Machine Vision and Applications (2016) 27:1299–1312
DOI 10.1007/s00138-016-0763-9

SPECIAL ISSUE PAPER

Can computer vision problems benefit from structured hierarchical classification?

Thomas Hoyoux[1] · Antonio J. Rodríguez-Sánchez[2] · Justus H. Piater[2]

Received: 8 December 2015 / Revised: 15 March 2016 / Accepted: 16 March 2016 / Published online: 6 May 2016

Abstract Research in the field of supervised classification has mostly focused on the standard, so-called "flat" classification approach, where the problem classes live in a trivial, one-level semantic space. There is however an increasing interest in the hierarchical classification approach, where a performance gain is expected by incorporating prior taxonomic knowledge about the classes into the learning process. Intuitively, the hierarchical approach should be beneficial in general for the classification of visual content, as suggested by the fact that humans seem to organize objects into hierarchies based on visually perceived similarities. In this paper, we provide an analysis that aims to determine the conditions under which the hierarchical approach can consistently give better performances than the flat approach for the classification of visual content. In particular, we (1) show how hierarchical methods can fail to outperform flat methods when applied to real vision-based classification problems, and (2) investigate the underlying reasons for the lack of improvement, by applying the same methods to synthetic datasets in a simulation. Our conclusion is that the use of high-level hierarchical feature representations is crucial for obtaining a performance gain with the hierarchical approach, and that poorly chosen prior taxonomies hinder this gain even though proper high-level features are used.

Keywords Hierarchical classification · Flat classification · Structured K-nearest neighbors · Structured support vector machines · Maximum margin regression · 3D shape classification · Expression recognition · Simulation framework · Feature representations

1 Introduction

Most of the theoretical work and applications in the field of supervised classification have been dedicated to the standard classification approach, where the problem classes are considered to be equally different from each other in a semantic sense [28]. In this standard approach, also known as "flat" classification, a classifier is learned from class-labeled data instances without any explicit information given about the high-level semantic relationships between the classes. A standard multiclass problem formulation will for example consider a bee, an ant and a hammer to be different to the same degree; they belong to different classes in a flat sense because the only available semantic information comes from the same unique semantic level. However, one could consider that ants and bees are part of a superclass of insects, while hammers belong to another superclass of tools, and it is intuitive that such hierarchical knowledge about the classes can help improve the classification performances. Based upon this realization, a new approach has emerged for dealing more efficiently with classification of content deemed to be inherently semantically hierarchical, i.e., the hierarchical classification approach [28]. The attention given to the hierarchical approach was also sustained by the advances made in

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under Grant Agreements No. 270273, Xperience, and No. 600918, PaCMan.

✉ Thomas Hoyoux
thomas.hoyoux@ulg.ac.be

✉ Antonio J. Rodríguez-Sánchez
antonio.rodriguezsanchez@uibk.ac.at

[1] Signal and Image Exploitation (INTELSIG), Montefiore Institute, University of Liège, Liège, Belgium

[2] Intelligent and Interactive Systems, Institute of Computer Science, University of Innsbruck, Innsbruck, Austria

machine learning generalized to arbitrary output spaces, i.e., the structured classification approach (e.g., [31]), of which the hierarchical approach is actually a special case.

The a priori hierarchical organization of classes has been shown to constitute a key prior to classification problems in several application domains, including text categorization [23], protein function prediction [7], and music genre classification [14]. As for classification based on visual features, a hierarchical prior intuitively seems especially appropriate as it reflects the natural way in which humans organize and recognize the objects they see, which is also supported by neurophysiological studies of the visual cortex [2,15,35]. In practice, some results have shown that there is indeed a gain in performance with the hierarchical approach in the visual-based application domain, e.g., for 3D object shape classification [3] and annotation of medical images [6]. A quite active and closely related line of work consists of the *supervised construction* of class hierarchies from images with multiple tag labels. The motivation is to reduce the complexity of visual recognition problems that have a very large number of instances. To build useful taxonomies, the proposed methods exploit either purely the semantic tag labels [19,29], or purely the visual information [10,18], or both as in [16], where the authors propose a way to learn a "semantivisual" hierarchy that is both semantically meaningful and close to the visual content.

In this paper, we are interested in determining the conditions under which the hierarchical approach can consistently give better performances than the flat approach for the classification of visual content. This paper is an extended version of the work published in [11], where we applied three hierarchical classification methods and their flat counterparts to two inherently hierarchical vision-based classification problems: facial expression recognition and 3D shape classification. Using evaluation measures designed for hierarchical classification, we showed in [11] that, for the considered methods and problems, the hierarchical approach provided no or only marginal improvement over the standard approach. We here extend our previous work by designing a simulation framework and conducting the comparative evaluation of the hierarchical and flat methods used in [11] this time applied to artificial problems generated with this simulation framework. Specifically, we generate completely synthetic datasets for which we can control the complexity through the manipulation of key aspects, such as the underlying hierarchical phenomenon at the origin of the data measurements, the amount of noise in the extraction of the features from the measurements, and the amount of knowledge about the underlying hierarchical phenomenon. Our goal with these simulation experiments is to draw useful insights to explain why the hierarchical approach did not outperform the flat approach when applied to our real vision-based classification problems.

The remainder of this paper is organized as follows. Section 2 describes the hierarchical framework and terminology we adopted for our previous and present work, and provides the details of the hierarchical methods used. Section 3 shows the experimental evaluation first presented in [11], where the hierarchical and flat methods were applied to real computer vision problems. Section 4 presents our simulation framework, as well as the experimental results obtained for artificial problems generated with this simulation framework. In light of the additional simulation results, we provide a discussion in Sect. 5 and draw a conclusion in Sect. 6.

2 Methods for hierarchical classification

2.1 Framework and terminology

Recently, a necessary effort to unify the hierarchical classification framework has been made [28]. We follow on their terminology which is summarized next. A class taxonomy consists of a finite set of semantic concepts $\mathcal{C} = \{c_i \mid i = 1 \ldots n\}$ with a partial order relationship $\prec$ organizing these concepts either in a tree or a directed acyclic graph (DAG). A classification problem defined over such a taxonomy is hierarchical: its classes and superclasses correspond to the leaf and interior nodes of the tree (or DAG), respectively. A flat classification problem only considers the leaf nodes of such a taxonomy as its classes and has no superclass. A hierarchical classification problem deals with either single- or multiple-path labeling, i.e., whether or not a single data instance can be labeled with more than one path, and either full or partial depth labeling, i.e., whether or not any path in a label must cover all hierarchy levels. In all cases, an indicator vector representation for the taxonomic label $\mathbf{y}$ of a data instance can be used, i.e., $\mathbf{y} \in \mathcal{Y} \subset \{0, 1\}^n$, where the ith component of $\mathbf{y}$ takes value 1 if the data instance belongs to the (super)class $c_i \in \mathcal{C}$, and 0 otherwise.

The real-world and simulation problems considered in this work are defined using tree taxonomies with full depth labeling. For the facial expression recognition problem, we define multiple path labeling (see Sect. 3.1.1), whereas for the 3D shape classification problem and for our simulation problems we define single path labeling (see Sects. 3.2.1 and 4.2).

Because they do not penalize structural errors, evaluation measures used in the standard flat classification approach may not be appropriate when comparing hierarchical methods to each other, or flat methods to hierarchical methods. In particular, they do not consider that misclassification at different levels of the taxonomy should be treated in different ways. In this work, we adopt the following measures [12], also recommended by [28]: hierarchical precision (hP), hierarchical recall (hR) and hierarchical F-measure (hF), defined as

$$\mathrm{hP} = \frac{\sum_i \left|\hat{P}_i \cap \hat{T}_i\right|}{\sum_i \left|\hat{P}_i\right|}, \quad \mathrm{hR} = \frac{\sum_i \left|\hat{P}_i \cap \hat{T}_i\right|}{\sum_i \left|\hat{T}_i\right|},$$
$$\mathrm{hF} = \frac{2\ \mathrm{hP}\ \mathrm{hR}}{\mathrm{hP} + \mathrm{hR}}, \tag{1}$$

where $\hat{P}_i$ is the set of the most specific class(es) predicted for a test data instance i and all its (their) ancestor classes, and $\hat{T}_i$ is the set of the true most specific class(es) of a test data instance i and all its (their) ancestor classes. These measures are extensions of the standard precision, recall and F-measure, and reduce to them as special cases for flat classification problems.

