches into smaller ones. This is converse to the process of *composition* taken here. An iterative bottom-up and top-down learning has been proposed by Hinton [11], however, the author proposes a generative model of the *appearance* of images (which is rigid with respect to shape deformations), whereas our aim here is to learn a generative model of the *shape* of the objects.

The learning frameworks most related to ours include [6, 8, 18, 21] and just recently [28]. However, most of these methods learn the features only in a bottom-up manner. Zhu et al. [28] first learn the whole hierarchy in a bottom-up pass but suggest the use of a top-down stage in which the missed parts such as the legs of a horse are added to the representation afterwards. In our work, the vocabulary at each layer is improved simultaneously by iterating between the bottom-up and top-down stage within the learning process.

The works on learning object taxonomies [14, 17, 27] perform detection by hierarchical cascade of classifiers which inherently differs from the generative approach taken here.

3 The hierarchical shape vocabulary

We first present the hierarchical framework which is mainly adopted from [6] but extended to a probabilistic formulation. Our novel iterative learning approach is presented in Section 4.

The hierarchical vocabulary. The vocabulary at each layer contains a set of *shape models* or *compositions*. Each shape model in the hierarchy has a sparse, star-shaped topology, and is modeled as a conjunction of a small number of *parts* (shapes from the previous layer). Each part is spatially constrained on the parent composition via a spatial relation which we model with a two-dimensional Gaussian. The number and the type of parts can be different for each shape model and is learned from the data without supervision. The definition is recursive, where each part is similarly composed of simpler subparts. At the lowest layer, the hierarchical vocabulary consists of a small number of short contour fragments at coarsely defined orientations. The vocabulary at the top-most layer contains compositions that code the whole shapes of the objects. We emphasize that each such composition does not code only the shape of one specific object, but exerts a certain degree of intra-class invariance.

The hierarchical vocabulary $\mathscr{V} = (V, E)$ is represented with a directed graph, where multiple edges between two vertices are allowed. The vertices V of the graph represent the model shapes and the edges E represent the composition relations between them. The graph $\mathscr{V}$ has a hierarchical structure, where the set of vertices V is partitioned into subsets $V^1, \dots, V^o$, each denoting the shapes at a particular layer. The vertices $v_i^1 = \psi_i = i\frac{\pi}{n}$, $i = 0, \dots, n-1$, at the lowest layer V^1 represent the n oriented contour fragments. The vertices at the top-most layer V^o, which will be referred to as the *object layer*, code the whole shapes of the objects. Each object class C is assigned a subset of vertices $V_C \subseteq V^o$ that code the shapes of that particular class. We will denote the set of edges between the vertex layers V^ℓ and $V^{\ell-1}$ with E^ℓ. Each edge $e_{Ri}^\ell = v_R^\ell v_i^{\ell-1}$ in E^ℓ is associated with the Gaussian parameters $\theta_{Ri}^\ell := \theta(e_{Ri}^\ell) = (\mu_{Ri}^\ell, \Sigma_{Ri}^\ell)$ of the corresponding spatial relation between the parent shape v_R^ℓ and its constituent shape $v_i^{\ell-1}$. We will use $\boldsymbol{\theta}_R^\ell = (\theta_{Ri}^\ell)_i$ to denote the vector of all the param-

eters of a particular shape model. The number of constituent shapes of v_R^ℓ will be denoted with $d(v_R^\ell)$. The pair $\mathscr{V}^\ell := (V^\ell, E^\ell)$ will be referred to as the *vocabulary at layer* ℓ.

The shapes at the lowest layer V^1 (oriented contour fragments) are the only ones defined in advance — the whole subsequent structure and the parameters of the vocabulary $\mathscr{V}$ are *learned* without supervision. The number of layers is also not set in advance.

Inference. Let I denote a query image in which we want to infer all (modeled) class instances. Inference is performed at several scales of I, but for the ease of exposition we will only consider one scale. We first convolve the image with 6 oriented Gabor kernels and find locations $\mathbf{X}$ of local maxima of the Gabor energy as proposed in [7]. In each point $x \in \mathbf{X}$ we extract a Gabor feature vector $\mathbf{f}$ containing the outputs of the 6 filters in point x. This gives us a set of 6-dimensional feature vectors $\mathbf{F} = \{\mathbf{f}\}$ at locations $\mathbf{X}$. The obtained set $(\mathbf{F}, \mathbf{X})$ represents the observations upon which we perform the inference.

