Can Computer Vision Problems Benefit from Structured Hierarchical Classification?

Thomas Hoyoux[1], Antonio J. Rodríguez-Sánchez[2], Justus H. Piater[2], and Sandor Szedmak[2]

[1] INTELSIG, Montefiore Institute, University of Liège, Belgium
[2] Intelligent and Interactive Systems, Institute of Computer Science, University of Innsbruck, Austria

Abstract. While most current research in the classification domain still focuses on standard "flat" classification, there is an increasing interest for a particular type of structured classification called hierarchical classification. Incorporating knowledge about class hierarchy should be beneficial to computer vision systems as suggested by the fact that humans seem to organize objects into hierarchical structures based on visual geometric similarities. In this paper, we analyze whether hierarchical classification provides better performances than flat classification by comparing three structured classification methods – Structured K-Nearest Neighbors, Structured Support Machines and Maximum Margin Regression – with their flat counterparts on two very different computer vision tasks: facial expression recognition – for which we emphasize the underlying hierarchical structure – and 3D shape classification.

1 Introduction

Most current efforts in the classification domain involve multiclass or binary classification, also known as flat classification [14]. In these types of classification a bee, an ant and a hammer are considered to be different to the same degree: they belong to different classes in a flat sense because all classes are defined at the same semantical level. However ants and bees are in the same superclass of insects, while hammers belong to another superclass about tools. In the context of classification based on visual features, exploiting geometrical and part similarities in a hierarchical fashion seems to reflect the natural way in which humans recognize the objects they see.

Accordingly, it is only logical to believe that current computer vision systems can benefit from classifiers that are able to exploit geometrical similarities using a pre-established taxonomy in a similar way the human visual system does. A recent survey paper [14] reports that – after analyzing dozens of articles published over the past decade – the hierarchical approach to classification outperforms the flat approach for various tasks, though mostly non visual ones. One could however wonder how well methods built upon the hierarchical perspective compare to flat ones in a broader analysis.

In this paper, we examine the potential of the hierarchical classification approach for solving two inherently hierarchical computer vision problems. The first problem of interest is the recognition of the facial expressions, seen as a combination of Action Units (AUs) as defined in the Facial Action Coding System (FACS) [4]. The second problem of interest is that of 3D shape classification, for which we take five popular 3D shape descriptors into consideration. In practice there are various ways of implementing the concept of hierarchical classification. In this work, we are interested in global hierarchical classification methods as opposed to local ones. Global exploration means that one single classifier is trained and used by considering the entire class hierarchy at once. For both problems of interest, results obtained with Maximum Margin Regression [1] (MMR), Structured Support Vector Machines [16] (SSVM) and a structured output version of the standard K-Nearest Neighbor algorithm (which we call SkNN) are compared. All three methods implement the global approach to hierarchical classification which makes them especially suited to such a comparison. We compare as well with their flat counterparts, which are MMR not using structured information, a multiclass kernel-based SVM, and the standard kNN. In order to get a meaningful evaluation of the hierarchical approach potentiality, we take care of using appropriately designed evaluation measures.

Our starting hypothesis is that, through the use of an out-of-the-box different implementation of a face expression recognition method, and five popular 3D descriptors, hierarchical classification can improve performances over flat classification and that it is the case independently of the classifier used and the task to which it is applied. We would like to stress here that our aim is not to outperform previous facial expression recognition methods in our first set of experiments, or to show which descriptor of the ones used here is better in our second set of experiments.

The remainder of this paper is organized as follows. Section 2 describes the framework and terminology we adopt for defining our hierarchical classification problems and methods, then provides the details of the hierarchical methods used in this work. Section 3 presents our experimental results for the computer vision problems we take into consideration. Those results are further discussed in section 4 and conclusions are drawn in section 5.

2 Materials and methods

2.1 Framework and terminology

Recently a necessary effort to unify the hierarchical classification framework has been made by [14]. We follow on their terminology which is summarized next.

