Feature Hierarchies for Object Classification

Boris Epshtein Shimon Ullman
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot, ISRAEL, 76100
{boris.epshtein , shimon.ullman}@weizmann.ac.il

Abstract

The paper describes a method for automatically extracting informative feature hierarchies for object classification, and shows the advantage of the features constructed hierarchically over previous methods. The extraction process proceeds in a top-down manner: informative top-level fragments are extracted first, and by a repeated application of the same feature extraction process the classification fragments are broken down successively into their own optimal components. The hierarchical decomposition terminates with atomic features that cannot be usefully decomposed into simpler features. The entire hierarchy, the different features and sub-features, and their optimal parameters, are learned during a training phase using training examples. Experimental comparisons show that these feature hierarchies are significantly more informative and better for classification compared with similar non-hierarchical features as well as previous methods for using feature hierarchies.

1. Introduction

The selection of effective image features is a crucial component of a successful classification scheme. A number of recent classification methods have used features composed of image patches, or fragments, selected from training images during a learning stage [1-5]. The success of these methods is mainly due to two reasons: first, they identify common object parts that characterize the different objects within the class, and second, the parts are combined in a manner that allows variations learned from training data. This notion is extended in the present work from the representation of objects to the representation of their constituent parts. Instead of representing a local part by a fixed image fragment, the part itself (such as an eye in face detection) is decomposed into its own optimal components (e.g. eyelid, eye corner, eye pupil, etc.), and the allowed variations in the configuration of the sub-parts are learned from the training data. The decomposition into sub-parts continues recursively and terminates at the level of 'atomic fragments', which cannot be broken down further without loss in mutual information. We describe in this paper an algorithm for obtaining informative feature hierarchies, and show that the resulting hierarchies are more informative and better for classification compared with holistic features. The input to the algorithm is a set of images belonging to the same object class and a set of non-class images. The output is a set of hierarchical features together with the learned parameters (combination weights, geometric relations) suitable for the recognition of novel instances of the learned class. Examples of the hierarchical features obtained by the algorithm are shown in Figures 1, 5.

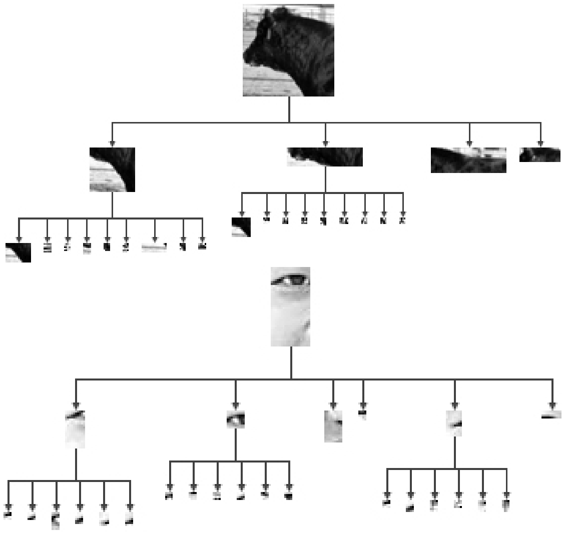

Figure 1: Examples of the hierarchies obtained by the algorithm.

Experimental evaluations show that the decomposition by our method increases the amount of information delivered by the fragments by a wide margin, improves the detection rate, and increases the tolerance for local distortions and illumination changes.

The rest of the paper is organized as follows. In the next section we briefly review previous relevant approaches to the problem of selecting and combining features for recognition. In Section 3 we describe the proposed algorithm. In Section 4 we show experimental results, comparing the use of hierarchical features with holistic features for object detection on several object classes. We conclude with general remarks on possible extensions and applicability of the method in Section 5.