In the process of inference we build an (directed) *inference graph* $\mathscr{G} = \mathscr{G}(I) = (Z, Q)$. The vertices Z are partitioned into vertex layers 1 to o (object layer), $Z = Z^1 \cup \cdots \cup Z^o$, and similarly also the edges $Q = Q^1 \cup \cdots \cup Q^o$. Each vertex $z^\ell = (v^\ell, x^\ell) \in Z^\ell$ represents a *hypothesis* that a particular shape $v^\ell \in V^\ell$ from the vocabulary is present at location x^ℓ. The edges in Q^ℓ connect each parent hypothesis z_R^ℓ to all of its part hypotheses $z_i^{\ell-1}$. The edges in the bottom layer Q^1 connect the hypotheses in the first layer Z^1 with the observations. With $\mathscr{S}(z)$ we denote the subgraph of $\mathscr{G}$ that contains the vertices and edges of all descendants of z. The set of all descendants of z at vertex layer Z^1 will be referred to as the *support* of a hypothesis z and denoted with $\mathrm{supp}(z)$.

Since our definition of each vocabulary shape model assumes that its parts are independent given the parent structure, we can calculate the likelihood of the hypotheses $z_i^{\ell-1} = (v_i^{\ell-1}, x_i^{\ell-1})$ under a hypothesis $z_R^\ell = (v_R^\ell, x_R^\ell)$ by taking a product over the individual compatibilities (spatial constraints) of its parts:

$$p(\mathbf{v}^{\ell-1}, \mathbf{x}^{\ell-1} \mid v_R^\ell, x_R^\ell, \boldsymbol{\theta}_R^\ell) = \prod_{e_{Ri}^\ell = v_R^\ell v_i^{\ell-1}} p(x_i^{\ell-1} \mid x_R^\ell, v_R^\ell, v_i^{\ell-1} \theta_{Ri}^\ell). \tag{1}$$

We will use p_{Ri} instead of $p(x_i^{\ell-1} \mid x_R^\ell, v_R^\ell, v_i^{\ell-1} \theta_{Ri}^\ell)$ to abbreviate the notation. The term p_{Ri} stands for the spatial constraint between the parent z_R^ℓ and its constituent part $z_i^{\ell-1}$. It is modeled by a normal distribution, $p_{Ri} = \mathscr{N}(x_i^{\ell-1} - x_R^\ell \mid \theta_{Ri}^\ell)$, where $\theta_{Ri}^\ell = (\mu_{Ri}^\ell, \Sigma_{Ri}^\ell)$. If the likelihood in (1) is above a threshold, we make edges between z_R^ℓ and its parts $z_i^{\ell-1}$.

The log-likelihood of the data under a shape hypothesis z_R^ℓ is then computed as:

$$\log p(\mathbf{F}, \mathbf{X}, \mathbf{z}^{1:\ell-1} \mid z_R^\ell; \mathscr{V}) = \sum_{z_{R'} z_{i'} \in E(\mathscr{S}(z_R^\ell))} \log p_{R'i'} + \sum_{\substack{z_{i'}^1 = (\psi_{i'}, x_{i'}^1) \\ z_{i'}^1 \in V(\mathscr{S}(z_R^\ell))}} \log p(\mathbf{F}, \mathbf{X} \mid z_{i'}^1). \tag{2}$$

This is simply obtained by a recursive application of (1), which assumes that the subgraph $\mathscr{S}(z_R^\ell)$ of a shape hypothesis z_R^ℓ has a tree structure (learning [6] ensures this). For the data term $p(\mathbf{F}, \mathbf{X} \mid z_{i'}^1)$ we use the probabilistic model of local contour orientations based on the Gabor filter responses, $p(\mathbf{f} \mid \psi)$, as proposed in [13].

4 Iterative bottom-up and top-down vocabulary learning

The original idea of [6] is to find a vocabulary of shape models that well represent the distribution of the spatial layouts of contour fragments inside local neighborhoods (*receptive*

fields or RFs). The sizes of the RFs are increased with each layer and part configurations are learned to explain larger and larger image areas. In the top layer the RF covers the whole image (object). In [6] the RFs were assumed independent, i.e. for each RF the frequency of the composition that best explained its contour content was updated and the method finally selected the set of most frequently occurring compositions. Since there is a huge variety of local shape configurations, the number of compositions quickly increases to a large number which makes further learning of combinations difficult (as also seen in [20, 22]).