A class taxonomy $\mathcal{C}$ is a finite set of nodes $\{c_i \mid i = 1 \dots n\}$ enumerating all classes and superclasses, which can be organized as a tree or as a Directed Acyclic Graph (DAG). A *hierarchical classification problem* and its corresponding *hierarchical classification algorithm* deals with either multiple or single labeled path(s), 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 labeled path must cover all hierarchy levels. In all our applications, we use tree taxonomies with full depth labeling. For facial expression recognition, we have multiple labeled paths per instance (see section 3.1) and for 3D shape classification we have a single labeled path for each instance (see section 3.2). In the context of hierarchical classification, we assume a vectorial representation for a label $\mathbf{y} \in \mathcal{Y}$, more specifically a Boolean category vector – or indicator vector – representation, i.e. $\mathcal{Y} \coloneqq \{0,1\}^n$, where the i^{th} component of $\mathbf{y}$ takes value 1 if the sample belongs to the (super)class – i.e. hierarchy node – $c_i \in \mathcal{C}$, and 0 otherwise.

Evaluation measures used in classical flat classification may not be appropriate when comparing hierarchical algorithms to each other, or flat algorithms to hierarchical ones. Those measures do not penalize structural errors and do not consider that misclassification at different levels of the class hierarchy should be treated in different ways. We will adopt the following metrics [6], also recommended by [14]: hierarchical precision (hP), hierarchical recall (hR) and hierarchical f-measure (hF). These metrics are extensions of the classical precision, recall and F-score measures and reduce to them as special cases if applied to a flat classification problem.

$$hP = \frac{\sum_i |\hat{P}_i \cap \hat{T}_i|}{\sum_i |\hat{P}_i|}, \qquad hR = \frac{\sum_i |\hat{P}_i \cap \hat{T}_i|}{\sum_i |\hat{T}_i|}, \qquad hF = \frac{2 * hP * hR}{hP + hR}, \quad (1)$$

where $\hat{P}_i$ is the set of the most specific class(es) predicted for a test example i and all its (their) ancestor classes, and $\hat{T}_i$ is the set of the true most specific class(es) of a test example i and all its (their) ancestor classes.

2.2 Structured hierarchical classifiers

We modified the classical kNN classification method to make it able to cope with a structured vectorial output, that is, vectorial outputs which are guaranteed to respect a pre-defined class taxonomy. We call the resulting classification method Structured output K-Nearest Neighbors (SkNN). We train the SkNN classifier in the same way as the standard kNN classifier, i.e. projecting the training data instances into the feature space. The choice of the feature map ϕ is left to the user, as well as the metric ρ used for finding the neighbors and the number k of neighbors to consider. Let $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ be the training set of a hierarchical classification problem. Given the k nearest neighbors $\mathcal{N} = \{(\mathbf{x}_i, \mathbf{y}_i) \,|\, i \in \{1 \dots k\}\} \subset \mathcal{D}$ to a test data instance $\mathbf{x} \in \mathcal{X}$, the classification rule for SkNN is as follows:

$$\hat{\mathbf{y}}(\mathbf{x}; \mathcal{N}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \langle \sum_{i=1}^{k} w_i \frac{\mathbf{y}_i}{||\mathbf{y}_i||}, \frac{\mathbf{y}}{||\mathbf{y}||} \rangle, \quad (2)$$

where w_i are the weights attributed to the neighbors, about which different strategies exist; they can for example reflect the distances of the neighbors to

the test instance, i.e. $w_i = \rho(\phi(\mathbf{x}_i), \phi(\mathbf{x}))^{-1}$, or they can be the same for all neighbors, i.e. $w_i = 1/k$, or any other weighting strategy the user would find suitable.

Our second hierarchical classification method is the Structured output Support Vector Machine (SSVM) [16], which extends classical SVM to handle arbitrary output spaces with non-trivial structure. SSVM defines the relation between an input data point $\mathbf{x} \in \mathcal{X}$ and its prediction $\hat{\mathbf{y}} \in \mathcal{Y}$ on the basis of a joint score maximization:

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

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

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

For our third structured output classification method, we apply a Maximum Margin based Regression (MMR) technique, see for example in [1], which is also an extension of the classical SVM but having 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 the SVM can be interpreted as a linear operator mapping the feature vectors of input items into the space of the feature vectors of the outputs. Inference with MMR is done in the same way as with SSVM (Eq. (3)) with the same joint feature map definition (Eq. (4)).

