# Similarity-based cross-layered hierarchical representation for object categorization [*]

Sanja Fidler Marko Boben Aleš Leonardis
Faculty of Computer and Information Science
University of Ljubljana, Slovenia
{sanja.fidler, marko.boben, ales.leonardis}@fri.uni-lj.si

## Abstract

*This paper proposes a new concept in hierarchical representations that exploits features of different granularity and specificity coming from all layers of the hierarchy. The concept is realized within a cross-layered compositional representation learned from the visual data. We show how similarity connections among discrete labels within and across hierarchical layers can be established in order to produce a set of layer-independent shape-terminals, i.e.* shapinals. *We thus break the traditional notion of hierarchies and show how the category-specific layers can make use of* all *the necessary features stemming from* all *hierarchical layers. This, on the one hand, brings higher generalization into the representation, yet on the other hand, it also encodes the notion of scales directly into the hierarchy, thus enabling a multi-scale representation of object categories. By focusing on shape information only, the approach is tested on the Caltech* 101 *dataset demonstrating good performance in comparison with other state-of-the-art methods.*

## 1. Introduction

Visual categorization and recognition of objects has been a subject of extensive research over the last decades. Many approaches have been developed that perform well on this challenging task [15, 17, 13, 9, 4, 19, 1, 12, 6, 22, 16, 3, 20]. However, the most successful methods up-to-date are mainly flat appearance-based systems that combine masses of discriminative extracted features with standard classifiers. It has not been until recent years that the computationally more plausible hierarchical systems have been proven to also reach state-of-the-art classification results [13, 18, 10, 14, 16].

Hierarchical systems ensure an efficient way to represent exponential variability present in the visual data. What can otherwise be only complemented by extracting millions of image patches or other highly discriminative features, hierarchies model within a relatively small and compact representation, enabling more robust detection strategies [2, 8, 11, 7]. Since hierarchical approaches offer computational means to address visual tasks on a larger scale, it is worth studying their design principles in greater depth.

The design of hierarchical representations to this end still poses many open questions, *i.e.* what is the best internal representation of the hierarchical nodes, to what extent unsupervised learning should and can be used, and how should the lower, statistics-driven layers be organized in order to serve as a good basis for higher-level object representations.

A number of architectures that tackle the creation of the lower, category-independent layers have been proposed. However, in most hierarchical categorization approaches, categorical representations and decisions are based only upon the top-most layer [13, 10, 16, 7]. We argue that this places a limit on the performance of hierarchies, giving rise to the problem of "terminal" structures in images as well as terminal structures coded in hierarchical nodes. For example, high curvatures or circle-detecting features do not usually statistically combine into more complex, higher-layer units, but rather *terminate* at a particular layer. In images, this is also true for even simpler features such as lines: a smaller object contains shorter lines that can be extracted with lower layers, while only a longer line detector (emerging from higher layers that usually code larger receptive fields) becomes sufficiently discriminative on a larger object. Since objects can appear on any scale and contain features of various granularities, it becomes crucial to combine features from *all* hierarchical layers pertaining to the corresponding shape terminals, e.g. *shapinals*, to produce the final description of objects.

In literature, relatively few strategies have been proposed and relatively few experiments performed that tried to determine the proper granularity of features (layers) to be used for the final classification [21]. Serre et al. [18] designed

[*] This research has been supported in part by: Research program P2-0214 (RS), EU FP7-215843 project POETICON, EU FP6-004250-IP project CoSy, and EU FP6-511051-2 project MOBVIS.

a two-layer hierarchy consisting of a set of Gabor filters at the bottom, $C1$ layer, and a learned set of features on the second, $C2$ layer. Their findings suggest that while the $C2$ features usually outperform the simple $C1$ ones, in some cases the base features turn out to be more useful for classification. On this basis, Wolf et al. [21] experimented with various concatenations of features across the two layers and demonstrated superior results compared to the categorization based upon one layer only.

This paper proposes a new concept in hierarchical representations that exploits features of different granularity and specificity coming from all layers of the hierarchy. The approach addresses the problem of statistically terminal shape nodes that emerge within hierarchical layers during learning, as well as shows how the pooling of layer-specific features extracted on objects can be performed in order to pass the appropriate granularities of shape to the higher-level, category-specific representation. The concept is realized within a cross-layered compositional representation learned from the visual data. We show how similarity connections among discrete states/labels within and across hierarchical layers can be established in order to produce a set of layer-independent shape-terminals, *i.e. shapinals*. We thus break the traditional notion of hierarchies and show how the category-specific layers can make use of *all* the necessary features stemming from *all* hierarchical layers. This, on the one hand, brings higher generalization into the representation, yet on the other hand, it also encodes the notion of scales directly into the hierarchy, thus enabling a multi-scale representation of object categories. By focusing on shape information only, the approach has been tested on the standard datasets demonstrating good performance in comparison with the other state-of-the-art methods.

The paper is organized as follows: in Section 3 we first give a brief overview of the proposed method, summarize the hierarchical compositional representation of [7], which serves as the basis to our model in Subsec. 3.1, propose means to make similarity connection between hierarchical nodes within (Subsec. 3.2) and across (Subsec. 3.3) layers, and finally show how the layer-independent representation with *shapinals* can be obtained (Subsec. 3.5). Experimental results are presented in Section 4, with the summary and conclusions given in Section 5.