2. Previous work

In several recent schemes for object detection and classification, the basic features used for classification are local image fragments, or patches, depicting significant object components, and selected from training images during a learning stage [1-5]. The features can be selected from a large pool of candidate fragments [1] or from a set of regions selected by interest operators [2, 3]. During the classification stage, the features are located in the image being classified, and then combined using a number of possible methods including a naïve-Bayesian combination [1], a probabilistic model combining appearance, shape and scale [3], the output of a classifier network [2] or an SVM-based classifier [4]. The features used by these methods were non-hierarchical, that is, they were not broken down into distinct simpler sub-parts, but detected directly by comparing the fragment to the image. Their similarity can be measured by different measures, including normalized cross-correlation, affine-invariant measures [6], and the SIFT measure [7].

A number of classification schemes have also used feature hierarchies rather than holistic features. Such schemes were often based on biological modeling, motivated by the structure of the primate visual system, which has been shown to use a hierarchy of features of increasing complexity, from simple local features in the primary visual cortex, to complex shapes and object views in higher cortical areas. In a number of these models, [8, 9], the architecture of the hierarchy (size, position and shape of features and their sub-features) is pre-defined rather than learned for different classification tasks. The learning of a particular class was obtained by the combination of weights from the upper level of the hierarchy. A hierarchical model trained by examples was studied in [10]. The study uses a network model in which both the combination weights and the convolution templates were learned from examples by back-propagation, whereas the number of hierarchy levels and positional tolerance were pre-defined. Previous comparisons [11] as well as our experiments (Section 4) show that the features used by these hierarchical models are not as informative and useful for classification as the classification features extracted by the methods reviewed above, and this accounts in part for limitations in their performance. In summary, classification features used in the past were either highly informative but non-hierarchical, or hierarchical features which were less informative and not as useful.

In the present work, we combine the advantages of learning informative classification fragments, with the learning of hierarchical structure with adaptive parameters. Informative object components are used for classification, but they are represented and detected using a hierarchy of simpler sub-parts. The next section describes the method of extracting the full hierarchy and its associated parameters.

3. The construction of the feature hierarchies

In this section, we describe the algorithm for obtaining the feature hierarchies. The algorithm proceeds along the following main stages. First, initial informative fragments are selected (Section 3.1). Second, the selected fragments are used to define new training sets for the selection of sub-features (3.2). These two steps are applied recursively until a level of 'atomic fragments' is reached. Third, parameters of the features hierarchy are optimized (3.3). Finally, the classification using the derived hierarchy is described in (3.4). We begin with a description of the initial selection of informative image fragments.

3.1. Selecting informative image fragments

We use a method for extracting good initial features similar to [11]. The process identifies fragments that deliver the maximal amount of information about the class. A large number (tens of thousands) of candidate fragments are extracted from the training images. We consider as initial candidates rectangular fragments of class images at multiple sizes and positions. We used fragments sizes ranging from 10% up to 50% of image size in each dimension, with scaling step of 1.2. For each fragment size, we examine fragments in positions placed on a regular grid with step equal to 1/3 of the size of a fragment. For every fragment, the optimal detection threshold is determined by maximizing the mutual information between the fragment and the class, as explained below. The normalized cross-correlation was used as similarity measure, but other measures, such as SIFT [7], can also be used. A binary variable is associated with every fragment in the following way:

$$f_i(I,\theta_i) = \begin{cases} 1, \ \mathit{if} \ \ S(I,f_i) > \theta_i \\ 0, \mathit{otherwise} \end{cases} \qquad (1)$$

Here *S(I, f_i)* is the maximal visual similarity between fragment f_i and image *I*, θ_i is the detection threshold associated with $f_{i.}$ A binary variable *C(I)* is used to represent the class, namely, *C(I)* = 1 if the image *I* belongs to the class being detected and 0 otherwise.