2.2 Structured hierarchical classifiers

For our first hierarchical classification method, we modify the standard k-nearest neighbors (kNN) method to allow it to cope with structured output that respects a pre-established class taxonomy. We call the resulting classification method Structured output K-nearest neighbors (SkNN). Let $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ be the training set of a hierarchical classification problem. The SkNN classifier is trained in the same way as the standard kNN classifier, i.e., by projecting each training data instance into a feature space using a feature map $\phi(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}$. Given the k nearest neighbors $\mathcal{N} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \mid i \in \{1 \ldots k\}\} \subset \mathcal{D}$ to a test data instance $\mathbf{x} \in \mathcal{X}$, found using a distance metric $\rho(\phi(\mathbf{x}), \phi(\mathbf{x}^{(i)}))$, the classification rule for SkNN is

$$\hat{\mathbf{y}}(\mathbf{x}; \mathcal{N}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \left\langle \sum_{i=1}^{k} w_i \frac{\mathbf{y}^{(i)}}{||\mathbf{y}^{(i)}||}, \frac{\mathbf{y}}{||\mathbf{y}||} \right\rangle, \tag{2}$$

where w_i are weights attributed to the neighbors, which can be chosen to reflect the distances of the neighbors to the test instance, e.g., $w_i = \rho(\phi(\mathbf{x}), \phi(\mathbf{x}^{(i)}))^{-1}$.

Our second hierarchical classification method is the structured output support vector machine (SSVM) [31], which extends the standard support vector machine (SVM) to cope with arbitrary output spaces with non-trivial structure. SSVM defines the relationship between a test data instance $\mathbf{x} \in \mathcal{X}$ and its prediction $\hat{\mathbf{y}} \in \mathcal{Y}$ on the basis of a joint score maximization,

$$\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \langle \mathbf{w},\ \psi(\mathbf{x}, \mathbf{y}) \rangle, \tag{3}$$

where $\mathbf{w}$ is a learned parameter vector and ψ is a user-defined joint feature map $\psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ which projects any pair $(\mathbf{x}, \mathbf{y})$ to its real-valued vectorial representation in a joint feature space. We define the joint feature map for our custom SSVM framework as

$$\psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d, \quad (\mathbf{x}, \mathbf{y}) \mapsto \phi(\mathbf{x}) \otimes \frac{\mathbf{y}}{||\mathbf{y}||}. \tag{4}$$

For our third hierarchical classification method, we use a maximum margin-based regression (MMR) technique (see [1], for example) which is also an extension to the standard SVM, but has several differences with the SSVM method that makes it much faster to train. MMR relies on the fact that the normal vector of the separating hyperplane in SVM can be interpreted as a linear operator mapping the input feature vectors to an output space with general structure. Inference with MMR is performed in the same way as with SSVM (Eq. 3), using the same joint feature map definition (Eq. 4). For each proposed hierarchical method, the inference argmax problem can be solved by exhaustively searching the set $\mathcal{Y}$, which is efficient enough in most applications. In any case, the optimum must belong to the set of valid taxonomic labels, which guarantees that the class taxonomy is respected at all times.

3 Real vision-based classification problems

3.1 Facial expression recognition

3.1.1 The problem

We define an expression using the facial action coding system (FACS) [8] which gives a very detailed description of the human facial movements in terms of action units (AUs). AUs represent atomic facial actions which can be performed independently (though not always spontaneously) by a person. Each AU is associated with the action of a muscle or a group of muscles. The FACS describes more than hundred AUs; a valid code in this system can be for instance $1 + 2 + 5 + 26$, where we have the presence of AU1 (inner eyebrow raiser), AU2 (outer eyebrow raiser), AU5 (upper lid raiser) and AU26 (jaw drop). AUs can be taxonomized according to the region of the face where the action occurs and the type of local deformation the action applies on the face. We therefore propose the tree taxonomy in Fig. 1 for the face expression, inspired by how AUs are usually grouped when presented in the literature [8]. As their names suggest, up-down actions, horizontal actions and oblique actions gather AUs for which the deformation movement in the frontal face is mostly vertical (e.g., AU26: jaw drop), horizontal (e.g., AU20: lip stretcher) or oblique (e.g., AU12 lip corner puller), respectively. Orbital actions group AUs for which the deformation seems to be radial with respect to a fixed point (e.g., AU24: lip pressor, which closes the mouth and puckers the lips, seemingly bringing them closer to the centroid point of the mouth region).

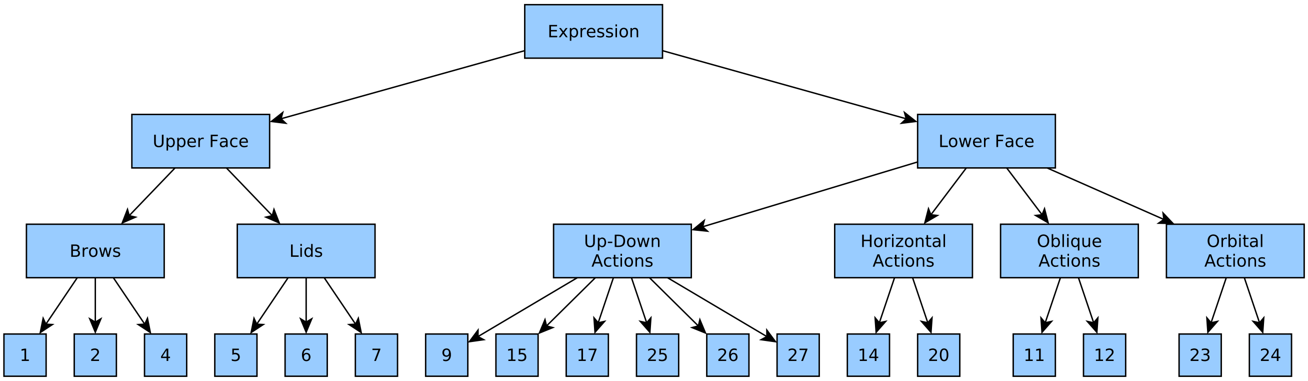

Fig. 1 Our facial expression taxonomy. The leaves (classes) correspond to Action Units

3.1.2 The extended Cohn–Kanade dataset (CK+)

The CK+ dataset [17] consists of 123 subjects between the age of 18 and 50 years, of which 69 % are female, 81 % Euro-American, 13 % Afro-American, and 6 % other groups. Subjects were instructed to perform a series of 23 facial displays. In total, 593 videos of 10–60 frames were recorded and annotated with an expression label in the form of an FACS code. All videos start with an onset neutral expression and end with the peak of the expression that the subject was asked to display. Additionally, landmark annotations are provided for all frames of all videos: 68 fiducial points have been marked on the face, sketching the most salient parts of the face shape.

3.1.3 Face features

We use face features very similar to the similarity normalized shape features (SPTS) and canonical normalized appearance features (CAPP) used in [17]. On the CK+ dataset, our features for a video consist of a 636-dimensional real-valued vector extracted from the peak expression frame of that video. 136 elements are encoding information about the face shape, while 500 elements encode information about the face appearance. We chose to subtract the onset frame data from the peak frame data, like it was done in [17], to avoid mixing our expression recognition problem with an unwanted identity component embodying static morphological differences. For that reason, the face features we use can be called “identity-normalized”.

3.1.4 Results

The three hierarchical classification methods of interest, i.e., SkNN, SSVM and MMR, are compared to their flat counterparts: kNN, Multiclass Kernel-based Vector Machines (MKSVM [5]) and “flat setup” MMR, i.e., MMR not exploiting the hierarchical information. For each tested method, there exists a main parameter, the tuning of which can have huge influence on the results. For SkNN and kNN, this parameter is the number of neighbors to consider during the test phase. For SSVM and MKSVM, the core parameter is the training parameter “C”, which, in the soft-margin approach, balances the allowed misclassification rate during the training procedure. We found empirically that, for MMR, using a polynomial kernel brings the best performances (whereas for SSVM and MKSVM we use a linear kernel), and the core parameter for MMR is therefore the degree of this polynomial kernel.

Figure 2 shows the hierarchical F-measure (hF) curves obtained for the facial expression recognition task. Globally, we can observe that hierarchical classification does not seem to outperform flat classification with either of the proposed hierarchical methods. Having a closer look at the highest points from each of those performance curves, i.e., the points with the best hF (Table 1), we can see that the flat and hierarchical approaches give very similar performances for this expression recognition problem.

3.2 3D shape classification

3.2.1 The problem

Given a tree taxonomy of 3D objects such as the one presented in Fig. 3, the task is to determine to which class a new object instance belongs, based on its 3D shape feature representation.