## 2. Contributions

This paper addresses three major problems that arise in hierarchical systems:

- **The problem of which hierarchical features extracted on objects should be forwarded to higher-level, category-specific representations.** We propose to extract the so-called *shapinals* defined as shape-termination features detected across hierarchical layers. The shapinals inhibit all the smaller, densely appearing features already coded within them, thus providing a compact description of objects.

- **The problem of scale normalization of objects and their respective features.** We show how scales can be encoded *directly* into the hierarchical layers by establishing similarity connections between features appearing across various levels of the hierarchy and show how the conjoint ratio of scales of object-specific features maps into the layer-relative firings of hierarchical nodes.

- **The problem of generalization.** Since hierarchies usually code information in discrete nodes, two nodes that respond to perceptually similar visual features are likely to be realized in different hierarchical labels. In order to provide robustness, generalizations and thus repeatability for intra-class variations of features, we propose means to enable geometrical comparisons of hierarchical nodes within layers.

Within the proposed framework we additionally show how the issue of statistically terminal nodes (nodes that do not combine into higher-layer features) is solved naturally, and how the concept of cross-layered representations is able to avoid the combinatorial explosion of feature combination and enables unsupervised learning of higher hierarchical layers.

## 3. Cross-layered similarity connections for a layer-independent representation

In images, objects may appear in various sizes and contain features of different specificity and granularity. By building hierarchical representations based on image statistics [7], it is thus to expect that such features will emerge across different layers of the hierarchy. Since the receptive field sizes increase with the level of hierarchy, small and simple feature detectors usually define the lower layers, while finer and more complex edge structures are coded with the higher hierarchical layers. Consequently, larger objects in images will produce higher-level dynamic bindings, while the detection of features on smaller objects will terminate in the lower layers. Not to lose any descriptive information pertaining to various objects, it is thus crucial to carefully pool features detected across multiple layers of the hierarchy (the schematic overview of the proposed concept is presented in Fig. 1).

Additionally, the hierarchical nodes are usually realized as discrete states/labels thus having generalization problems. As, for example, no two horses have the same shape of backs or mouths, it is hard to expect that the same hierarchical node would fire in both cases (depicted in Fig. 2).

We thus need the means of finding the similarities among different hierarchical nodes in a geometrical sense.

We propose to create similarity connections between hierarchical nodes *within layers* to achieve invariance for high variability in object shape and draw similarities *across layers* to achieve a proper scale normalization of features. We show how a layer-independent description of objects defined by the so-called shape-terminals, *i.e. shapinals*, can be passed to the higher-level, the category-specific representation. If performed in this manner, the problem of terminal nodes within the hierarchical "library" is solved in a natural way. There is no need to by-pass or float features to the top-most layer and thus unnecessarily load the complexity of representation, which may prevent the unsupervised creation of higher layers (the problem arising in [7]). Instead, at each hierarchical stage of learning, only a subset of the layer's statistically most repeatable features can be combined further, yet the final, cross-layered description of objects will retain its descriptive power.

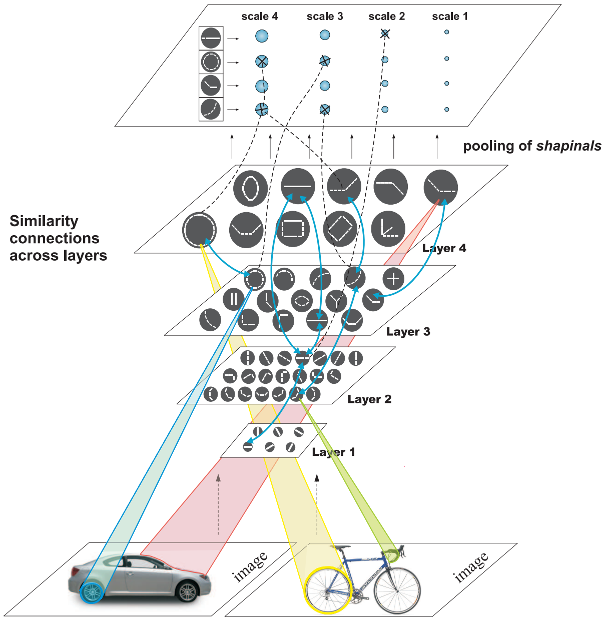

Figure 1. Cross-layered, scale independent representation.

## 3.1. The base model: hierarchical compositional framework [7]

We build on our previously proposed approach [7], where we proposed an unsupervised learning framework to obtain a hierarchical compositional representation of object categories. Starting with simple oriented filters the approach learns the first three layers of optimally sharable features, defined as loose spatial compositions, *i.e. parts*. Upon the third layer, a higher-layer categorical representation is derived with minimal supervision. The model is in essence composed of two recursively iterated steps, 1.) a layer-learning process that statistically extracts parts by sequentially increasing the number of subparts contained in local image neighborhoods, and 2.) a part detection step that finds the learned compositions in images with an efficient and robust *indexing and matching* scheme. The advantage of the proposed representation lies in the capability to model exponential variability present in images, yet still retaining the computational efficiency by keeping the number of indexing links per each part approximately constant across layers.