For each proposed method, the inference argmax problem can be done 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 structured labels, which guarantees that the class taxonomy is respected at all times.

3 Experimental evaluation

3.1 Facial expression recognition

The problem. We define an expression using the Facial Action Coding System (FACS) [4] 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. They are associated with the action of a muscle or a group of muscles. The FACS describes more than a 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 Figure 1 for the face expression, inspired by how AUs are usually grouped when presented in the literature [4]. 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).

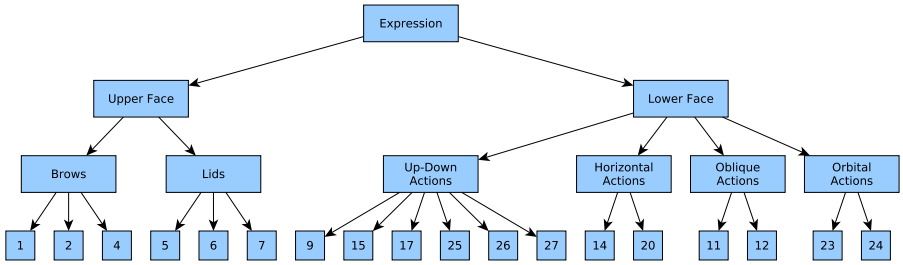
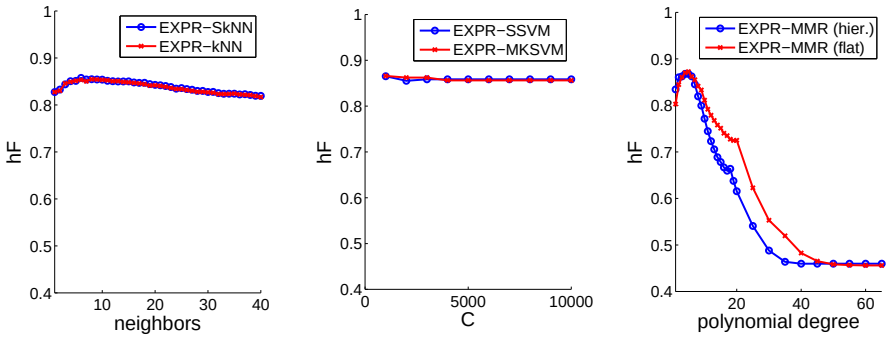


Fig. 1: Facial expression taxonomy. Leaves correspond to Action Units.

The Extended Cohn-Kanade Dataset (CK+). The CK+ dataset [8] consists of 123 subjects between the age of 18 to 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 sequences of 10 to 60 frames were recorded and annotated with an expression label in the form of a FACS code. All sequences 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 sequences: 68 fiducial points have been marked on the face, sketching the most salient parts of the face shape.

Face features. We use face features very similar to the similarity-normalized shape (SPTS) and canonical normalized appearance (CAPP) features used in [8]. On the CK+ dataset, they consist of a 636-dimensional real-valued vector for each video sequence. 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 [8], in order 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".

Results. The three hierarchical classification methods of interest – SkNN, SSVM and MMR – are compared to their flat counterparts – kNN, Multiclass Kernel based Vector Machines (MKSVM [3]) and "flat setup" MMR, i.e. MMR not exploiting the hierarchical information. For each tested method, there exists a main parameter whose tuning can have a large 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 "C" parameter, which – in the soft-margin approach – tells the SSVM optimization training process the allowed misclassification rate on each training sample. For MMR, the core parameter is the degree of the polynomial kernel used in the method. Fig. 2 shows the hierarchical F-measure (hF) curves obtained for the facial expression recognition task. We can observe that, globally, hierarchical classification does not outperform flat classification with either method on the proposed range of parameter values, and that it even performs less well than flat classification in the case of MMR. Having a closer look at the best – i.e. highest - hF points on each of those performance curves, one can see that the best classification results are not always in favor of hierarchical classification (Table 1). Surprisingly, they suggest that there is no improvement in the recognition rate when bringing high-level hierarchical information within the classification task of the face features. It can even be said that this additional information seems to bring confusion in the case of MMR.