3.2.2 The princeton shape benchmark (PSB)

The PSB dataset (Fig. 3) [27] is one of the largest and most heterogeneous datasets of 3D objects: 1814 3D models corresponding to a wide variety of natural and man-made objects are grouped into 161 classes. These models encode the polygonal geometry of the object they describe. The grouping was based not only on semantic and functional concepts

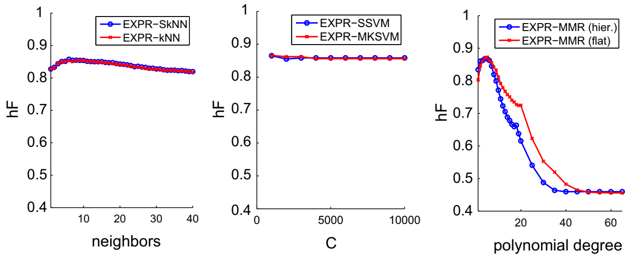

Fig. 2 Facial expression recognition results. *Blue* and *red curves* show hF for hierarchical and flat classification respectively, against the number of neighbors for SkNN vs. kNN (*left*), the "C" parameter for SSVM vs. MKSVM (*center*), and the degree of the polynomial kernel for hierarchical vs. flat setup MMR (*right*)

Table 1 Best hF performances from Fig. 2, along with the corresponding hP and hR performances obtained for the facial expression recognition task

	hP (%)	hR (%)	hF (%)
SkNN	83.63	88.00	85.76
kNN	83.12	87.98	85.48
SSVM	85.22	87.87	86.52
MKSVM	85.68	87.54	86.60
MMR (hier.)	85.84	87.76	86.79
MMR (flat)	86.46	88.07	87.26

(e.g., "furniture" is a superclass of "table") but also on shape attributes (e.g., round tables belong to the same class).

3.2.3 3D shape descriptors

Each object instance is encoded into a point cloud which is sampled from its original mesh file: 5000 points from the triangulated surface, where the probability of a point being selected from a triangle is related to the area of the triangle that contains it. From this sampling, we calculate five 3D descriptors for each object: ensemble of shape functions (ESF) [34], viewpoint feature histogram (VFH) [24], intrinsic spin images (SI) [32], signature of histograms of orientations (SHOT) [30] and unique shape contexts (USC) [4]). The reasons for choosing those descriptors are (1) uniqueness (preference to heterogeneity of algorithms) and (2) accessibility (the methods used are available from the point cloud library [25]). By applying our methods to different descriptors, we wish to multiply the classification experiments to enhance our comparison between hierarchical and flat methods for the 3D shape classification problem.

3.2.4 Results

We perform 3D shape classification with each of the five descriptors, i.e., ESF, VFH, SI, SHOT and USC, using each of the three hierarchical classification methods of interest, i.e., SkNN, SSVM and MMR, as well as their flat counterparts, i.e., kNN, MKSVM and "flat setup" MMR. Again, we make the most influential parameter for each method vary in our tests; those are the number of neighbors for SkNN and kNN, the "C" parameter for SSVM and MKSVM, and the degree of the polynomial kernel for MMR.

Figure 4 shows the hierarchical F-measure (hF) curves obtained for all test cases. There seems to be, for some of the five descriptors, a consistent yet very slight trend showing some performance improvement when using hierarchical classification. Indeed, the VFH and ESF descriptors seem to benefit a little from hierarchical information in all three methods, as it is further illustrated in Table 2 which gives details about the best hF values obtained for all test cases. For SI, SHOT and USC descriptors, results are mixed: either hierarchical or flat classification performs slightly better, depending on the method. Again, hierarchical classification does not clearly appear to give better results than flat classification but for a few cases.

4 Artificial classification problems

4.1 Motivation

The results presented in Sect. 3 for real-world problems are not easy to interpret. After systematically applying three different hierarchical classification methods to two different vision-based problems with several different types of features, we failed to showcase the superiority of the hierarchical approach over the flat one for classification based on visual features. However, as stated in Sect. 1, such superiority (1) has been demonstrated in general in other fields, such as text categorization and protein function prediction, and (2) would have been expected as suggested by neurophysiological studies of the visual cortex.

In our previous work, we hypothesized that the features we used, which are commonly used in 2D and 3D computer vision for general purpose, might lack the information necessary to exploit a hierarchical prior. Based on that hypothesis, we can ask ourselves three further questions about the underlying causes:

1. Do the features fail to capture any hierarchical information by nature?
2. Do the features capture hierarchical information structurally different from the hierarchical prior?
3. Do the features capture hierarchical information structurally similar to the hierarchical prior, with so much noise that our hierarchical methods fail?

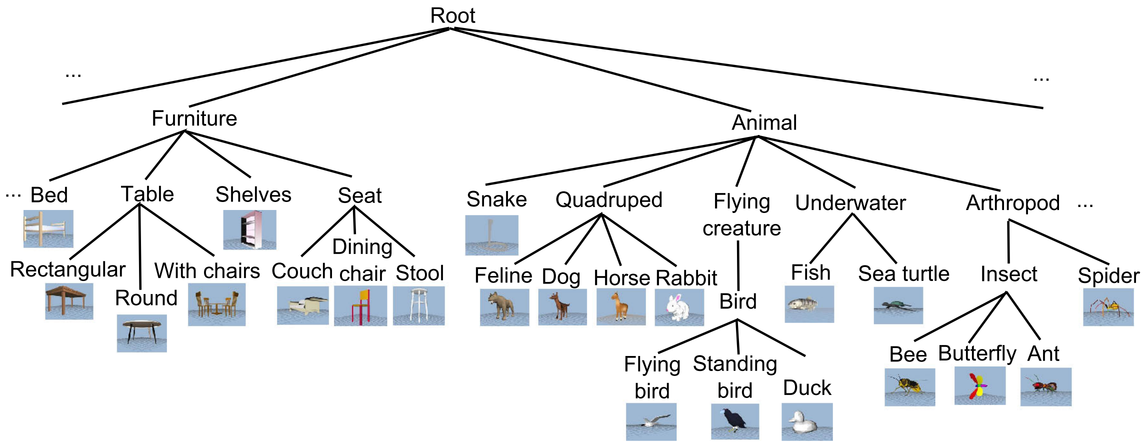

Fig. 3 The "Furniture" and "animal" sub-trees of the Princeton Shape Benchmark, with snapshots of some of the models that belong to the leaves (classes) of those sub-trees