Candidate fragments are evaluated by the amount of mutual information [13] they deliver about the class. The advantages of selecting features by their MI are discussed in [14]. The mutual information between the two binary variables is defined as:

$$MI(f_i(\theta_i);C) \equiv \sum_{\substack{f_i(\theta_i)=\{0,1\} \\ C=\{0,1\}}} p(f_i(\theta_i),C)\log(\frac{p(f_i(\theta_i),C)}{p(f_i(\theta_i))p(C)}) \qquad (2)$$

The mutual information is a function of the detection threshold θ_i. If the threshold is too low, the information delivered by the fragment about the class will be low, because the fragment will be detected with high frequency in both class and non-class images. A high threshold will also yield low mutual information, since the fragment will be seldom detected in both class and non-class images. At some intermediate value of threshold, the mutual information reaches a maximum. The detection threshold for each fragment is selected to maximize the information $MI(f_i;C)$ between the fragment and the class. From the initial pool of candidate fragments, the most informative fragments are selected successively. After finding the fragment with the highest mutual information score, the search identifies the next fragment that delivers the maximal amount of additional information with respect to previously selected fragments. At iteration i the fragment f_i was selected to increase the mutual information of the fragment set by maximizing the minimal addition in mutual information with respect to each of the first i-1 fragments.

$$f_i = \arg\max_{f_k \in K_i} \min_{f_j \in S_i}(MI(f_k, f_j;C) - MI(f_j;C)) \qquad (3)$$

Here K_i is the set of candidate fragments, S_i is the set of selected fragments up to iteration i, f_i is the fragment to be selected at iteration i. The min is taken over all previously selected f_j, to avoid redundancy: if f_k is similar to one of the selected fragments, this minimum will be small. The max stage then finds the candidate in the pool with the largest additional contribution. In empirical testing, this algorithm was shown to select highly effective classification features [12].

We drop the dependence on thresholds θ since they are already set to the optimal value for each fragment separately. The update rule for the fragment sets is:

$$K_{i+1} = K_i \setminus \{f\} \qquad (4)$$
$$S_{i+1} = S_i \cup \{f\}$$

The initial K_0 is the set of all candidate fragments; S_0 is the set containing the fragment with highest mutual information with the class. The iterations end when the increment in mutual information gained by a new feature is less than some small threshold ε (0.08) or until the number of selected fragments has reached a pre-selected limit.

For each fragment the extraction process determines an allowed region, or Region of Interest (ROI) within which the fragment is searched. The size of the ROI is also set by an information maximization process described further in Section 3.3.

3.2. Selecting optimal sub-fragments

The top-level classification features described above appear often in the images containing object to be detected and seldom in non-class images. In a similar manner, useful sub-features should appear often in the regions containing 'parent' feature and seldom elsewhere. To identify such sub-features, we construct for each fragment f a set of positive examples, which are image regions containing the fragment f, and negative examples, where detection of it should be avoided as much as possible. The negative examples are selected from non-class images that give "false alarms", and therefore lie close to the class/non-class boundary. The positive examples for the fragment f were provided by identifying all the locations in the class images where the fragment f was detected. This set was then increased, since the goal of the fragment decomposition is to successfully detect additional examples that were not captured by the fragment f alone. For this end, the positive set was increased by lowering the detection threshold of the fragment f, yielding examples where f is either detected or almost detected. The reduced threshold was determined to increase the positive set by 20%. This amount of increase was chosen to add a significant number of almost-detected examples, and avoid examples that are dissimilar to f. A set of negative examples was similarly derived from the non-class images. Figure 2 shows the example of an informative fragment together with positive and negative examples of this fragment extracted from the training data.

For the extraction of the sub-fragments of a feature on lower level of the hierarchy, the same procedure of obtaining positive and negative examples is used. Positive examples come from regions in class images where the parent feature was detected or almost detected within its ROI, and negative examples come from regions in the non-class images where the feature was detected. In this case, the feature position in the training images was determined by the computation of optimal positions of all the hierarchy nodes together (Section 3.4), so that at most one example was taken from each training image.