(a) SkNN vs. kNN (b) SSVM vs. MKSVM (c) hier. vs. flat MMR

Fig. 2: Facial expression recognition results. Blue – resp. red – curves show hF for hierarchical – resp. flat – classification against (a) the number of neighbors for SkNN vs. kNN (b) the "C" parameter for SSVM vs. MKSVM (c) the degree of the polynomial kernel for MMR (hierarchical vs. flat setup).

3.2 3D shape classification

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

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

The problem. Given a tree taxonomy of 3D objects such as the one at Fig. 3, the task is to determine to which class (and ancestor classes) a new object instance belongs based on its 3D shape information.

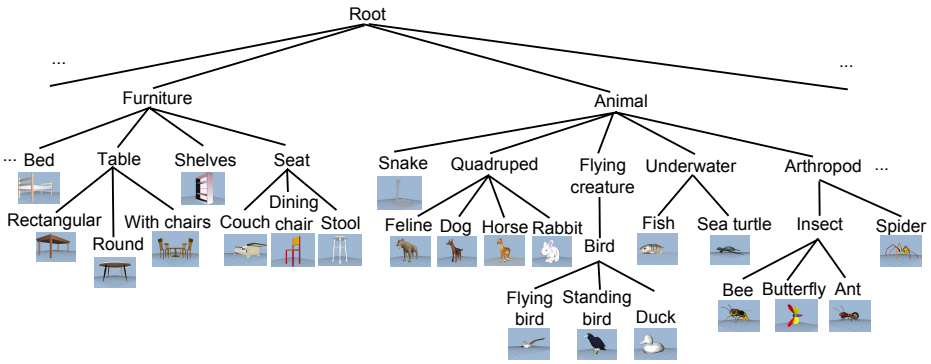


Fig. 3: Princeton Shape Benchmark dataset. Two superclasses are presented (Furniture, Animal), which show how they are hierarchically organized as well as some examples of each of the leaf classes.

The Princeton Shape Benchmark (PSB). The PSB dataset (Fig. 3) [13] is one of the largest and most heterogeneous dataset 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 done hierarchically and corresponds to how humans *see* the similarities between objects (e.g., an ant and a bee belong to the same superclass of insects).

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) [19],

Viewpoint Feature Histogram (VFH) [11], Intrinsic Spin Images (SI) [17], Signature of Histograms of Orientations (SHOT) [15] and Unique Shape Contexts (USC) [2]). The reason for choosing those are (1) Uniqueness (preference to heterogeneity of algorithms) (2) Accessibility (The methods used are available from the Point Cloud Library [10]). Our aim here is not to show which descriptor is the best, since some will work better under some conditions than others, but to show that independently of the *background* of the descriptor - as well as the classifier - there is a benefit in performing a hierarchical classification over a flat classification.

Results. 3D shape classification using five different descriptors – ESF, VFH, SI, SHOT and USC – is performed using each one of the three hierarchical classification methods of interest – SkNN, SSVM and MMR – as well as their flat counterparts – kNN, MKSVM [3] and "flat setup" MMR, i.e. MMR not exploiting the hierarchical information. Again, we make vary the most influencial parameter for each method in our tests; those are the number of neighbors for SkNN and kNN, the "C" parameter for SSVM and MKSVM (controlling the misclassification rate during training) and the degree of the polynomial kernel for MMR. All other parameters of the methods we consider remain fixed. Fig. 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 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. We discuss and comment further on these results in the next section.

4 Discussion

Object recognition and face analysis are very challenging problems for a computer, as indicated by the fact that for fifty years thousands of scientists have been working towards improved solutions. Some of those scientists have resorted to trying to model the human visual system, and many models have appeared, [12, 18, 9] to name a few. We think it is not enough to model neurons through mathematical approximations, we must also follow human strategies in order to improve our computer vision systems. One such strategy can be information transfer, which is currently being explored in the machine learning literature [5, 7].