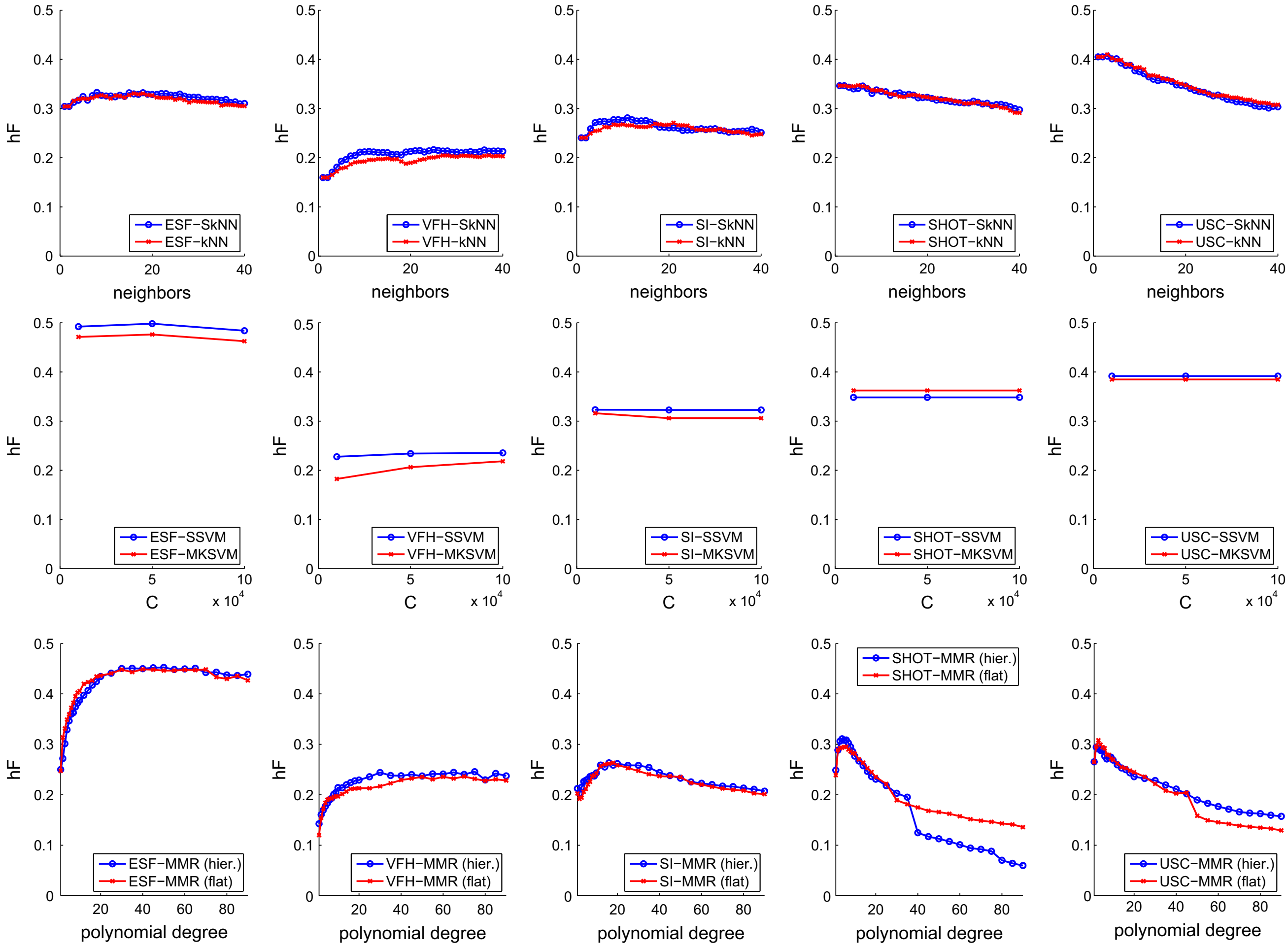

Fig. 4 3D shape classification results. *Blue* and *red curves* show hF for hierarchical and flat classification respectively, against the number of neighbors for SkNN vs. kNN in the first row, the "C" parameter for SSVM vs. MKSVM in the second row and the degree of the polynomial kernel for MMR (hierarchical vs. flat setup) in the third row. Each column corresponds to the use of a particular descriptor: ESF, VFH, SI, SHOT and USC

Table 2 Best hF performances from Fig. 4, along with the corresponding hP and hR performances obtained for the 3D shape classification task using the shape descriptors ESF, VFH, SI, SHOT and USC

	Measure	ESF	VFH	SI	SHOT	USC
SkNN	hP (%)	32.23	20.38	27.24	34.36	40.26
	hR (%)	34.40	23.07	29.07	34.95	41.08
	hF (%)	33.28	21.64	28.12	34.65	40.67
kNN	hP (%)	32.00	19.60	26.42	33.99	40.78
	hR (%)	34.22	21.42	27.79	35.48	41.18
	hF (%)	33.07	20.47	27.09	34.72	40.98
SSVM	hP (%)	49.72	23.47	31.15	33.43	37.58
	hR (%)	49.92	23.62	33.58	36.35	40.88
	hF (%)	49.82	23.55	32.32	34.83	39.16
MKSVM	hP (%)	47.78	21.84	31.01	35.79	37.56
	hR (%)	47.45	21.84	32.23	36.67	39.41
	hF (%)	47.61	21.84	31.61	36.22	38.46
MMR (hier. setup)	hP (%)	45.56	24.70	26.07	30.35	28.40
	hR (%)	44.93	24.44	26.57	31.86	30.53
	hF (%)	45.24	24.57	26.32	31.09	29.43
MMR (flat setup)	hP (%)	44.72	23.63	26.05	28.98	29.96
	hR (%)	45.02	23.62	26.62	30.03	31.70
	hF (%)	44.87	23.62	26.33	29.50	30.81

To give answers to these rather general questions, we believe that it is a good strategy to not focus into a specific problem but instead consider a general approach. To do so, we have designed a simulation framework which generates abstract, artificial classification problems, the complexity of which can be controlled through the manipulation of key aspects for hierarchical classification. From the results obtained with our hierarchical and flat classification methods applied to these artificial problems, we wish to draw useful insights about the conditions under which the hierarchical approach can offer a real gain in performance.

4.2 Simulation framework

4.2.1 Abstraction of the classification problem

To build a meaningful simulation framework, we need to have a clear view of the concepts at work in the hierarchical and flat classification approaches (Fig. 5). In abstract terms, the repeated manifestation of a phenomenon is measured by a sensor on one hand, and a semantic classification of the possible states of the phenomenon made by an observer on the other hand. We are interested in phenomena that have a natural hierarchical relationship between their states, i.e., an *underlying* taxonomy.[1] Being aware of the hierarchical nature of a phenomenon, the observer may organize the semantic classes in a hierarchical manner, i.e., define a *perceived* taxonomy. This perceived taxonomy does however not necessarily correspond perfectly to the natural underlying taxonomy.

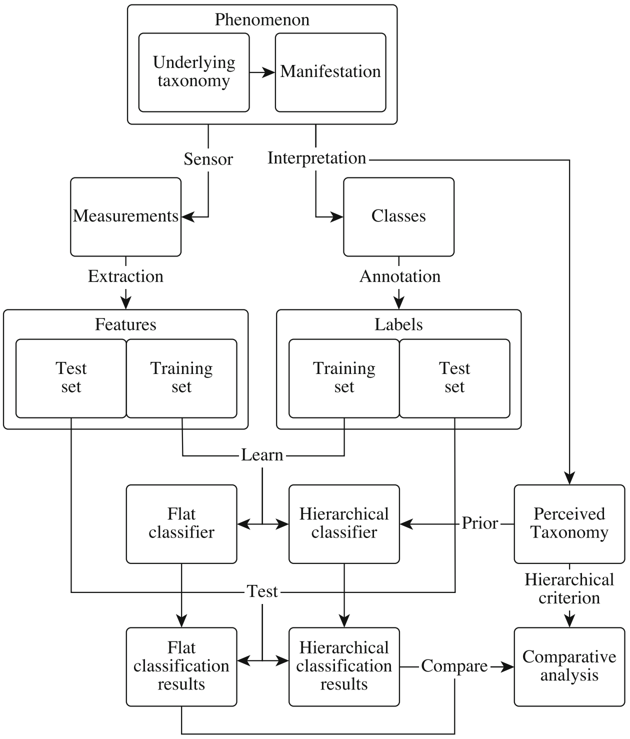

Fig. 5 Schematic view of the hierarchical and flat classification approaches used in our simulation framework

[1] It is arguable whether or not there exists such a thing as an "underlying taxonomy" for a phenomenon; taxonomies may be thought of as always being arbitrary, their value lying in their usefulness, not in some underlying, self-evident truth.

The semantic classes provided by the observer are then used to label a collection of measurements, yielding a labeled dataset. At the same time, a feature extraction method is applied to the collection of measurements, with the primary goal of capturing the essential characteristics present in the data for a supervised classification task. If we assume that the measurements contain information about the underlying taxonomy of the phenomenon,[2] a so-called high-level feature extraction should be able to capture at least part of this essential information, while low-level feature extraction is likely to fail to capture any of it.

A set of labeled features is therefore available for learning a classifier, i.e., a machine that predicts the class associated with new measurements of the same phenomenon, on the basis of features extracted using the same feature extraction method. To evaluate the generalization capability of a classifier, the set is split into a training and a test set. The training of a hierarchical classifier differs from the training of a flat one by the fact that the hierarchical learning method is given the *perceived* taxonomy as a prior, whereas the flat learning method does not make use of a hierarchical prior about the classes.

Making use or not of a hierarchical prior during learning, all other things remaining equal, the classification performances of the hierarchical and flat classifiers can be compared on the basis of the perceived taxonomy which, in this case, is used as a hierarchical penalty criterion for the misclassification of the elements of the test set (e.g., using the hierarchical F-measure, see Sect. 2.1). Indeed, for both types of classifiers, the superclasses come as a byproduct of the predicted class according to a given taxonomy, and those superclasses can be used to penalize misclassification of examples, making emphasis on serious hierarchical errors according to the given taxonomy.

4.2.2 *Artificial datasets with taxonomies*

Following on what has been discussed in Sect. 4.2.1, we are interested in generating datasets obtained from phenomena with underlying taxonomies. To simulate such taxonomies, we consider perfect k-ary trees, where all leaf nodes are at the same level L (the root is at the level 1) and all internal nodes have degree k, i.e., k children. For such trees, the total number of nodes is given by $\frac{k^L-1}{k-1}$, and the number of leaf nodes is k^{L-1}. In our view, a path from the root to a leaf node of such a taxonomy corresponds to a state of the hierarchical phenomenon that is being measured by the sensor and interpreted by the observer. Note that we do not consider problems where a single manifestation of a phenomenon can simultaneously correspond to multiple paths in the underlying tree taxonomy. We consider 10 different phenomena with such underlying taxonomies (see Table 3). For each phenomenon, we assume (1) that the observer was able to establish the existence of all the different states and make them correspond to semantic classes, and (2) that 200 manifestations per state were measured by the sensor and correctly class-labeled by the observer. We then have 10 labeled datasets which are perfectly class-balanced.