Once the positive and negative sets of examples are established, sub-fragments are selected by exactly the same information maximization procedure used at the first level. The candidate sub-fragments in this case are the sub-images with their center point within the parent fragment, and having area not greater than ¼ of the parent's area. Sub-features are added to the tree until the additional information was lower than a threshold (0.08) or their number reached a pre-defined maximum (10 fragments). Experimentally, fragments with smaller contributions did not improve significantly the detection of the parent feature. If the decomposition of *f* into simpler features increased the delivered information, the same decomposition was also applied to *f*'s sub-features. Each of the sub-fragments was considered in turn a parent fragment, positive and negative examples were found and the set of its informative sub-fragments was selected. Otherwise, decomposition was terminated, with *f* considered an atomic fragment. Atomic fragments were usually simple, typically containing edges, corners or lines. Hierarchy examples are shown in Figures 1, 5.

As explained in Section 3.4, during the classification stage, only the atomic features are directly correlated with the input image, and their responses are combined using weights learned at the training stage.

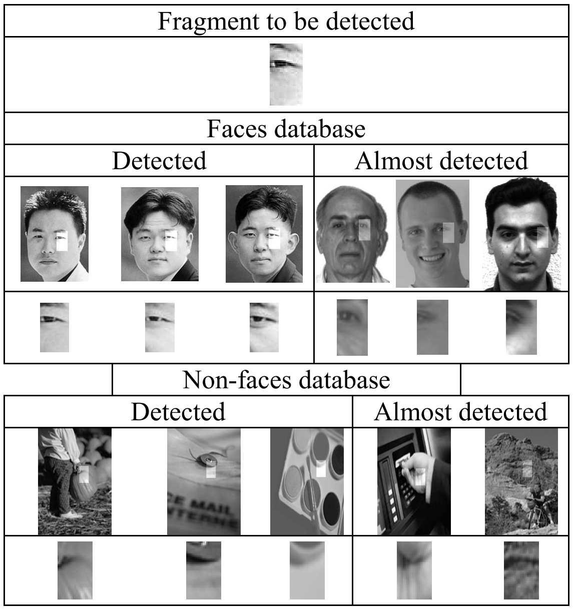

Figure 2: The positive and negative examples for fragment detection. The original fragment *f* is shown on top. Row 1: class image examples where this fragment was detected (left) or almost detected (right). Row 2: the positive examples. Row 3: Non-class images where the fragment was detected (left) or almost detected (right). Row 4: Negative examples.

3.3. Optimizing the regions of interest

For each fragment, a region of positional tolerance is extracted, called the feature's region of interest (ROI). The ROI defines the area in novel images where the fragment is searched for. The locations of the ROIs of sub-fragments in every image are determined relative to the detected position of their parent fragment. The amount of information a fragment delivers about the class depends on the size of its ROI. When the ROI is too small, the information is low, because in many class images the fragment will fall outside the ROI and therefore will not be detected. If the size of ROI is too large, the number of false detections will increase. At some intermediate size of the ROI, the mutual information reaches a maximum (Figure 3). The size of ROI for a fragment f was therefore chosen to maximize the mutual information *MI(f;C)*. For first-level fragments, the optimization process evaluated different candidate ROI sizes from zero to half the size of the search window, and found the size that brought the MI to the maximum. The search window is a fixed region within the input image, where the algorithm looks for the entire object. This window was set in our experiments to size 200x200 pixels. To detect an object within a larger image, the search window can either scan the image or move only to selected salient locations [16]. The locations of the ROIs of first-level fragments were defined relative to the center of the search window.

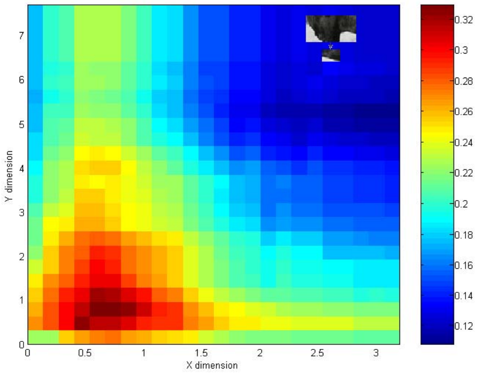

Figure 3: Plot of MI of a sub-fragment as a function of its ROI size, the maximum is selected as ROI.