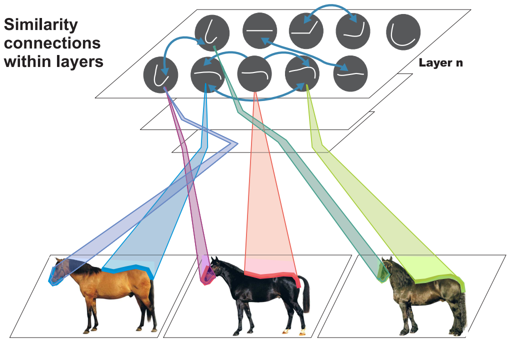

Figure 2. For greater generalization we establish similarities between hierarchical nodes within layers.

We adopt the terminology and the notation from [7]. The rotation invariance is dropped from our implementation, however, since we believe it can be incorporated into the model along with other invariances at a later stage of learning.

Let $\mathcal{L}_n$ denote the $n$-th Layer. Each element of $\mathcal{L}_n$, i.e. part, is envisioned to model spatial relations between its subparts, which furthermore model the spatial relations between their constituent subcomponents, etc. Parts are thus defined recursively in the following way. Each part $\mathcal{P}_i^n$ in $\mathcal{L}_n$ is characterized by the center of mass, and a list of subparts (parts of the previous layer) with their respective positions relative to the center of $\mathcal{P}_i^n$. One subpart is the so-called *central part* that indexes into $\mathcal{P}_i^n$ from the lower, $(n-1)$th layer. Specifically, a $\mathcal{P}_i^n$ that is centered to $(0,0)$ encompasses a list $\{(\mathcal{P}_j^{n-1}, (x_j, y_j), (\boldsymbol{\sigma}_{1j}, \boldsymbol{\sigma}_{2j}))\}_j$, where $(x_j, y_j)$ denotes the relative position of subpart $\mathcal{P}_j^{n-1}$, while $\boldsymbol{\sigma}_{1j}$ and $\boldsymbol{\sigma}_{2j}$ denote the principal axes of an elliptical Gaussian encoding the variance of its position around $(x_j, y_j)$. The hierarchy starts with a fixed $\mathcal{L}_1$ composed of a set of oriented Gabor filters, $\{filter_i\}$.

Along the lines of how the parts are formed [7], the relative positions with variances $\{(x_j, y_j), (\boldsymbol{\sigma}_{1j}, \boldsymbol{\sigma}_{2j})\}$ may be preferably replaced with the segmented *spatial maps*, $\{map_j\}_j$, which capture the variability of subparts more accurately than the fitted Gaussians. Spatial map is a two-dimensional map that contains the learned disposition of locations of each subpart relative to the central part, upon which the parameters of the approximately Gaussian distribution are estimated [7]. We will use both terms, interchangeably.

### 3.2. Similarity between parts within layers

The parts defined as hierarchical compositions can be learned without supervision from a set of images. The drawback, however, is the fact that they are realized as discrete labels (part types) without a proper geometrical parametrization that would enable a comparison between them. Consequently, two visually similar curvatures are likely to be encoded in two different hierarchical nodes. Grouping by co-occurrence [7] only partially solves this problem: since perceptibly equal structures can be composed in different ways, parts formed as different compositions will highly co-occur. However, two visual shapes that are only similar to a certain extent are likely to have a small, random co-occurrence. It is thus crucial to have the means of comparing two different parts in a geometrical sense.

In a strictly mathematical sense, the optimal similarity between two hierarchically formed shapes encoding statistically learned spatial variations, would be to draw samples from the distribution of spatial variance for each subpart, with the process repeated down to the original Layer 1, which contains a fixed set of shape points. Each sampled part would thus take the form of a number of shape-forming points. Here-on, two parts could be compared using standard shape similarity techniques. The final similarity value would be the average similarity calculated over a number of sampled versions of two compared parts. However, due to a larger number of parts (in the order of a thousand) in the higher hierarchical layers, this method would be computationally very demanding.

Since each part is formed as a recursive spatially loose composition, a comparison should be performed in a similar manner. We consider two parts to be perceptually similar if both have a similar spatial configuration of subparts. However, the compatibility of spatial variations of subparts within two parts that come directly from the adjacent lower layer should contribute more to the similarity function than the spatial variations encoded in their subparts which code much smaller receptive fields.