Can a computer vision system benefit from classifying objects in a hierarchical fashion? This was the starting point of this study. We have selected two very different tasks; the first one was about recognizing facial expressions seen as complex hierarchical combinations of Action Units which, as atomical components

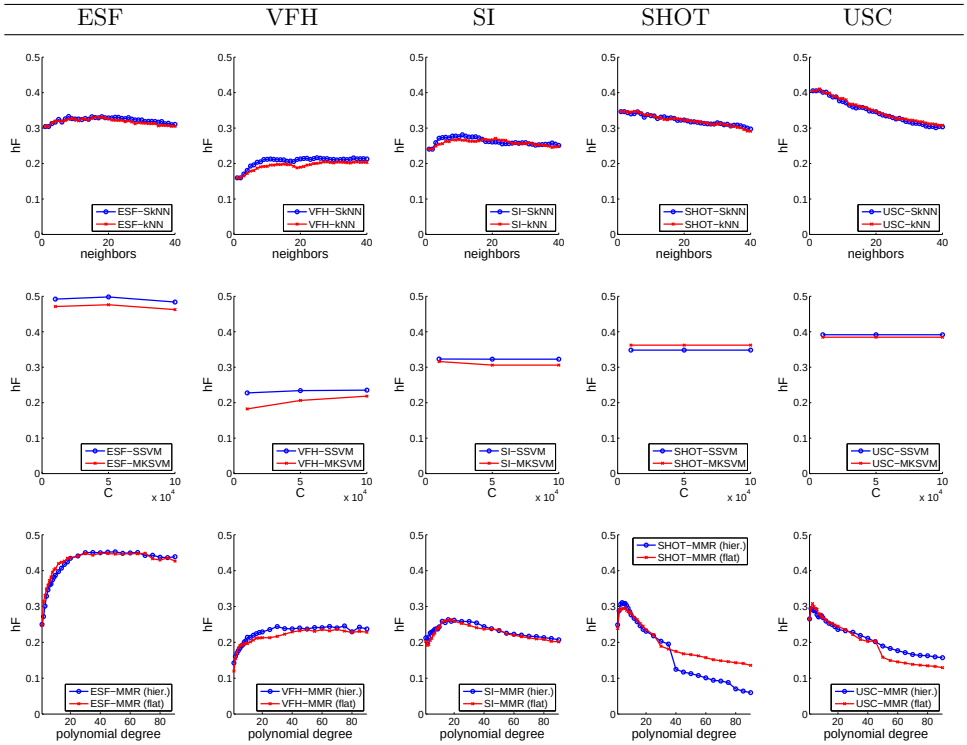


Fig. 4: 3D shape classification results. Blue – resp. red – curves show hF for hierarchical – resp. flat – classification against the number of neighbors for SkNN – resp. kNN – in the first row, the "C" parameter for SSVM – resp. MKSVM – in the second row and the degree of the polynomial kernel for hierarchical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

of the expression, define not only a multiclass but also a leaf-level multilabel hierarchical classification problem. The expression taxonomy we proposed and the face features we used in this work were inspired and supported by the expert literature. The second task involved the classification of a very large number of classes and objects. We have chosen one of the most heterogeneous datasets where there are objects as different in shape and meaning as a house, an ant, a tree or a plane. However the main reason for selecting this dataset was its similarity to how humans classify objects, which is related to our main hypothesis. Additionally we have used five different descriptors for this second task, and three different classifiers were compared for both tasks.

While we expected an improvement when using hierarchical classification this has not been the case for any of the two tasks. Even though a small number of combinations of descriptors-classifier benefited from it in the second task, the

	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%**

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

improvement was marginal, and in most cases the performances were similar for flat and for hierarchical classification. We would like to mention that we also tested many variations of the presented experiments. Those included PCA feature reduction, face features focusing more or only on the shape component, different ways to normalize the features, different weighting strategies during training and inference, different loss functions among which one based on the Hamming distance and another one on the hierarchical F-measure, etc. In all test cases the results obtained were very similar to the ones reported here.

At this point we may ask ourselves why this is the case, since in many other fields hierarchical classification boosts results over flat classification (protein function prediction, music genre classification, text categorization, etc.) even in some areas of computer vision – where it has been much less explored, though. Our hypothesis is that these computer vision methods based on structured classifiers fail to exploit the structure of the hierarchy because the features and descriptors commonly used carry no information about parts. Humans classify objects in terms of their geometric and parts similarities: a dog and a rabbit are quadruped animals, but there is no real representation for a *quadruped* animal when using current descriptors and geometric similarities are exploited to a very small extent due to the way these descriptors are created (section 3.2). In this sense, it is worth to mention that under some classification strategies, some descriptors seem to take advantage - even if minimal - of this geometric simi-

larity thanks to the inclusion of some local shape information. Our experiments also show that SSVM seems to be the best at this specific task although at the expense of much more computationally intensive training (training the SSVM classifier on half of the PSB dataset typically took between 30 and 50 hours, depending on which descriptor was used).