During the hierarchy construction, the initial ROI size of a sub-fragment was set to be equal to the size of its parent. After the hierarchy was completed, additional optimization of the ROI sizes was performed in a top-down manner: first, the ROI of the uppermost node was optimized to maximize the mutual information between the class variable and hierarchy's detection variable, while all other ROIs were fixed. A similar process was then applied to its children, and the optimization proceeded down the hierarchy, where at each stage the ROIs of the higher levels are kept fixed.

Another set of hierarchy parameters is the combination weights of the sub-features responses. The optimization of the combination weights is described in Section 3.4 below together with the use of these weights in the classification process.

3.4. Classification by feature hierarchies

Performance of the hierarchical features was evaluated using a network model similar to HMAX [9], with layers performing max and weighted sum operations. For a given feature, the maximal response of each sub-feature is taken over the sub-feature's ROI, and then the responses of all sub-features are combined linearly:

$$r = w_0 + \sum_{i=1}^{n} w_i s_i \tag{5}$$

where r is the combined response, s_i the maximal response of sub-feature i within its ROI, w_i are the weights of the combination, and n the number of sub-features. For the atomic sub-features, the response was equal to the maximal normalized cross-correlation between the sub-feature and the image within the ROI. The final response s_p of the parent feature was obtained by a sigmoid function,

$$s_p = \frac{2}{1 + e^{-r}} - 1 \tag{6}$$

which normalizes s_p to the range [-1,1].

The response of the topmost node of the hierarchy, corresponding to the entire object, is then compared to 0. Positive response means that object is detected. The amount of information about the class carried by the hierarchy is defined as the mutual information between the class variable C and the hierarchy detection variable H, equal 1 when the response of the topmost node is positive and zero otherwise.

The combination weights were adjusted during training using iterative optimization that alternates between optimizing positions and weights, as described below. First, the weights are initialized randomly in the range [0..1]. The scheme then alternates between the following two steps.

Positions Step: fix weights, optimize feature positions. For every position of the parent fragment within its ROI the positions of sub-fragments (within their relative ROIs) that maximize the responses of the sub-fragments were found. Then, the position of the parent fragment that maximizes its response s_p is chosen. This routine can be implemented efficiently using Dynamic Programming.

Weights Step: fix feature positions, optimize weights. The combination weights of the features were optimized using the standard Back-Propagation algorithm with batch training protocol. The algorithm ends when no feature changes its position at Positions Step.

This weight selection procedure can be shown to converge to a local minimum of classification error. Experimentally, we found that the algorithm converged in less than 10 iterations. The obtained optimum is stable, starting from multiple random initial weights we end up with similar performance.

3.5. Summary of the hierarchy construction algorithm

The full process of constructing hierarchical features can be summarized by the following steps.

INPUT: A set P of class images. A set N of non-class images.

OUTPUT: A feature hierarchy H – a tree with nodes corresponding to parts and sub-parts of the object being recognized together with a set of associated parameters: ROI of every node and combination weights.

HIERARCHY CONSTRUCTION ALGORITHM:

1. Initialize H as a tree containing a single node (root) f^0, corresponding to the entire object.

2. Using the original training sets (P and N), extract a set $S(f^0)$ of first-level informative fragments as described in Section 3.1. Add the fragments from $S(f^0)$ as children to f^0. Evaluate the mutual information $MI(H;C)$ as described in Section 3.4.