We thus define a *similarity* measure $\mathrm{sim}_{n,k}(\mathcal{P}_i^k, \mathcal{P}_j^k)$ between parts $\mathcal{P}_i^k$ and $\mathcal{P}_j^k$, both from Layer $k$, with respect to Layer $n$ recursively as follows.

$$\mathrm{sim}_{n,k}(\mathcal{P}_i^k, \mathcal{P}_j^k) = \min\bigl\{\mathrm{sim}'_{n,k}(\mathcal{P}_i^k, \mathcal{P}_j^k), \mathrm{sim}'_{n,k}(\mathcal{P}_j^k, \mathcal{P}_i^k)\bigr\}$$

for $1 < k \leq n$ and

$$\mathrm{sim}_{n,1}(\mathcal{P}_i^1, \mathcal{P}_j^1) = \rho\bigl(\mathcal{R}_{f_{n,1}}(\mathit{filter}_i), \mathcal{R}_{f_{n,1}}(\mathit{filter}_j)\bigr)$$

otherwise, where

$$\mathrm{sim}'_{n,k}(\mathcal{P}_i^k, \mathcal{P}_j^k) = M\bigl(\mathrm{psim}_{n,k}(\mathcal{P}_{i_1}^{k-1}, \mathcal{P}_j^k), \ldots, \mathrm{psim}_{n,k}(\mathcal{P}_{i_m}^{k-1}, \mathcal{P}_j^k)\bigr),$$

and

$$\mathrm{psim}_{n,k}(\mathcal{P}_{i_t}^{k-1}, \mathcal{P}_j^k) = \max_{j_l}\bigl\{\mathrm{sim}_{n,k-1}(\mathcal{P}_{i_t}^{k-1}, \mathcal{P}_{j_l}^{k-1}) \cdot \rho\bigl(\mathcal{R}_{f_{n,k}}(\mathit{map}_{i_t}), \mathcal{R}_{f_{n,k}}(\mathit{map}_{j_l})\bigr)\bigr\}.$$

Here $\rho(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$ is a measure for the similarity between maps $A$ and $B$ ($A \cdot B$ denotes a component-wise inner product of two matrices and $\|A\|$ is a norm induced by this product). Note that $\rho(A, A) = 1$ and $\rho(A, B) = 0$ when the supports of $A$ and $B$ are disjoint.

Further, $\mathcal{R}_{f_{n,k}}(\cdot)$ denotes resampling by a factor $f_{n,k}$, where $f_{n,k}$ refers to the quotient of the receptive field sizes of layers $k$ and $n$, respectively. Since the receptive field sizes of subparts relative to part from Layer $n$ reduce by factor 2, we take $f_{n,k} = 0.5^{n-k}$. By resampling the spatial maps of lower layers, we are weighting down the influences that the lower layer subparts have on the final similarity calculation. Thus, from the perspective of Layer 4 for example, the different orientations of the filters that compose a certain part down at the base, Layer 1 level, become more alike and virtually an unimportant factor in the similarity calculation.

Next, we observe that $\mathrm{psim}_{n,k}(\mathcal{P}_{i_t}^{k-1}, \mathcal{P}_j^k)$ gives a similarity between $\mathcal{P}_{i_t}^{k-1}$, a subpart of $\mathcal{P}_i^k$, and the best matching subpart of $\mathcal{P}_j^k$. For $\mathrm{sim}'_{n,k}(\mathcal{P}_i^k, \mathcal{P}_j^k)$ we would like to give an *average* similarity between the subparts of $\mathcal{P}_i^k$ and their best matched subparts of $\mathcal{P}_j^k$. To do this, we take:

$$M_{\text{avg}}(x_1, x_2, \ldots, x_m) = \frac{1}{m}\sum_{i=1}^{m} x_i,$$

$$M(x_1, x_2, \ldots, x_m) = \begin{cases} 0, & \text{if there exists } j \text{ s.t. } x_j < T, \\ M_{\text{avg}}(x_1, x_2, \ldots, x_m), & \text{otherwise} \end{cases}$$

In the similarity function $M$ as defined above, the similarity between two parts, where some subpart cannot be matched to any subpart of the second part within a specified tolerance $T$, is set to zero. Finally, the similarity between two parts $P_i^n$ and $P_j^n$, both from Layer $\mathcal{L}_n$, is obtained as $\mathrm{sim}_{n,n}(P_i^n, P_j^n)$.

The advantage of calculating the similarity in this way lies in its recursive formulation. When calculating similarities within Layer $n$, we can efficiently re-use the similarities calculated from the layers below. Since the majority of similarities are small on Layer $n-1$, the number of Layer $n$ parts that need to be compared also becomes very low.

We must emphasize, however, that all the similarity calculations are performed during an offline-stage and thereafter become a part of the hierarchical library (Subsection

3.4). This information can then be used efficiently in online processing of images.

## 3.3. Similarity between parts across layers

Comparing parts from different layers is somewhat harder, since their receptive fields are at least by a factor 2 apart. We propose to calculate the similarities in the following way. Let a similarity measure between the parts $\mathcal{P}_i^k$ and $\mathcal{P}_j^{k'}$, on Layers $k$ and $k'$ with respect to Layer $n$, be

$$\begin{aligned} &\mathrm{sim}_{n,k,k'}(\mathcal{P}_i^k, \mathcal{P}_j^{k'}) = \\ &M\big(\mathrm{psim}_{n,k,k'}(\mathcal{P}_{i_1}^{k-1}, \mathcal{P}_j^{k'}), \ldots, \mathrm{psim}_{n,k,k'}(\mathcal{P}_{i_m}^{k-1}, \mathcal{P}_j^{k'})\big) \end{aligned} \tag{1}$$