Here we exploit the fact that the top, object-layer models will have a tree structure, meaning that they are composed of disjunct parts at each layer. This means that we do not need our vocabulary shapes to explain each of the RFs in an image, but must only be able to explain all the contour fragments in an image in a global sense — as a union of a few matched compositions.

The following sections present our novel iterative vocabulary learning approach. We first present the theoretical framework and propose the bottom-up and top-down learning phases in the following subsections.

4.1 Theoretical framework

Our goal is to find a hierarchical vocabulary $\mathscr{V}$ that well represents the distribution $p(I \mid C) \approx p(\mathbf{F}_I, \mathbf{X}_I \mid C; \mathscr{V})$ at minimal complexity of inference, where C denotes the class variable. Specifically, we seek for a vocabulary $\mathscr{V} = \cup_\ell \mathscr{V}^\ell$ that optimizes the function f over the data $D = \{(\mathbf{F}_n, \mathbf{X}_n, C_n)\}_{n=1}^N$ (N training images):

$$\mathscr{V}^* = \arg\max_{\mathscr{V}} f(\mathscr{V}) \qquad \text{where} \qquad f(\mathscr{V}) = L(D \mid \mathscr{V}) - \lambda \cdot T(D, \mathscr{V}) \tag{3}$$

The first term represents the log-likelihood:

$$L(D \mid \mathscr{V}) = \sum_{n=1}^{N} \log p(\mathbf{F}_n, \mathbf{X}_n \mid C; \mathscr{V}) = \sum_{n=1}^{N} \log \sum_{\mathbf{z}} p(\mathbf{F}_n, \mathbf{X}_n, \mathbf{z} \mid C; \mathscr{V}) \tag{4}$$

The second term $T(D, \mathscr{V})$ in (3) penalizes the complexity of the model. We define it as the complexity needed to match the vocabulary $\mathscr{V}$ against the images:

$$\begin{aligned} T(D, \mathscr{V}) &= \sum_{n=1}^{N} \sum_{\ell=2}^{o} T^\ell(D_n; \mathscr{V}^\ell), \\ T^\ell(D_n; \mathscr{V}^\ell) &= \sum_{(v_i^{\ell-1}, x) \in Z_n^{\ell-1}} \sum_{v_R^\ell v_i^{\ell-1} \in E^\ell} d(v_R^\ell), \end{aligned} \tag{5}$$

where $d(v_R^\ell)$ denotes the number of parts of v_R^ℓ. In the first sum in (5) we assume we have obtained the inference graph $\mathscr{G} = (Z_n, \mathcal{Q}_n)$ under $\mathscr{V}$ from the n-th image. Here $T(D, \mathscr{V})$ roughly corresponds to the number of operations needed to perform inference [6].

The parameter λ controls the amount of penalization by the complexity term. If λ is small, then f prefers large vocabularies with frequent shapes, whereas the higher values of λ penalize the redundancy of the shapes in the vocabulary. We set λ by experimentation.

Since optimizing (3) is intractable, we choose to learn $\mathscr{V}$ sequentially, layer by layer. At each step, we will find $\mathscr{V}^\ell$ that maximizes f defined in (3) where we assume that the class layer $\mathscr{V}^o$ is directly above layer ℓ, i.e. $o = \ell + 1$. For the complexity term $T(D, \mathscr{V})$ we must

make a prediction about the value of $T^o(D;\mathscr{V}^o)$:

$$T^o(D_n;\mathscr{V}^o) = |Z_n^\ell| \cdot \frac{|\mathbf{F}|}{|\overline{\text{supp}}^\ell|}. \tag{6}$$

This term represents a rough estimate of the cost needed to infer a class layer o based on $\mathscr{V}^\ell$ from images. To describe a class C with the shapes from $\mathscr{V}^\ell$ in an image I, our class representation would need to use at least $|\mathbf{F}|/|\overline{\text{supp}}^\ell|$ shapes. Here $|\mathbf{F}|$ denotes the number of all features vectors in I and $|\overline{\text{supp}}^\ell|$ denotes the average size of the support of shape hypotheses from layer ℓ. Thus $|\mathbf{F}|/|\overline{\text{supp}}^\ell|$ stands for the size of the partition of the hypothesis layer Z^ℓ into hypotheses with disjoint supports, the number of which denotes an estimate for the number of constituent shapes $d(C)$ of class C. The term $|Z_n^\ell|$ assumes that a class representation would need to be matched at each hypothesis in Z_n^ℓ.