3. For each leaf fragment f determine the sets $P(f)$ and $N(f)$ of positive and negative examples as in Section 3.2.

4. Find the set $S(f)$ of the most informative sub-fragments of f.

5. Add the fragments from $S(f)$ as children of f and re-evaluate $MI(H;C)$. If it does not increase compared with the case without $S(f)$ – remove $S(f)$, mark the leaf node f as 'atomic' fragment. Otherwise leave $S(f)$ in H.

6. Repeat steps 3 – 5 until all leaf fragments are marked as 'atomic'.

7. Optimize the ROI sizes of the hierarchy nodes, as in Section 3.3.

The classification stage using the hierarchy of features H can be summarized as follows:

INPUT: A novel image I. A feature hierarchy H, extracted from examples.

OUTPUT: A binary decision variable S (1 if the object was found in I, and 0 otherwise).

CLASSIFICATION ALGORITHM:

1. Compute the correlations of all the leaf nodes of H with the image I. Call the 2D arrays (of size equal to the size of I) containing the correlation values the *response maps.*

2. For every node of H whose children's response maps have been computed, compute its own response map: for each position of the feature within the image I, find maximal responses of its children within their ROIs and combine their responses using equations (5) and (6). Store the response of the node in its own response map.

3. Repeat Step 2 from the bottom nodes to their parents until the top of the hierarchy is reached.

4. Using the response map of the topmost node, find the maximal response within its ROI (eq. 6) and compare it to 0. If the response is greater than 0, set S to 1, otherwise set it to 0.

4. Experiments

Hierarchical features were extracted and compared with holistic (non-hierarchical) ones using their mutual information, and by comparing classification performance. The mutual information measures the quality of the features directly in a manner that does not depend on the particular classifier [12, 14].

The information carried by the hierarchical and holistic features was compared using 3 object classes: faces (200 faces, 500 non-faces in the training set, 800 faces, 1500 non-faces in the test set), cows (100 cows, 500 non-cows in the training set, 220 cows, 2500 non-cows in the test set) and airplanes (320 airplanes, 500 non-airplanes in the training set, 750 airplanes, 2500 non-airplanes in the test set).

For each object class, the most informative holistic feature was first determined using the algorithm described in Section 3.1. For comparison, a hierarchy of sub-features was extracted from this feature. In computing the ROC curves of a feature [15], the hits and false alarms were defined by using the hierarchy as a single feature classifier. That is, test images were classified based on the feature in question; hits corresponded to class image identified correctly, false alarms to non-class images identified incorrectly. The experiment was repeated 50 times for each class, the image database each time split randomly into training and test set. Overall, 150 top-level fragments were extracted, and for each one a hierarchy was constructed using the algorithm above. The information supplied by the first-level hierarchical features increased in the test set for all fragments (n=150, 3 classes) by a large amount compared with the corresponding holistic features (average increase 46.6%, s.d. 30.5%, $p < 10^{-9}$ one-tailed paired t-test). The holistic and hierarchical features were also compared using their complete ROC curves, showing a significant advantage of the hierarchical detection over the entire range, (0-90% false alarm, n=150, p<0.000001).

Further decomposition into a multi-level hierarchy provided additional significant gain in information (n=97 features, average increase 10.0%, s.d. 10.7% $p < 10^{-9}$ one-tailed paired t-test). The ROC detection curves also improved significantly. Results of the comparisons are shown in Figure 4a,b. Figure 4a shows the comparison of ROC curves obtained by a holistic feature (blue), the same feature decomposed into a single level of sub-features (magenta), full hierarchy (red), and a decomposition using fixed spacing and sub-fragment sizes (black). Figure 4b shows the mean difference between the ROC curves of classifier based on a single holistic feature and its hierarchical decomposition (averaged over 50 runs).

The results show that hierarchical features are significantly more informative and lead to much better classification results compared with holistic features. Significant improvement is obtained already with a single additional level.