for $1 < k' < k \leq n$, and

$$\mathrm{sim}_{n,k,1}(\mathcal{P}_i^k, \mathcal{P}_j^1) = \rho\big(\mathcal{CR}(\mathcal{P}_i^k), \mathcal{R}_{f_{1,k}}(\mathit{filter}_j)\big)\,, \tag{2}$$

for $1 < k \leq n$, where $\mathcal{CR}(\mathcal{P}_i^k)$ denotes the reconstruction of the part $\mathcal{P}_i^k$ with filters on Layer 1 and

$$\begin{aligned} &\mathrm{psim}_{n,k,k'}(\mathcal{P}_{i_t}^{k-1}, \mathcal{P}_j^{k'}) = \\ &\qquad \max_{j_l}\big\{\mathrm{sim}_{n,k-1,k'-1}(\mathcal{P}_{i_t}^{k-1}, \mathcal{P}_{j_l}^{k'-1}) \cdot \\ &\qquad\qquad \rho\big(\mathcal{R}_{f_{n,k}}(map_{i_t}), \mathcal{R}_{f_{n,k'}}(map_{j_l})\big)\big\}\,. \end{aligned}$$

In (1) we recursively compare each subpart of $\mathcal{P}_i^k$ with subparts of $\mathcal{P}_j^{k'}$ and taking the *average* given by a suitable function $M$ (see previous section). The second case (2) covers the calculation of similarities when we compare a part $\mathcal{P}_i^k$ on Layer $k$ to a part $\mathcal{P}_j^1$ on Layer 1. Here we choose to reconstruct the part $\mathcal{P}_i^k$ from the filters on Layer 1 (by recursively using relative centers of subparts, $(x_i, y_i)$), resize the filter corresponding to $\mathcal{P}_j^1$, and calculate the correlation between the two maps.

Note that this measure, in contrast to the measure within layers defined in the previous section, is not symmetric. To make it symmetric does not take much effort, but would further complicate the definition.

## 3.4. Creating layer-independent labels

For each part $\mathcal{P}^n$ we can now make connections to all the parts in the hierarchy that have a similarity above a chosen threshold (we take it to be $0.5$). This can be seen as organizing the parts topologically; the connections and their strength define a similarity neighborhood of each part. Such organization of parts offers numerous advantages and brings higher robustness into the representation: whenever a certain part is sought in the detection stage, any part from its defined similarity neighborhood may contribute to the final hypothesis.

Next, the layer-independent set of labels is created with a simple greedy algorithm. Firstly, each part in the hierarchy is assigned into a separate group. At each step, two groups are joined (are assigned the same label), if their similarities are higher than a chosen threshold. The similarity between the joined group and the rest of the groups is set as the minima of similarities of parts forming the group with the parts forming the remaining groups. When no two groups of parts are above the threshold the process ends. Groups are labelled and a mapping from the parts in the hierarchical library into the obtained layer-independent set of labels is formed: $\mathit{Cross\_layer}(\mathcal{P}_i^k) = l$, where $l$ denotes the label of the group to which a part $\mathcal{P}_i^k$ belongs.

The layer-independent labels thus equalize parts coding lines, circles, etc. across layers of the hierarchical library.

## 3.5. *Shapinals*: Obtaining a cross-layered object description

We first summarize the part detection process adopted from [7] and then show how the final set of *shapinals* is obtained from the set of features detected across layers.

For any given image, the part detection process starts by describing the image in terms of small oriented edges (*i.e.* Gabor filters). This is done on only a small set of scales – each rescaled version of the original image is processed separately. By extracting local maxima of the Gabor energy function that are above a low threshold, an image is transformed into a list of $\mathcal{L}_1$ parts; $\{\pi_i^1\}_i$, where $\pi_i^n$ stands for a *realization* of the $\mathcal{L}_n$ part $\mathcal{P}^n$ with a corresponding location at which it was recovered in an image; $\pi_i^n = \{\mathcal{P}^n, x_i, y_i\}$, where $i$ denotes the successive number of the found part. At each hierarchical step a set of links $\Lambda_n$ is additionally defined, where $\Lambda_n(\pi_i^n)$ represents a list of all image location points that contributed to part $\pi_i^n$. $\Lambda_n$ is calculated from $\Lambda_{n-1}$ at each step up in the hierarchy, while $\Lambda_1$ is simply a list of all image pixels on which a particular Gabor filter fired. Due to compositional nature of parts, each $\pi_i^n$ binds together several parts from the adjacent, lower layer, *i.e.* $\{\pi_j^{n-1}\}_j$, thus the set of links for $\pi_i^n$ is computed as

$$\Lambda_n(\pi^n) = \bigcup_j \Lambda_{n-1}(\pi_j^{n-1})$$

Through the set of links, all the found parts from all the hierarchical layers "meet" at the pixel, image-level. Each higher level interpretation is then found by iterating the indexing and matching step [7].