Table 3 Underlying taxonomies of the phenomena under consideration in our simulation experiments

	k	L	#nodes	#leaves (classes)
Binary trees	2	3	7	4
	2	4	15	8
	2	5	31	16
	2	6	63	32
	2	7	127	64
Ternary trees	3	3	13	9
	3	4	40	27
	3	5	121	81
Quadtrees	4	3	21	16
	4	4	85	64

In our view, the observer has also established a perceived taxonomy embodying the hierarchical relationships between the semantic classes. For each dataset, we consider a first experimental simulation condition where the perceived taxonomy perfectly matches the actual underlying taxonomy of the phenomenon associated with this dataset. We also consider a second simulation condition where the perceived taxonomy does not match the underlying one at different degrees. The artificial problems generated with this second condition simulate real problems where the chosen taxonomies are arbitrary and do not optimally reflect the hierarchical nature of the phenomenon. To test this second simulation condition, we focus on the dataset associated with the underlying binary tree taxonomy with 7 levels (127 nodes, 64 leaves).

4.2.3 *Artificial high-level features*

We assume that the measurements in our datasets somehow encode the underlying hierarchical nature of the phenomenon, which applies in most practical cases. For each dataset, we simulate a series of high-level feature extractions with different levels of noise. More precisely, a combination of a dataset and a noise level yields a unique classification problem involving the noisy features and the class labels for this dataset (as well as the perceived taxonomy as a prior when hierarchical classification is considered).

[2] The measurements may comply particularly well to a specific taxonomic model, that would be the best, i.e., most useful taxonomic approximation of the nature of the phenomenon.

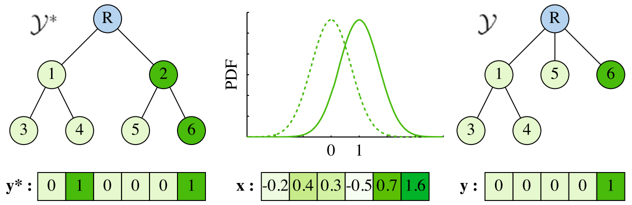

Fig. 6 *Left* an underlying taxonomy $\mathcal{Y}^*$ and a representation $\mathbf{y}^*$ of a measurement in this taxonomy. *Center* a feature vector $\mathbf{x}$ for the measurement, generated from $\mathbf{y}^*$ with a noise level $\sigma^2 = 0.5$. *Right* a label $\mathbf{y}$ for the measurement, in a perceived taxonomy $\mathcal{Y}$ obtained from $\mathcal{Y}^*$ by the elimination of the interior node 2

Given a dataset and a noise level, the extraction of a feature vector $\mathbf{x} \in \mathbb{R}^n$ from a measurement is made using the representation $\mathbf{y}^*$ of this measurement in the *underlying* taxonomy $\mathcal{Y}^* \subset \{0, 1\}^n$ of the phenomenon associated with the dataset, i.e.,

$$\mathbf{x}_i \sim \mathcal{N}(\mathbf{y}_i^*, \sigma^2) \quad \forall\, i \in \{1, \ldots, n\}, \quad \mathbf{y}^* \in \mathcal{Y}^*, \tag{5}$$

where σ^2 is a Gaussian noise variance, which embodies the noise level. For each of the 10 datasets, we consider 51 progressive noise levels (and therefore 51 classification problems), by choosing σ^2 in $\{0, 0.05, \ldots, 2.5\}$.

In our design to generate the features, each feature in the feature vector is discriminative for one of the n nodes of the underlying taxonomy (see Fig. 6, left and center). Such features are high-level and capture the hierarchical information up to some degree of noise. With our first simulation condition, where the perceived taxonomy is defined as equivalent to the underlying taxonomy, i.e. where a hierarchical label $\mathbf{y} \in \mathcal{Y}$ is equal to $\mathbf{y}^* \in \mathcal{Y}^*$, these features are actually discriminative for the classes and superclasses of the hierarchical classification problem. However with our second simulation condition, where the perceived taxonomy differs from the underlying one, these features may be less discriminative for the superclasses, which do not exactly represent the real hierarchical nature of the phenomenon (see Fig. 6, right).

4.2.4 Results

To avoid overloading the reader with excessive experimentation, we only show and discuss results relative to the methods SkNN and SSVM, and their respective flat counterparts kNN and MKSVM (see Sect. 2.1). We also consider only one case for the value of their most influential parameter, i.e., the number of neighbors $k = 10$ for SkNN and kNN and the training parameter $C = 100$ for SSVM and MKSVM. Those values were empirically obtained to be near-optimal for all methods given our artificial classification problems. For each classification problem, we split the set of labeled features into a training and a test set of the same size, i.e., with 100 examples per class for both the training and test sets.

Figures 7 and 8 show the results for the experiments with the first simulation condition (Sect. 4.2.2), for SkNN vs. kNN and SSVM vs. MKSVM, respectively, applied to all of our artificial classification problems. In this type of simulation where the underlying taxonomy is perfectly perceived by the observer and used as a prior for hierarchical learning, we can see the hierarchical approach to classification outperforms the flat approach for all test cases where the noise level is non-zero. The gain in performance is even more pronounced when the number of classes is larger (using deeper trees or larger tree degrees), with up to 13.31 % hF gain for the 7-level binary tree (64 classes), 11.99 % hF gain for the 5-level ternary tree (81 classes), and 11.56 % hF gain for the 4-level quadtree (64 classes). Table 4 gives quantitative results for all test cases. It can be noticed that the median hierarchical gain over the range of noise levels increases with the number of taxonomy levels.

To test the second simulation condition, we applied SkNN vs. kNN to classification problems involving the dataset associated with the underlying 7-level binary tree taxonomy (Sect. 4.2.2). In this type of simulation, a perceived taxonomy different from the underlying taxonomy is used both for training the hierarchical classifiers and calculating the hierarchical F-measure. We simulate two perceptual errors that the observer could make when defining the perceived taxonomy: (1) ignoring or missing some of the hierarchical relationships between the states of the phenomenon, and (2) creating hierarchical relationships that do not exist between the states of the phenomenon. In practice, defining a perceived taxonomy containing the first (second) type of error corresponds to removing (swapping) interior nodes in the underlying taxonomy.

Figures 9 and 10 show the results obtained with progressive interior node elimination and substitution, respectively. Table 5 gives quantitative results for these experiments. We can see that interior node elimination does not hamper the superiority of the hierarchical classification approach over the flat one, up to 90 % node removal. This can be explained by the fact that this type of misinterpretation of the hierarchical phenomenon does not violate the underlying “IS-A” or “PART-OF” partial order present in the underlying taxonomy. Therefore, providing a prior taxonomy that is even severely altered by this error type is still beneficial to the classification problem. However, in the case of interior node substitution, the gain in performance with the hierarchical approach steadily decreases with the proportion of nodes swapped. Indeed, this type of misinterpretation violates the natural hierarchical order present in the underlying taxonomy of the phenomenon.