The problem when optimizing the vocabulary at each layer separately (the *bottom-up phase*), is that the local solution does not necessarily lead to the globally optimal solution. A vocabulary that is well expressive but very small at one layer, may, when further combined, result in a vocabulary of lower expressiveness in the layers above, and thus give us poor top-level models of object class shapes. We thus introduce the *top-down phase* of learning, in which we revise the vocabulary at each layer as to maximally improve the expressiveness of vocabulary at the layer above.

In the bottom-up phase, we first learn a large set of candidate shape models at each layer as in [6]. Among these we select a subset of shapes which optimize f in (3) by using the greedy algorithm. We further improve this selection by performing stochastic optimization. In the top-down phase, we improve the vocabulary at each layer by feedback from the layer above. We explain the steps in the following subsections.

4.2 Bottom-up learning phase

When learning the vocabulary $\mathscr{V}^\ell = (V^\ell, E^\ell)$ at layer ℓ, we assume that for each training image I we have the inference graph $\mathscr{G}_I = (Z^1 \cup \cdots \cup Z^{\ell-1}, Q)$ built up to layer $\ell-1$. To learn a large set of candidate shape models we use the algorithm by [6]. It gives us a temporary vocabulary $\mathscr{V}_*^\ell$ which contains shapes that optimize the log-likelihood $L(D \mid \mathscr{V}^\ell)$, where, however, D is a set of RFs collected around each point in an image. From $\mathscr{V}_*^\ell$ we will select a subset of shapes which optimize the global function f as defined in (3).

Greedy selection. We select a subset of the shape models from $\mathscr{V}_*^\ell$ using a greedy approach. At each step of the iteration, we select a shape from $\mathscr{V}_*^\ell$ that maximally increases the score f defined in (3). Learning stops when the best scoring shape falls below a pre-defined threshold. We denote the selected shapes with $\mathscr{V}_g^\ell$.

Stochastic optimization. We further employ a stochastic MCMC algorithm to get the final vocabulary $\mathscr{V}^\ell$ at layer ℓ. The first state of the Markov chain is the vocabulary $\mathscr{V}_g^\ell$ obtained with the greedy selection. Let $\mathscr{V}_t^\ell$ denote the vocabulary at the current state of the chain. We either exchange/add/remove one shape model from $\mathscr{V}_t^\ell$ with another one from $\mathscr{V}_*^\ell \setminus \mathscr{V}_t^\ell$ to get the vocabulary $\mathscr{V}_{t+1}^\ell$. The vocabulary $\mathscr{V}_{t+1}^\ell$ is accepted as the next state of the Markov chain with probability

$$\min(1,\ \alpha^{f(\mathscr{V}_{t+1})-f(\mathscr{V}_t)}),\quad \alpha > 1 \tag{7}$$

according to the Metropolis-Hastings algorithm.

The vocabulary at layer ℓ is then defined as the $\mathscr{V}^\ell$ producing the maximal value $f(\mathscr{V})$, after running several iterations of the M-H algorithm. We usually perform 100–200 steps.

4.3 Top-down learning phase

The idea behind the top-down phase is to improve the vocabulary at layer ℓ in order to increase the value of f under layer $\ell+1$. This procedure is summarized as follows. *Step 1*. We first find the critical data points under the current layer $\ell+1$. *Step 2*. We re-learn layer ℓ based on the critical points. *Step 3*. We re-learn layer $\ell+1$ and repeat the process.