From the complete list of parts detected across all layers and a small number of scales, we extract a set of *shapinals* as follows. At each step, we select a part $\pi_i$ with the highest cardinality of $\Lambda(\pi_i)$, corresponding to the number of image points it describes. By performing local inhibition (described in Alg.1), all the smaller parts that are either already bound within the selected part or do not code any additional pixel-level information, are discarded from the pending list of parts. When adding the selected part $\pi_i = \{\mathcal{P}, x, y\}$ to the list of *shapinals*, we assign it a layer-independent label, $Cross\_layer(\mathcal{P})$ and a value of $lod$, *i.e. level-of-detail*. This value codes the approximate size of the part taking into account the scale and level of hierarchy at which it was detected. Since the receptive fields of hierarchical layers increase by a certain factor (usually 2), the notion of size is thus also encoded within the hierarchy and can naturally be combined with image scales. By sampling 2 scales per octave, level-of-detail can be calculated as $lod = round(i_{layer} + 0.5 \cdot scale)$. To give an example, a line detected at layer 2 and scale 3 will be given the same value of $lod$ as a line detected at layer 3 and scale 1. The algorithm is summarized in Alg. 1.

The use of *shapinals* for higher, category-specific representation, offers numerous advantages. Firstly, smaller as well as larger structures are encoded within *shapinals*, thus virtually no information is lost through hierarchical processing. Secondly, the representation is scale-invariant: the labels with corresponding relative positions will stay approximately the same for two objects on different scales, only the $lod$ value will differ. However, the difference of $lod$-s between the two objects will stay roughly constant, thus enabling a robust voting scheme for the scale of the object.

Since a higher-level, category-specific representation is outside the scope of this paper, we only briefly describe how the extracted *shapinals* can be used with standard classifiers. As an input, we form a histogram of different types of *shapinals* at various $lod$ values. The histogram is obtained by summing the responses for each part type appearing in each quadrant of an image, thus each part type produces four entries in the histogram. The first dimensions of the feature vector are reserved for the *shapinals* with the highest $lod$, followed by *shapinals* at $lod - 1$, etc. In our experiments we used three stacked histograms of *shapinals* corresponding to three $lod$s. This way, the relative scales of features are encoded into the final feature vector. The histograms were additionally normalized as proposed in [13].

## 4. Experimental results

To test the proposed framework, the hierarchical library was first created by employing our unsupervised learning approach [7] on a set of 1500 natural images. The learned compositional hierarchy consisted of 160 parts on Layer 2 and 553 Layer 3 parts (a few examples from both layers are depicted in Fig. 3). The complete learning process took approximately 5 hours on one core of an Intel Core-2 CPU 2.4 Ghz computer. It is evident from Fig. 3 that a number of perceptually similar parts emerge across and within layers, giving rise to our proposed similarity-based cross-layered representation.

**Algorithm 1** : Pooling of *shapinals* in images

```
INPUT: A list of parts found at each layer and scale:
$\Pi_{all} = \{\pi^{i_{layer}}_{i,scale}, \Lambda_{i_{layer}}(\pi_i)\}_{i,scale,i_{layer}}$
$\Pi_{shapinals} = \emptyset$
sort $\Pi_{all}$ by decreasing value of $|\Lambda_{i_{layer}}|$ (corresponding to the number of described image points)
while $\Pi_{all} \neq \emptyset$ do
    add the part that describes most image points to the shapinal list with its corresponding layer-independent label, $\pi_{i,i_{layer},scale} \in \Pi_{all}(1)$ and the value of level-of-detail (see text for explanation):
    $\Pi_{shapinals} := \Pi_{shapinals} \bigcup \{Cross\_layer(\pi_i), lod_i\}$,
    where $lod_i = round(i_{layer} + 0.5 \cdot scale)$

Perform local inhibition with the selected part:
    find all parts $\pi_j \in \Pi_{all}$ that have
    $1 - \frac{|\Lambda_j(\pi_j) \bigcap \Lambda_j(\pi_i)|}{\max\{|\Lambda_j(\pi_i)|,|\Lambda_j(\pi_j)|\}} < thresh$
    (we take $thresh = 0.3$)
    remove $\{\pi_j\}$ from $\Pi_{all}$: $\Pi_{all} = \Pi_{all} \setminus \{\pi_j\}$
end while
return A list of shapinals, $\Pi_{shapinals}$
```

The calculated non-zero similarities for one third-layer part between other parts in the hierarchical library are depicted in Fig.4 (2). Upon these similarities, the parts were grouped to produce 102 layer-independent labels. An example of the extracted *shapinals* based on these labels is shown in Fig.4 (4).

To put the proposed hierarchical concept in relation to other hierarchical approaches as well as other categorization methods, which focus primarily on shape information, the approach was tested on the Caltech 101 database [5]. The Caltech 101 dataset contains images of 101 different object categories with the additional background category. The number of images varies from 31 to 800 per category, with the average image size of roughly 300 × 300 pixels.

Each image was processed on 3 different scales, spaced apart by $\sqrt{2}$. The average processing times per image per layer obtained with our C++ implementation are reported in Table 1. Most of the processing time is spent filtering an image with 6 Gabor filters ($\mathcal{L}_1$), which has not been optimized for performance. To demonstrate the utility of our approach, we ran several classification trials with different levels of the hierarchy as well as the final, *shapinal* representation. The features were combined with a linear SVM for multiclass classification. For this experiment we used 15 images for training and 15 images for testing, disjunct from the training set (this experimental methodology was taken after [22]). The results, averaged over 8 random splits, are reported in Table 2 with classification rates of other hierarchical approaches shown for comparison.