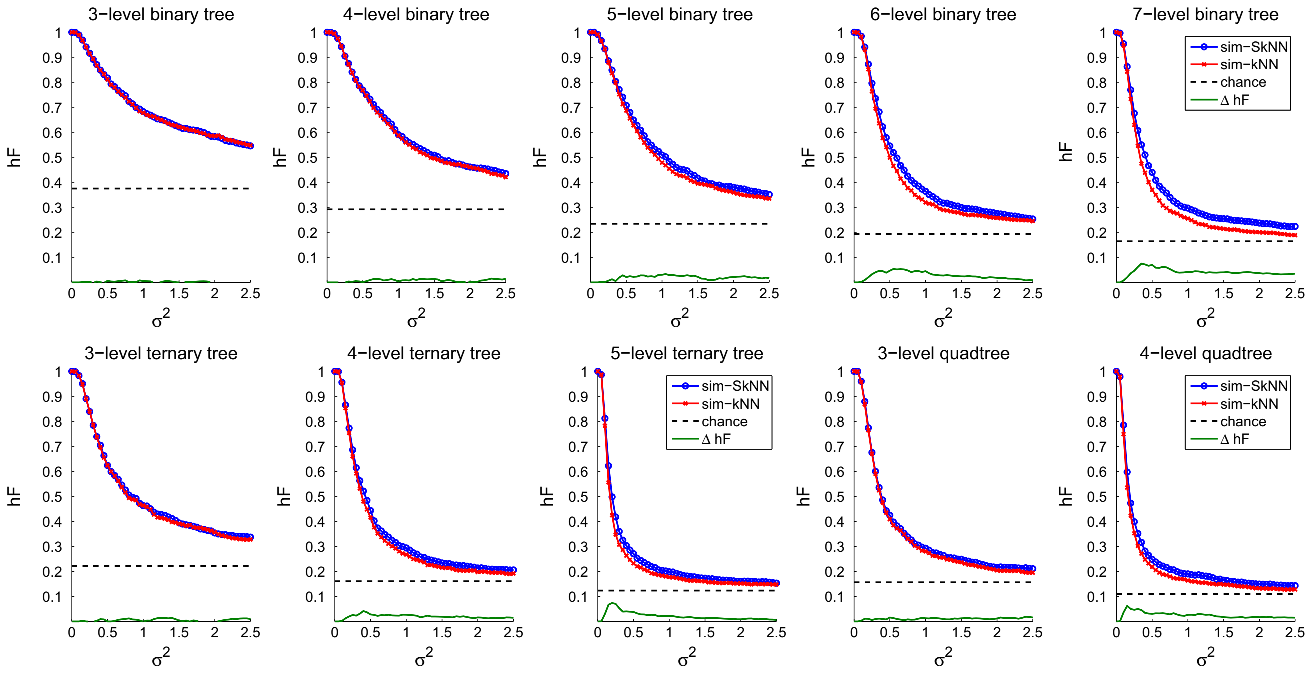

Fig. 7 SkNN vs. kNN results for the first simulation condition, using binary (*top row*) and ternary/quad trees (*bottom row*). *Blue and red curves* show hF for the hierarchical and flat classification, respectively, against the level of noise used in feature extraction. *Dashed black lines* show the chance level for hF. *Green curves* show ΔhF, i.e., the performance gain in hF with the hierarchical approach

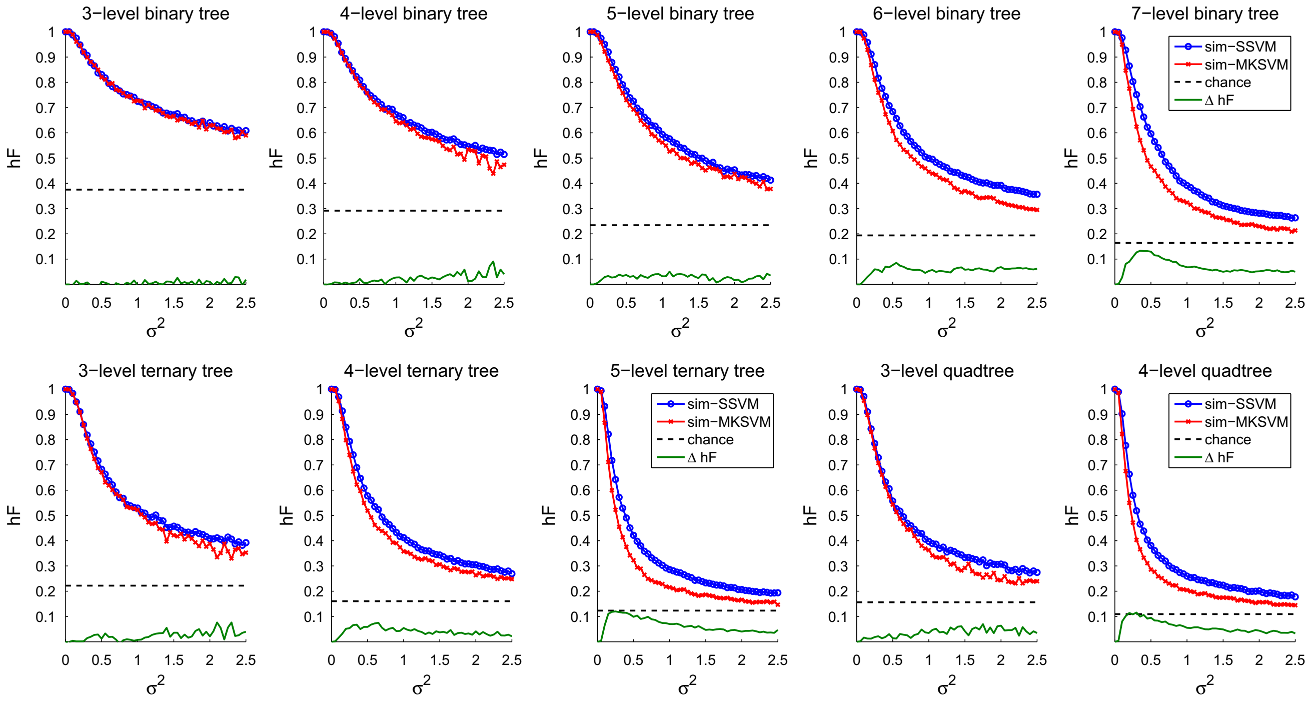

Fig. 8 SSVM vs. MKSVM results for the first simulation condition, using binary (*top row*) and ternary/quad trees (*bottom row*). *Blue and red curves* show hF for the hierarchical and flat classification, respectively, against the level of noise used in feature extraction. *Dashed black lines* show the chance level for hF. *Green curves* show ΔhF, i.e., the performance gain in hF with the hierarchical approach

Table 4 Median and maximal ΔhF, i.e., performance gains in hF with the hierarchical approach, in our results for the first simulation condition shown in Figs. 7 and 8, for SkNN vs. kNN and SSVM vs. MKSVM, respectively

	ΔhF with binary trees									
	L = 3		L = 4		L = 5		L = 6		L = 7	
	Med (%)	Max (%)	Med (%)	Max (%)	Med (%)	Max (%)	Med (%)	Max (%)	Med (%)	Max (%)
SkNN vs. kNN	0.10	0.80	0.73	1.49	2.10	3.32	2.48	5.35	3.97	7.52
SSVM vs. MKSVM	0.40	2.90	2.15	9.06	2.98	5.03	5.92	8.54	5.70	13.31
	ΔhF with ternary trees						ΔhF with quadtrees			
	L = 3		L = 4		L = 5		L = 3		L = 4	
	Med (%)	Max (%)	Med (%)	Max (%)	Med (%)	Max (%)	Med (%)	Max (%)	Med (%)	Max (%)
SkNN vs. kNN	0.35	1.39	1.95	4.16	1.48	7.36	1.12	1.86	2.21	6.22
SSVM vs. MKSVM	2.08	7.64	3.93	7.50	5.86	11.99	3.52	6.98	5.08	11.56