Step 1. We first define the *critical points* in each image under the current layer $\ell+1$. These are the points at which the ratio Δ of the likelihoods

$$\Delta(\mathbf{F},\mathbf{X}) = \frac{\sum_{\mathbf{z}\in Z^{1:\ell+1}} p(\mathbf{F},\mathbf{X},\mathbf{z} \mid C, \mathscr{V}^{1:\ell+1})}{\sum_{\mathbf{z}\in Z^{1:\ell}} p(\mathbf{F},\mathbf{X},\mathbf{z} \mid C, \mathscr{V}^{1:\ell})} \tag{8}$$

is low. This ratio tells us how well an observation in $(\mathbf{F},\mathbf{X})$ is explained under the vocabulary $\mathscr{V}^{1:\ell+1}$ relative to how well it is explained under $\mathscr{V}^{1:\ell}$. Note that $\Delta \in [0,1]$ and we set $\Delta(f,x)=1$ when the denominator in (8) is 0. From $(\mathbf{F},\mathbf{X})$ we select the points according to the distribution $1-\Delta$, i.e. each point $(f,x) \in (\mathbf{F},\mathbf{X})$ is sampled with probability $1-\Delta(f,x)$. This heuristics has worked well in our experiments. The selected points are used as the data D' for learning new shapes at layer ℓ. If $\Delta(f,x)=0$, it means that (f,x) is not explained under the current vocabulary at layer $\ell+1$ and will be added to D' with probability 1. On the other hand, $\Delta(f,x)=1$ means that (f,x) is explained equally well by layer ℓ and $\ell+1$ and will not be used in the re-learning process.

Step 2. In this step, we learn a new set of shape models $\mathscr{V}_*^\ell$ using the algorithm by [6], however, we only need to consider those RFs that contain points (contour fragments) from D'. Specifically, we use only those RFs in which at least half of its points are from D'. The shapes in $\mathscr{V}_*^\ell$ will thus be optimized to explain the critical points. From the joint vocabulary $\mathscr{V}^\ell \cup \mathscr{V}_*^\ell$ we need to select a subset of shapes which will produce an overall improved score $f(\mathscr{V}^{1:\ell+1})$ of the layer above. Therefore we perform the optimization from the previous subsection on the joint vocabulary $\mathscr{V}^\ell \cup \mathscr{V}_*^\ell$ over the original data D. To allow the algorithm to select the shapes also from $\mathscr{V}_*^\ell$ we must choose a different (smaller) λ_* when evaluating f for layer ℓ. We sample λ_* from $(0,\lambda]$ randomly with respect to a uniform distribution.

Step 3. Running the optimization from Section 4.2 gives us a new layer $\mathscr{V}_{t+1}^\ell$ which we then use to learn a new layer $\mathscr{V}_{t+1}^{\ell+1}$. The new vocabulary $\mathscr{V}_{t+1}^{\ell:\ell+1}$ is accepted with probability $\min(1,\ \alpha^{f(\mathscr{V}_{t+1})-f(\mathscr{V}_t)})$. Based on the new layers ℓ and $\ell+1$, the complete top-down phase is repeated several times and the best scoring vocabulary $\mathscr{V}^{\ell:\ell+1}$ is finally chosen.

In principle, we could optimize more than two layers simultaneously, however, the two-consecutive-layer optimization has turned out to be sufficient in our experiments.

4.4 Learning of the object shapes

The complete hierarchical vocabulary is learned by performing bottom-up and top-down optimization at each layer. Learning stops when no more layers are formed (no further combination of shapes increases the value of f). In our experiments, learning for all the classes stopped at layer 6. The shape models $\{v^o\}$ learned at the top layer are expected to code the whole shapes of the objects. We additionally perform cross-validation of the learned object models using an image validation set. Specifically, we impose a selection that discards those v^o that yield too many false positives on the validation images.

5 Experimental results

The approach was tested on 11 diverse object classes from the standard recognition datasets: Apple logo, bottle, giraffe, mug and swan from the ETH shape data set [5], bicycle_side, car_front, and cow_side from GRAZ [19], INRIA horses [5], TUD motorbike [12], and UIUC multi-scale car_side. These datasets pose a significant challenge due to the high amount of clutter, the great scale differences of objects and their significant intra class variability. All of the experiments were performed on one core of an Intel Xeon-4 CPU 2.66 Ghz. The algorithm is implemented in C++.

Examples of the learned shapes (for several classes) are depicted in Fig. 1. Each shape is composed of a few shapes from the previous layer, but the spatial relations are not shown.