Classification with *shapinals* was also tested by varying the number of training examples. For testing, 50 examples were used for categories where this was possible and less otherwise. The classification rate was normalized accordingly. In all cases, the result was averaged over 8 random splits. The results are presented and compared with other categorization methods in Fig. 6. We must emphasize that the proposed model focuses on *shape* information *only*. To make an even clearer case, the texture information was discarded from classification altogether. This was done by inhibiting all the parts found in an image at each layer that had “too many” subparts in their local neighborhoods. An example of detected texture is depicted in Fig. 4 (3). We thus compare our method to only those categorization approaches that do not combine several other modalities. It is worth noting, however, that most of the shown methods also rely on processing texture as well.

The proposed hierarchical concept offers another advantage over previous hierarchical frameworks. By pooling information from multiple layers of the hierarchy, the descriptive power remains virtually intact with an increasing number of layers. We can thus afford to learn additional layers by using just a set of the most repeatable parts from the last layer (Fig 4(1) shows the learned four-level hierarchy). The benefit is two-fold: firstly, information coded in higher layers can be extracted from images more robustly than simply using lower layers with higher image scales; secondly, the reliability of extracting shape-terminations on larger objects increases with level of hierarchy, which we believe will be beneficial when forming more elaborate category-specific representations than the standard SVM. This is part of our on-going research.

Fig. 5 (left) shows a few examples of part detections using the similarity connections described in Subsec. 3.2, while Fig. 5 (right) depicts an example of part detections using the across-layer similarity connections as described in Subsec 3.3. It can be seen how finer and larger structures can reliably be detected with higher hierarchical layers.

Table 1. Average processing time for different steps per image

| | Processing time per image |
|---|---:|
| Layer 1 | 1.6 s |
| Layer 2 | 0.54 s |
| Layer 3 | 0.66 s |
| *Shapinals* | 0.22 s |

## 5. Summary and conclusions

This paper proposed a new concept in hierarchical representations that exploits features of different granularity and specificity coming from *all* layers of the hierarchy. The concept was realized within a cross-layered compositional representation learned from the visual data. We showed how similarity connections among discrete labels within and across hierarchical layers can be established in order to produce a set of layer-independent shape-terminals, i.e. *shapinals*.

Table 2. Average classification rate on Caltech 101

| | $N_{train} = 15$ | $N_{train} = 30$ |
|---|---:|---:|
| Layer 2 only | 55 | / |
| Layer 3 only | 52.9 | / |
| **shapinals** | **60.5** | **66.5** |
| Serre et al. [18] | 44 | / |
| Mutch et al. [13] | 51 | 56 |
| Ranzato et al. [16] | / | 54 |
| Ommer et al. [14] | / | 61.3 |
| Wolf et al. [21] | 51.18 | / |

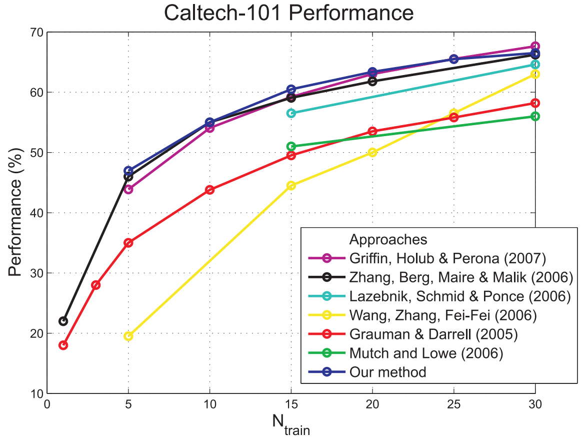

Figure 6. Caltech 101 classification results for methods focusing primarily on shape.

The results confirm the utility of the proposed approach in the classification task. However, we believe the main advantage of the presented hierarchical concept will show when devising more sophisticated higher-level, category-specific representations. This is part of our on-going work.

## References

[1] A. Agarwal and B. Triggs. Hyperfeatures - multilevel local coding for visual recognition. In *ECCV (1)*, pages 30–43, 2006.

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

[3] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *ICCV ’07*, 2007.

[4] D. J. Crandall and D. P. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV (1)*, pages 16–29, 2006.

[5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE. CVPR’04, Workshop on Generative-Model Based Vision*, 2004.

[6] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR(2)*, pages 264–271, 2003.

[7] S. Fidler and A. Leonardis. Towards scalable representations of visual categories: Learning a hierarchy of parts. In *CVPR’07*.