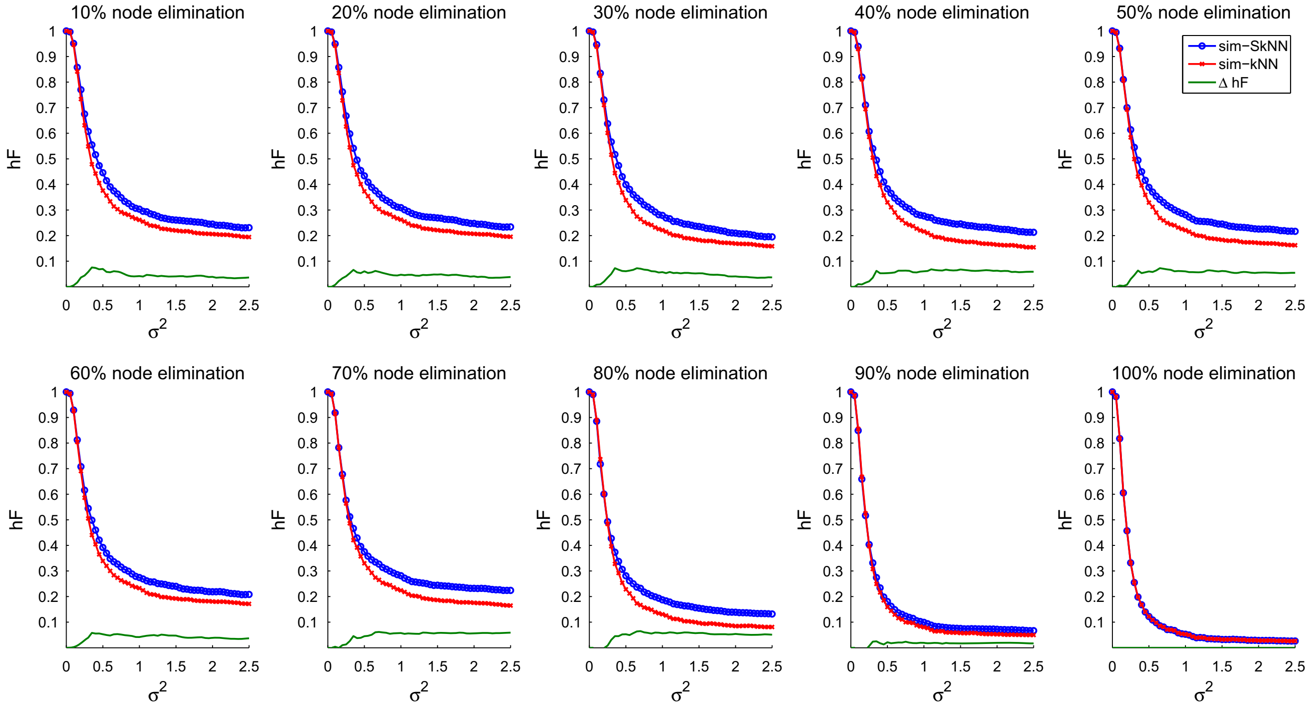

Fig. 9 SkNN vs. kNN results for the second simulation condition, with the elimination perceptual error on the underlying 7-level binary tree taxonomy. *Blue and red curves* show hF for the hierarchical and flat classification, respectively, against the level of noise used in feature extraction. *Green curves* show ΔhF, i.e., the performance gain in hF with the hierarchical approach

5 Discussion

Designing efficient computer vision-based recognition systems that could match the very strong human ability for visual recognition represents a difficult challenge, which has engaged the efforts of the computer vision community for several decades. Most of the practical computer vision-based recognition problems translate into hard classification tasks, for which a standard classification approach is typically used with either general-purpose features or complex task-specific feature representations. A promising avenue towards unifying the solutions to such problems is to try to better emulate the way in which humans classify visual content, notably by modeling the human visual system through biologically inspired feature representations, which hold some structure that follows the organization of the visual cortex (e.g., [20,26,33]). The use of such high-level features has indeed been shown to improve classification performances [21,22,26].

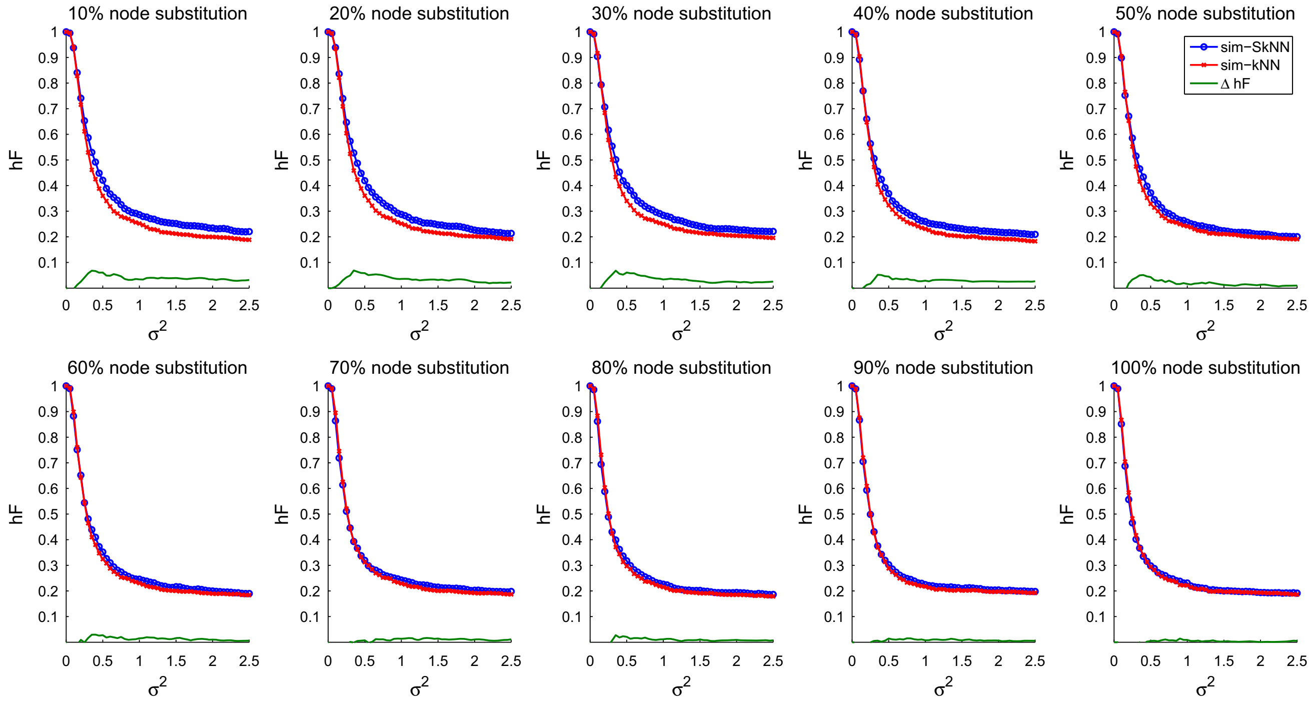

Fig. 10 SkNN vs. kNN results for the second simulation condition, with the substitution perceptual error on the underlying 7-level binary tree taxonomy. *Blue and red curves* show hF for the hierarchical and flat classification, respectively, against the level of noise used in feature extraction. *Green curves* show ΔhF, i.e., the performance gain in hF with the hierarchical approach

Table 5 Median and maximal ΔhF, i.e., performance gains in hF with the hierarchical approach, in our results for the second simulation condition shown in Figs. 9 and 10, for interior node elimination and substitution, respectively

	SkNN vs. kNN with the 7-level binary tree altered by interior node elimination									
Elimination ratio (%)	10	20	30	40	50	60	70	80	90	100
Med(ΔhF) (%)	4.08	4.51	5.05	6.17	5.67	4.13	5.67	5.29	1.82	0.00
Max(ΔhF) (%)	7.58	6.62	7.28	6.82	7.28	5.80	6.17	6.50	2.51	0.00
	SkNN vs. kNN with the 7-level binary altered by interior node substitution									
Substitution ratio (%)	10	20	30	40	50	60	70	80	90	100
Med(ΔhF) (%)	3.63	3.26	2.62	2.70	1.45	1.19	1.03	0.78	0.70	0.29
Max(ΔhF) (%)	6.80	6.90	6.79	5.19	5.11	2.95	1.63	2.77	1.53	1.45

Inspired by how humans organize visual objects into taxonomies where classes share some level of semantic similarity, another path for improvement is to apply a hierarchical approach to classification, i.e., to use a taxonomy embodying such semantic hierarchical relationships as a prior for the supervised learning process. This is also motivated by the superiority of the hierarchical classification approach in other fields such as text categorization and protein function prediction [7,23,28], where the features are typically high-level and where the possible states of the observed phenomenon are connected via well-understood hierarchical relationships. Enforcing such a hierarchical prior to the classification of visual content has been shown to be advantageous in some works (e.g., [3,6]), but far less often than in other fields [28]. Particularly in our previous work [11], the results of which are also reported in Sect. 3, we found that there was no added value in using a straightforward hierarchical approach with general-purpose features and descriptors, for the tasks of facial expression recognition and 3D shape classification. However, we showed via simulation experiments in this work that the hierarchical methods we used in [11] consistently outperform their flat counterparts with high-level features capturing the underlying hierarchical relationships present in the data, even when strong noise is added to those features. We also showed that the advantage of the hierarchical approach disappears when the enforced prior taxonomy contains hierarchical perceptual errors with respect to the underlying taxonomy of the phenomenon from which the data were obtained.

Based on our work, we believe that vision-based classification systems can benefit from hierarchical classification under the following conditions:

1. The features must be high level and designed to capture the underlying hierarchical information present in the measurements of the visual phenomenon.
2. The underlying hierarchical nature of the visual phenomenon must be well-understood for the hierarchical prior to be helpful.

About the first condition, high-level hierarchical feature representations can be obtained through biologically inspired design [20,26,33] or example-driven discovery which includes information transfer [9,13] and hierarchy learning [10,16,18,19,29]. In our real-world problems (Sect. 3), the features we used were not designed to capture hierarchical information. Also, the work on 3D shape classification presented in [3], which is related to our second real-world problem, showed improved classification performances by training local binary classifiers for the nodes of a prior taxonomy, which actually yielded in practice the production and aggregation of high-level features in a hierarchical representation.

Regarding the second condition, a deep and accurate understanding of the semantics behind a visual phenomenon should be acquired before the hierarchical learning process. Ways to obtain such information include hierarchy discovery from labeled examples with focus on the semantics, possibly using "human in the loop" strategies to ensure that the discovered hierarchies are semantically meaningful. Such discovery could also be performed jointly with the design of high-level hierarchical feature representations, e.g., through building on a strategy similar to [16].