When evaluating the detection performance, a detection is counted as correct, if the predicted bounding box b_{fg} coincides with the ground truth b_{gt} more than 50%. On ETH and INRIA datasets, this threshold is lowered to 0.3 to enable a fair comparison with the related work [4]. The performance is given either with recall at equal error rate or positive detection rate at low FPPI, depending on the type of results reported on these datasets thus-far.

Training time. It takes, for example, 23 minutes to train on Apple logo, 25 for bottle, 31 for giraffe, 31 for mugs, 17 for swans, 25 for cow, and 35 for the horse class. Training on 50 horses (as in our case) takes roughly 2 hours in [23] (C# implementation, 2.2 GHz machine).

Inference time. On average, the inference takes from $2-4$ seconds per image. For example, inference takes 3.6 seconds for Apple logos, 3.4 for bottles, 3.2 for giraffes, 3.6 for mugs, and 1.9 seconds for swans. To compare (in sec) with other object class detection approaches: [24]: $20-30$, [28]: 16.9, [9]: 20, [3]: $12-18$, on somewhat older hardware.

Detection performance. The ETH experiments are performed in a 5-fold cross-validation obtained by sampling 5 subsets of half of the class images at random as in [9]. Training of the hierarchy was performed only within the given ground truth bounding boxes. The test set for evaluating detection consists of all the remaining images in the dataset. The detection performance is given as the detection rate at the rate of 0.4 false-positives per image (FPPI), averaged over the 5 trials. Similarly, the results on INRIA horses are reported by sampling 5 subsets of 50 class images at random and using the remaining 120 for testing. The test set also includes 170 negative images to allow for a higher FPPI rate. The detection performance is reported in Table 1, with a few example detections shown in Fig. 4. With respect to the most related approach [4] we achieve a better performance on all the categories, most notably on giraffes (24.7%). On the average, our method performs comparably to [9] who employ a fully discriminative framework in contrast to the generative approach taken here.

For the other classes we report the recall at EER in Table 1. On the GRAZ classes and TUD motorbikes our performance is comparable to the current state-of-the-art results. Note, however, that both, [15] and [12], used over 150 training examples of motorbikes, while we only used 50. We also achieve a better performance on UIUC cars with respect to the layer 3 + voting for center as reported by the original approach by [6].

Comparison with baseline. To further utilize our learning approach we compare it against the baseline ([6] — no optimization) on 3 object classes (cow, mug and giraffe). We additionally compare it with the one where no top-down optimization is performed. We compare the sizes of all three learned representations (*baseline* vs *no top-down opt* vs *full optimization*) in Fig. 2. It can be observed that the hierarchy obtained by the baseline approach is roughly 3 times larger than the optimized hierarchy. This reflects in inference time which is shown in Fig. 3(a). On the average, inference time for the baseline is 2.6 times higher than that of our approach. However, since the baseline only performs one bottom-up pass in

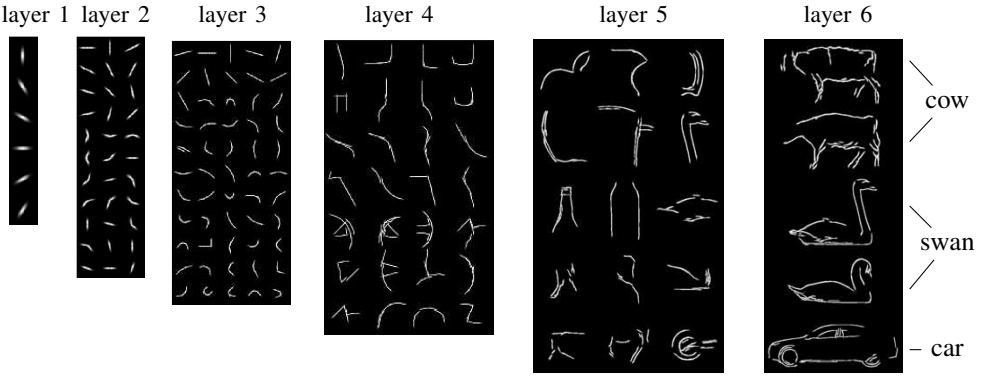

Figure 1: Example shapes in the learned hierarchical vocabulary. Each shape in the hierarchy is a composition of a few shapes from the layer below. Each shape also models spatial relations between its parts, which are not shown — only the mean of each shape is depicted.

learning, its training time is significantly faster, see Fig. 3(b). The detection rates are shown in Fig. 3(c), which demonstrate that our approach is by 10% superior over the baseline. This is likely because the baseline only selects the features by frequency of appearance and is thus prone to missing out on the more discriminative features. Consequently, it does not separate the foreground objects from a more generic background sufficiently well. While the *no top-down opt* approach outperforms the baseline, it still does not reach up to the performance of the full optimization approach.