We strongly believe that computer vision systems have to follow the strategies of parts representations. As we have commented before, we mean not to criticize face descriptors or 3D shape descriptors which are very well suited for the environments for which they were developed. On the other hand we believe that there is still much room for improvement at the representation level in computer vision systems. Our study shows that structured classifiers cannot exploit hierarchical information for better, more efficient classification considering the current status of 3D shape representation and expression analysis. We propose that richer and more abstract representations are needed in order to advantageously emulate human strategies for better computer vision systems. This also echoes one of the conclusions from [14] stating that even though most researchers think that classes at different hierarchy levels are better discriminated by features of a different nature, not much attention has been given to how efficient feature selection for hierarchical classification should be performed, in particular in the global approach.

5 Conclusions

The starting hypothesis that led us to design this work was that computer vision systems – similarly to the human visual system – may benefit from structured class hierarchies by using classifiers that can exploit those structures and thus provide better classification results. Without disproving this hypothesis in general, our experimental work shows that there is still work to do in computer vision systems at the representation level, before structured machine learning methods can take full advantage of the information present in the hierarchical organization of objects or facial expressions.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7-ICT/2011-2015 (Challenge 2, Cognitive Systems and Robotics) under grant agreement no. 270273, "Xperience". This work has also been supported by the grant no. 600914 "PaCMan" (Probabilistic and Compositional Representations of Object for Robotic Manipulation) within the FP7-ICT-2011-9 program (Cognitive Systems).

Bibliography

[1] Astikainen, K., Holm, L., Pitkänen, E., Szedmak, S., Rousu, J.: Towards structured output prediction of enzyme function. In: BMC (2008)

[2] Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE PAMI 24(24), 509–522 (2002)

[3] Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. JMLR 2, 265–292 (2001)

[4] 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 (1997)

[5] Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE PAMI 28(4), 594–611 (2006)

[6] Kiritchenko, S., Matwin, S., Famili, A.F.: Functional annotation of genes using hierarchical text categorization. In: in BioLINK SIG: LLIKB (2005)

[7] Lampert, C., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot learning of object categories. IEEE PAMI (2013)

[8] Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: CVPRW. pp. 94–101 (2010)

[9] Rodríguez-Sánchez, A., Tsotsos, J.: The importance of intermediate representations for the modeling of 2D shape detection: Endstopping and curvature tuned computations. IEEE CVPR pp. 4321–4326 (2011)

[10] Rusu, R.B., Cousins, S.: 3D is here: Point cloud library (PCL). In: IEEE ICRA. pp. 1–4 (2011)

[11] Rusu, R., Bradski, G., Thibaux, R., Hsu, J.: Fast 3D recognition and pose using the viewpoint feature histogram. In: IROS. pp. 2155 –2162 (2010)

[12] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M.: Robust object recognition with cortex-like mechanisms. IEEE PAMI 29(3), 411–426 (2007)

[13] Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: Shape Modeling Applications. pp. 167–178 (2004)

[14] Silla Jr, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. DMKD 22(1-2), 31–72 (2011)

[15] Tombari, F., Salti, S., Di Stefano, L.: Unique shape context for 3D data description. In: workshop on 3D object retrieval. pp. 57–62. ACM (2010)

[16] Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML. p. 104. ACM (2004)

[17] Wang, X., Liu, Y., Zha, H.: Intrinsic spin images: A subspace decomposition approach to understanding 3D deformable shapes. In: 3DPVT. vol. 10, pp. 17–20 (2010)

[18] Weidenbacher, U., Neumann, H.: Extraction of surface-related features in a recurrent model of V1-V2 interactions. PLOS ONE 4(6), e5909 (06 2009)

[19] Wohlkinger, W., Vincze, M.: Ensemble of shape functions for 3D object classification. In: IEEE ROBIO. pp. 2987 –2992 (2011)