6 Conclusion

The original hypothesis for designing our work was that computer vision-based systems should consistently benefit from using the hierarchical approach to classification. We failed to prove this hypothesis through our experiments on real-world problems, even though state-of-the-art hierarchical classification methods and feature descriptors were used. Via simulation experiments, we showed how crucial feature representation is for the hierarchical approach to offer a real gain in performance over the flat approach. We also showed how the misinterpretation of the underlying hierarchical nature of a phenomenon may hamper this performance gain. In light of these real-world and simulation results, we believe that, in the context of hierarchical classification based on visual content, the focus should be given to the design of high-level hierarchical feature representations and to a deep understanding of the semantics behind visual phenomena.

References

1. Astikainen, K., Holm, L., Pitkänen, E., et al.: Towards structured output prediction of enzyme function. In: Proc. of BMC'08, BioMed Central, vol. 2, p. S2 (2008)
2. Baldassi, C., Alemi-Neissi, A., Pagan, M., et al.: Shape similarity, better than semantic membership, accounts for the structure of visual object representations in a population of monkey inferotemporal neurons. PLOS Comput. Biol. **9**(e1003), 167 (2013)
3. Barutcuoglu, Z., DeCoro, C.: Hierarchical shape classification using Bayesian aggregation. In: SMI'06, IEEE, pp. 44–44 (2006)
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI **24**(24), 509–522 (2002)
5. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. JMLR **2**, 265–292 (2001)
6. Dimitrovski, I., Kocev, D., Loskovska, S., Džeroski, S.: Hierarchical annotation of medical images. Pattern Recognit. **44**(10), 2436–2449 (2011)
7. Eisner, R., Poulin, B., Szafron, D., et al. (2005) Improving protein function prediction using the hierarchical structure of the gene ontology. In: Proc. of CIBCB'05, IEEE, pp 1–10
8. Ekman, P., Rosenberg, E.L.: What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system (FACS). Oxford University Press, Oxford (1997)
9. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. PAMI **28**(4), 594–611 (2006)
10. Griffin, G., Perona, P.: Learning and using taxonomies for fast visual categorization. In: Proc. of CVPR'08, IEEE, pp. 1–8 (2008)
11. Hoyoux, T., Rodríguez-Sánchez, A.J., Piater, J.H., Szedmak, S.: Can computer vision problems benefit from structured hierarchical classification? In: Proc. of CAIP'15, pp. 403–414. Springer, Berlin (2015)
12. Kiritchenko, S., Matwin, S., Famili, A.F.: Functional annotation of genes using hierarchical text categorization. In: Proc. of BioLINK SIG: LLIKB'05 (2005)
13. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. PAMI **36**(3), 453–465 (2014)
14. Lee, J.H., Downie, J.S.: Survey of music information needs, uses, and seeking behaviours: preliminary findings. In: Proc. of ISMIR'04, Citeseer, vol 2004, p. 5 (2004)
15. Lee, T.S., Mumford, D.: Hierarchical bayesian inference in the visual cortex. JOSA A **20**(7), 1434–1448 (2003)
16. Li, L.J., Wang, C., Lim, Y., et al.: Building and using a semantivisual image hierarchy. In: Proc. of CVPR'10, IEEE, pp. 3336–3343 (2010)
17. Lucey, P., Cohn, J.F., Kanade, T., et al: The extended Cohn–Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In: Proc. of CVPRW'10, pp. 94–101 (2010)
18. Marszałek, M., Schmid, C.: Constructing category hierarchies for visual recognition. In: Proc. of ECCV'08, pp. 479–491. Springer, Berlin (2008)

19. Mittelman, R., Sun, M., Kuipers, B., Savarese, S.: A Bayesian generative model for learning semantic hierarchies. Frontiers in psychology **5**, 417 (2014)
20. Rodríguez-Sánchez, A., Tsotsos, J.: The roles of endstopped and curvature tuned computations in a hierarchical representation of 2D shape. PLOS One **7**(8), 1–13 (2012)
21. Rodríguez-Sánchez, A.J., Tsotsos, J.K.: The importance of intermediate representations for the modeling of 2d shape detection: Endstopping and curvature tuned computations. In: Proc. of CVPR'11, IEEE, pp. 4321–4326 (2011)
22. Rodríguez-Sánchez, A.J., Szedmak, S., Piater, J.: SCurV: A 3D descriptor for object classification. In: Proc. of IROS'15 (2015)
23. Ruiz, M.E., Srinivasan, P.: Hierarchical text categorization using neural networks. Inf. Retr. **5**(1), 87–118 (2002)
24. Rusu, R., Bradski, G., Thibaux, R., Hsu, J.: Fast 3D recognition and pose using the viewpoint feature histogram. In: Proc. of IROS'10, pp. 2155–2162 (2010)
25. Rusu, R.B., Cousins, S.: 3D is here: Point cloud library (PCL). In: Proc. on ICRA'11, pp. 1–4 (2011)
26. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M.: Robust object recognition with cortex-like mechanisms. PAMI **29**(3), 411–426 (2007)
27. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton Shape Benchmark. In: Proc. of SMI'04, pp. 167–178 (2004)
28. Silla Jr., C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. DMKD **22**(1–2), 31–72 (2011)
29. Snow, R., Jurafsky, D., Ng, A.Y.: Semantic taxonomy induction from heterogeneous evidence. In: Proc. of ACL'06, association for computational linguistics, pp. 801–808 (2006)
30. Tombari, F., Salti, S., Di Stefano, L.: Unique shape context for 3D data description. In: Proc. of ACM'10, ACM, pp. 57–62 (2010)
31. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proc. of ICML'04, ACM, p. 104 (2004)
32. Wang, X., Liu, Y., Zha, H.: Intrinsic spin images: a subspace decomposition approach to understanding 3D deformable shapes. In: Proc. of 3DPVT'10, vol. 10, pp. 17–20 (2010)
33. Weidenbacher, U., Neumann, H.: Extraction of surface-related features in a recurrent model of V1–V2 interactions. PLOS One **4**(6), e5909 (2009)
34. Wohlkinger, W., Vincze, M.: Ensemble of shape functions for 3D object classification. In: Proc. of ROBIO'11, pp. 2987–2992 (2011)
35. Yamane, Y., Carlson, E., Bowman, K., et al.: A neural code for three-dimensional object shape in macaque inferotemporal cortex. Nat. Neurosci. **11**(11), 1352–1360 (2008)

Thomas Hoyoux received his M.Sc. degree in computer science from the University of Liège, Belgium, where he is currently working and pursuing a Ph.D. under the supervision of Prof. Jacques Verly at the laboratory for Signal and Image Exploitation. From 2010 to 2013, he joined the University of Innsbruck, Austria, as a research assistant in the Intelligent and Interactive Systems group led by Prof. Justus Piater. His research interests are broadly in the areas of computer vision and machine learning, with particular emphasis on applications to the analysis of facial expressions for the extraction of high-level semantics.

Antonio J. Rodríguez-Sánchez is currently an assistant professor in the Intelligent and Interactive Systems group of the Institute of Computer Science at Universität Innsbruck (Austria) under the supervision of Prof. Justus Piater. He was born in Santiago de Compostela (A Coruña), a beautiful city in the northwest of Spain. He completed his Ph.D. at the Center for Vision Research in York University (Toronto, Canada) in 2010 on modeling attention and intermediate areas of the visual cortex under the supervision of John K. Tsotsos. He obtained the degree of M.Sc. in computer science at the Universidade da Coruña (Spain) in 1998. He received his B.Sc. in computer science at Universidad de Córdoba (Spain) with honors. His current research interests include computational neuroscience, computer vision, machine learning and neural networks.

Justus H. Piater is a professor of computer science at the University of Innsbruck, Austria, where he leads the Intelligent and Interactive Systems group. He holds a M.Sc. degree from the University of Magdeburg, Germany, and M.Sc. and Ph.D. degrees from the University of Massachusetts Amherst, USA, all in computer science. Before joining the University of Innsbruck in 2010, he was a visiting researcher at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany, a professor of computer science at the University of Liège, Belgium, and a Marie-Curie research fellow at GRAVIR-IMAG, INRIA Rhône-Alpes, France. His research interests focus on visual perception, learning and inference in sensorimotor systems and other dynamic and interactive scenarios, and include applications in autonomous robotics and video analysis. He has published more than 150 papers in international journals and conferences, several of which have received best-paper awards, and currently serves as Associate Editor of the IEEE Transactions on Robotics.