Table 1: Detection results. *Left:* Average detection-rate (in %) at 0.4 FPPI for ETH and INRIA datasets. *Right:* Recall at EER (%) for the GRAZ, TUD, and UIUC datasets.

		[2]	[3]	**our appr.**
ETH shape	apple	83.2 (1.7)	89.9 (4.5)	87.3 (2.6)
	bottle	83.2 (7.5)	76.8 (6.1)	**86.2** (**2.8**)
	giraffe	58.6 (14.6)	90.5 (5.4)	83.3 (4.3)
	mug	83.6 (8.6)	82.7 (5.1)	**84.6** (**2.3**)
	swan	75.4 (13.4)	84.0 (8.4)	78.2 (5.4)
	avg.	76.8	84.8	83.7
INRIA	horse	84.8 (2.6)	/	**85.1** (**2.2**)

		related work		**ours**
GRAZ	bicycle	72 [13]	67.9 [23]	68.5
	bottle	91 [13]	90.6 [23]	89.1
	cow	100 [13]	98.5 [23]	96.9
	carfront	90 [13]	70.6 [23]	76.5
	mug	93.3 [13]	90 [23]	90
TUD	mbike	87 [12]	88 [15]	83.2
UIUC	car	93.5 [1]	92.1 [6]	**93.5**

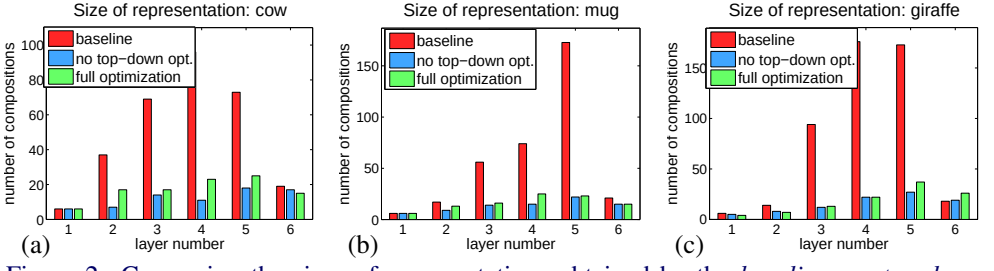

Figure 2: Comparing the sizes of representations obtained by the *baseline*, *no top-down optimization*, and our *full optimization* approach for the (a) cow, (b) mug, and (c) giraffe.

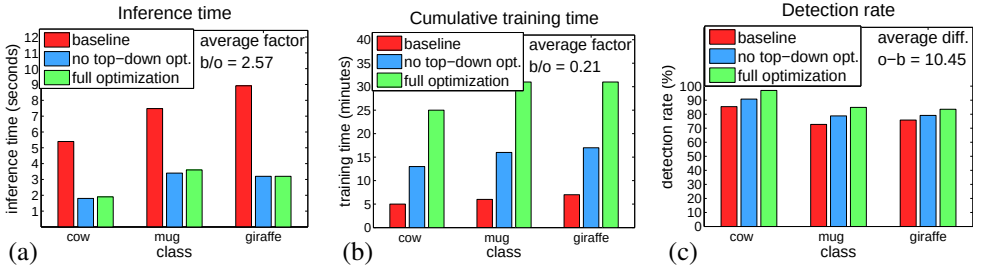

Figure 3: A comparison of (a) inference, and (b) training times, and (c) detection rates for the *baseline*, *no top-down optimization*, and our proposed *full optimization* approach.