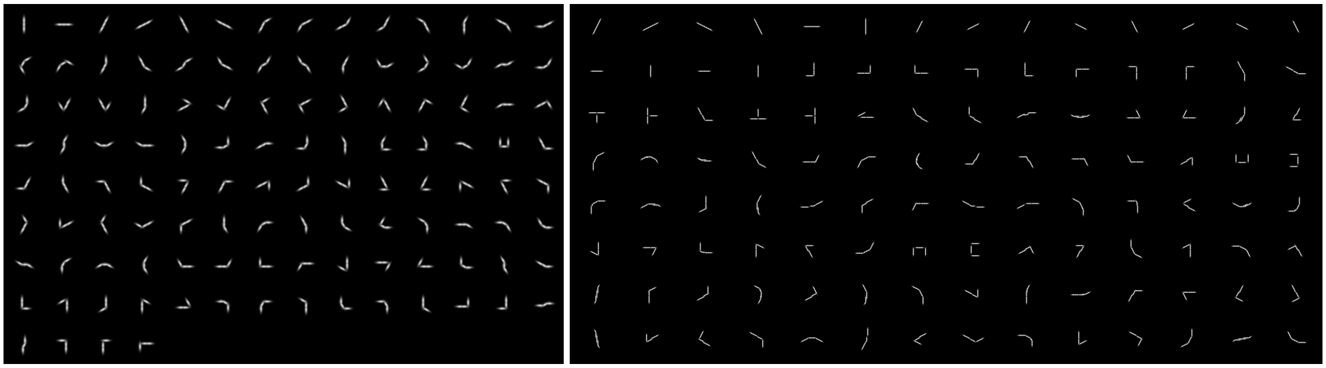

Figure 3. $\mathcal{L}_2$ and $\mathcal{L}_3$ parts (only a subset is shown) used in the Caltech 101 experiments.

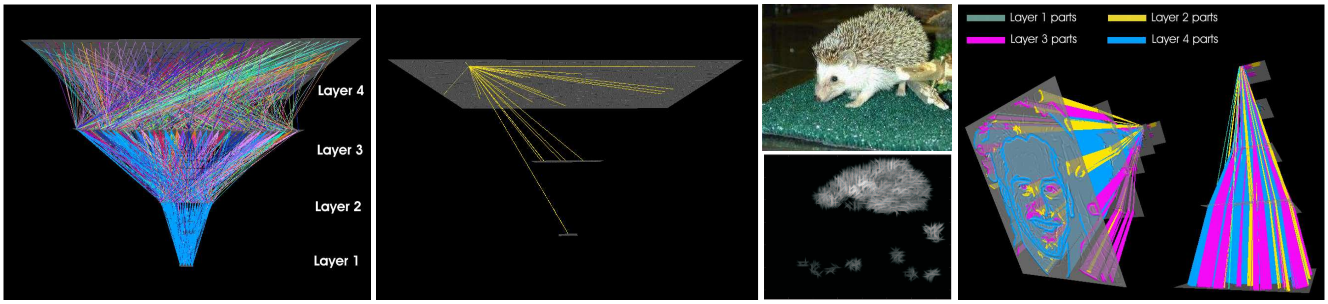

Figure 4. From left to right: 1.) The complete four-layer hierarchy with compositional links depicted. 2.) Similarity connections for one third layer part for the hierarchy from Fig. 3. 3.) Texture found and discarded from the categorization process. 4.) Extracted shapinals.

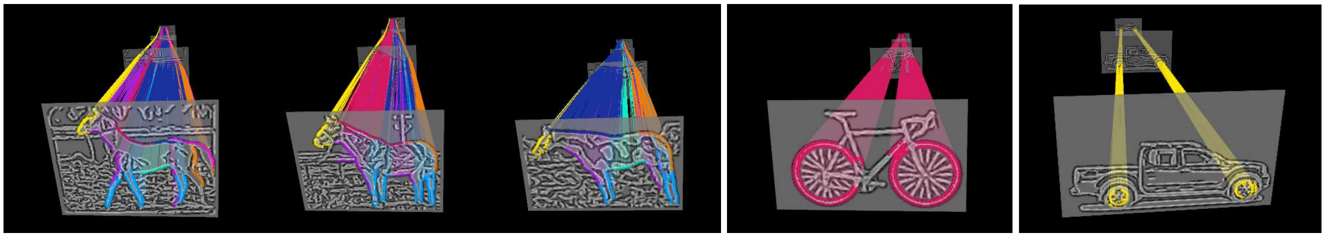

Figure 5. Left: Repeatability of parts by using calculated similarity. Right: 'Circle' part detections across different layers.

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

[9] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV'05*, pages 1458–1465.

[10] F.-J. Huang and Y. LeCun. Large-scale learning with svm and convolutional nets for generic object categorization. In *CVPR*, pages 284–291, 2006.

[11] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In *CVPR (2)*, pages 2145–2152, 2006.

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

[13] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR06*, pages 11–18, 2006.

[14] B. Ommer and J. M. Buhmann. Learning the compositional nature of visual objects. In *CVPR'07*, 2007.

[15] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV (2)*, pages 575–588, 2006.

[16] M. Ranzato, F.-J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR'07*.

[17] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlatons. In *CVPR (2)*, pages 2033– 2040, 2006.

[18] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Object recognition with cortex-like mechanisms. *PAMI*, 29(3):411–426, 2007.

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

[20] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.

[21] L. Wolf, S. Bileschi, and E. Meyers. Perception strategies in hierarchical vision systems. In *CVPR '06*, pages 2153–2160.

[22] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR (2)*, pages 2126–2136, 2006.