We found in comparisons that optimizing the size and locations of the sub-fragments relative to their parent fragments add significantly to the MI compared with a hierarchy that uses fixed (and optimized) sizes for the sub-fragments and spacing between them. If the sub-fragments' centers were arranged on a uniform grid, rather than selecting their optimal locations during training, the MI decreases (average 43% s.d. = 35% $p < 10^{-10}$ paired t-test), and the detection performance of the units decreases. The fixed spacing was set to the average spacing obtained at each level by the adaptive scheme. Optimizing ROI size also adds significantly to the MI compared with a fixed ROI size that was optimized for each level separately (average 8.1% s.d. 13.7% p < 0.0055).

The performances of classifiers based on multiple holistic features and multiple hierarchies were also compared. The comparison was performed on 3 object classes: airplanes (same as above), horses (160 horses and 500 non-horses in the training set, 160 horses and 2500 non-horses in the test set) and side views of cars (160 cars and 500 non-cars in the training set, 160 cars and 2500 non-cars in the test set). First, to determine the number of fragments required for full classifier, the Equal Error Probabilities (EEP) were computed for classifiers based on 1 to 50 fragments. The classifier performance asymptoted at 30-40 fragments (Fig. 4c). Next, the performances of full classifiers using 50 holistic features and 50 hierarchical features were compared (Fig. 4d). The comparison clearly shows the advantage of hierarchical features.

The informative sub-fragments selected by our method were also compared with an alternative method in which informative sub-features are obtained by directly maximizing the mutual information using gradient ascent with simulated annealing. For this experiment, a set of 20 informative parent fragments of size 40x40 pixels each was computed from two object classes (faces, horses). For each fragment, a set of positive and negative examples was determined as described in Section 3.2. Two sub-features were compared: the most informative sub-fragment of size 20x20 pixels, computed using the method described in Section 3.2 and the sub-feature of the same size computed using gradient ascent with simulated annealing. In this case, the sub-fragment starts either from a uniform or a random grey-level image. These grey levels are then modified by a gradient ascent computation that used MI as the optimization measure. For each parent fragment, the computation was performed 10 times, the image database each time split randomly into training and test set. The comparison shows that the image sub-fragments are significantly more informative than features learned by the gradient ascent procedure (average MI increase 36% s.d 31%). Selecting sub-features from the training images thus leads to better features than synthesizing new ones by gradient ascent. The likely reason is that the search in the space of all possible sub-features has multiple local maxima which are significantly lower than the optimal sub-features.

5. Discussion

We presented a scheme for extracting feature hierarchies for classification. The top-level features are informative image fragments, which are then broken down successively into informative sub-fragments. The extraction is automatic, including the selection of the sub-features as well as their combination weights and ROIs. The hierarchy outperforms the single-level features by a wide margin, both in the amount of delivered information and recognition performance. The amount of positional tolerance of sub-features and their positions should be learned from examples, since using sub-parts of uniform size and spacing degrades performance.

Classification using the feature hierarchy was implemented in the current work using a simple feed-forward combination scheme, with weights extracted during learning. We also tried a combination scheme using a Bayesian network structure with bi-directional computation. An advantage of the modified scheme, which will be discussed in more detail in future work, is that it uses the entire hierarchy to recognize not only complete objects, but also object parts. In this manner, feature hierarchies can be used to improve the performance of recognition and classification schemes, and also to extend them to provide a fuller description of the objects together with their parts and sub-parts at different levels.

Acknowledgements

This work was supported by IMOS Grant 3-992 and conducted at the Moross Laboratory for Vision and Motor Control.

References

[1] E. Sali, S. Ullman, “Combining class-specific fragments for object classification”. In *Proc. 10th British Machine Vision Conference*, volume 1, 203 – 213, 1999

[2] S. Agarwal, A. Awan, D. Roth, “Learning to detect objects in images via a sparse, part-based representation”. *IEEE TPAMI*, 26(11):1475-1490, 2004.

[3] R. Fergus, P. Perona, A. Zisserman, “Object Class Recognition by Unsupervised Scale-Invariant Learning”. *Proc. of the IEEE Conf. Comp. Vis. Pattern Recog.* 264-271, 2003.