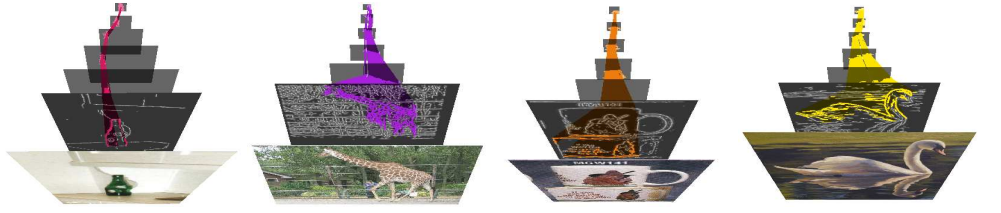

Figure 4: Example detections on the ETH shape database. The links are the edges of the subgraphs $\mathscr{S}(z^o)$ of the object hypotheses z^o and are color-coded to denote different classes.

6 Summary and conclusions

In this paper, we presented a stochastic optimization approach to learning a compact hierarchical shape vocabulary for object class detection. The optimization iterates between the bottom-up and top-down learning stages, optimally revising the individual layers.

We have evaluated the approach on 11 diverse object classes and demonstrated the advantages in terms of speed of inference and detection performance over the previous approach [6] as well as the current state-of-the-art methods.

Acknowledgment

This research has been supported in part by the following funds: EU FP7-215843 project POETICON, EU FP7-215181 project CogX, Research program Computer Vision P2-0214 and Project J2-2221 (Slovenian Research Agency).

References

[1] N. Ahuja and S. Todorovic. Connected segmentation tree — a joint representation of region layout and hierarchy. *IEEE CVPR*, 2008.

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

[3] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3): 273–303, March 2007.

[4] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *IEEE CVPR*, 2007.

[5] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, 30(1):36–51, 2008.

[6] S. Fidler and A. Leonardis. Towards scalable representations of visual categories: Learning a hierarchy of parts. In *IEEE CVPR*, 2007.

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

[8] F. Fleuret and D. Geman. Coarse-to-fine face detection. *International Journal of Computer Vision*, 41(1/2):85–107, 2001.

[9] M. Fritz and B. Schiele. Decomposition, discovery and detection of visual categories using topic models. In *IEEE CVPR*, 2008.

[10] S. Geman, D. Potter, and Z. Chi. Composition systems. *Quarterly of Applied Mathematics*, 60(4):707–736, 2002.

[11] G. E. Hinton. Learning multiple layers of representation. *Trends in Cognitive Sciences*, 11(10):428–434, 2007.

[12] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.

[13] N. Lüdtke and R. C. Wilson. A mixture model for population codes of gabor filters. *IEEE Transactions on neural networks*, 14(4):794–803, 2003.

[14] M. Marszałek and C. Schmid. Constructing category hierarchies for visual recognition. In *ECCV*, volume IV of *LNCS*, pages 479–491. Springer, 2008.

[15] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *IEEE CVPR*, pages 26–36, 2006.

[16] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *IEEE CVPR*, pages 11–18, 2006.

[17] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE CVPR*, volume 2, pages 2161–2168, 2006.

[18] B. Ommer and J. M. Buhmann. Learning the compositional nature of visual objects. In *IEEE CVPR*, 2007.

[19] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision*, 80(1):16–44, 2008.

[20] M. A. Ranzato, F.-J. Huang, Y-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *IEEE CVPR*, 2007.

[21] F. Scalzo and J. H. Piater. Statistical learning of visual feature hierarchies. In *Workshop on Learning, CVPR*, 2005.

[22] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Analysis and Machine Intell.*, 29 (3):411–426, 2007.

[23] J. Shotton, A. Blake, and R. Cipolla. Multi-scale categorical object recognition using contour fragments. *IEEE Trans. Pattern Analysis and Machine Intell.*, 30(7):1270–1281, 2008.

[24] S. Todorovic and N. Ahuja. Unsupervised category modeling, recognition, and segmentation in images. *IEEE Trans. Pattern Analysis and Machine Intell.*, 2009.

[25] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, 29 (5):854–869, 2007.

[26] S. Ullman and B. Epshtein. *Visual Classification by a Hierarchy of Extended Features.* Towards Category-Level Object Recognition. Springer-Verlag, 2006.

[27] P. Zehnder, E. Koller Meier, and L.J. Van Gool. An efficient shared multi-class detection cascade. In *BMVC*, 2008.

[28] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *ECCV*, volume 2, pages 759–773, 2008.