[4] B. Heisele, T. Serre, M. Pontil, T. Vetter and T. Poggio, “Categorization by learning and combining object parts”, *Neural Information Processing Systems*, 2001.

[5] B. Leibe and B. Schiele, “Interleaved object categorization and Segmentation”, *Proceedings of British Machine Vision Conference (BMVC'03)*, 2003.

[6] K. Mikolajczyk and C. Schmid, “Scale & affine invariant point detectors”, *Int. J. Comp. Vis.*, 60(1), pp. 63-86. 2004.

[7] D. Lowe, “Distinctive image features from scale-invariant keypoints”, *Int. J. Comp. Vis. 60(2),* 91-100, 2004.

[8] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. *Biol. Cyber. 36,* 193-202 1980

[9] M. Riesenhuber, T. Poggio, “Hierarchical models of object recognition in cortex”, *Nat. Neurosci. 2(11*), 1019–1025, 1999.

[10] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, “Backpropagation applied to handwritten zip code recognition”, *Neural Comp. 1 (4),* 541-551, 1989.

[11] S. Ullman, M. Vidal-Naquet, E. Sali, “Visual features of intermediate complexity and their use in classification”, *Nat. Neurosci.* 5(7), 1-6, 2002.

[12] F. Fleuret, “Fast Binary Feature Selection with Conditional Mutual Information“, *Journal of Machine Learning Research*, vol. 5, pp.1531–1555, 2004.

[13] T. Cover, J. Thomas, “Elements of information theory”, Wiley, NY, 1991.

[14] N. Vasconcelos, M.Vasconcelos, “Scalable Discriminant Feature Selection for Image Retrieval and Recognition”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[15] D. Green, J. Swets, “*Signal Detection Theory and Psychophysics”,* Wiley, NY, 1966.

[16] L. Itti, Ch. Kosh, E. Niebur. “A model of saliency-based visual attention for rapid scene analysis”, *IEEE TPAMI*, vol. 20, No. 11, pp. 1254-1259, 1998.

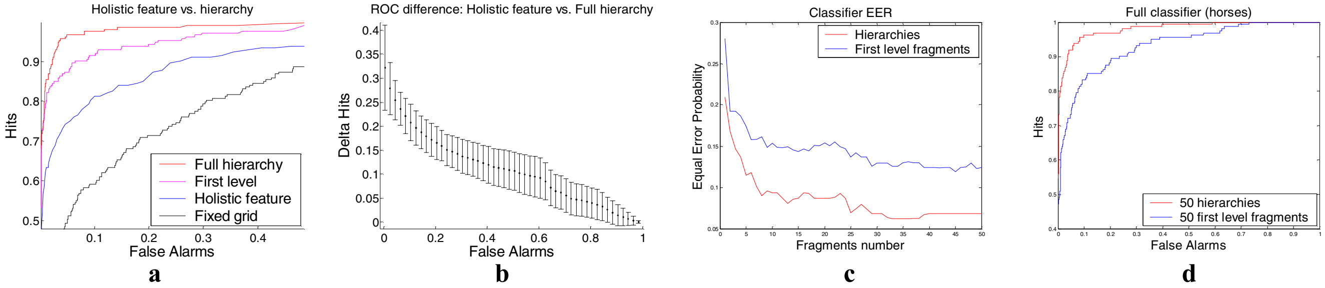

Figure 4: **a.** Example of ROC curves obtained by a holistic feature and the same feature represented hierarchically (see text). **b.** Average ROC difference (additional hits) and s.d. of one holistic feature vs. the same feature represented hierarchically for one class (cows, 50 runs). **c.** Equal Error Probability of classifiers based on hierarchical (red) and holistic (blue) features. **d**. Example of ROC curve of full classifier based on hierarchical (red) and holistic (blue) features.

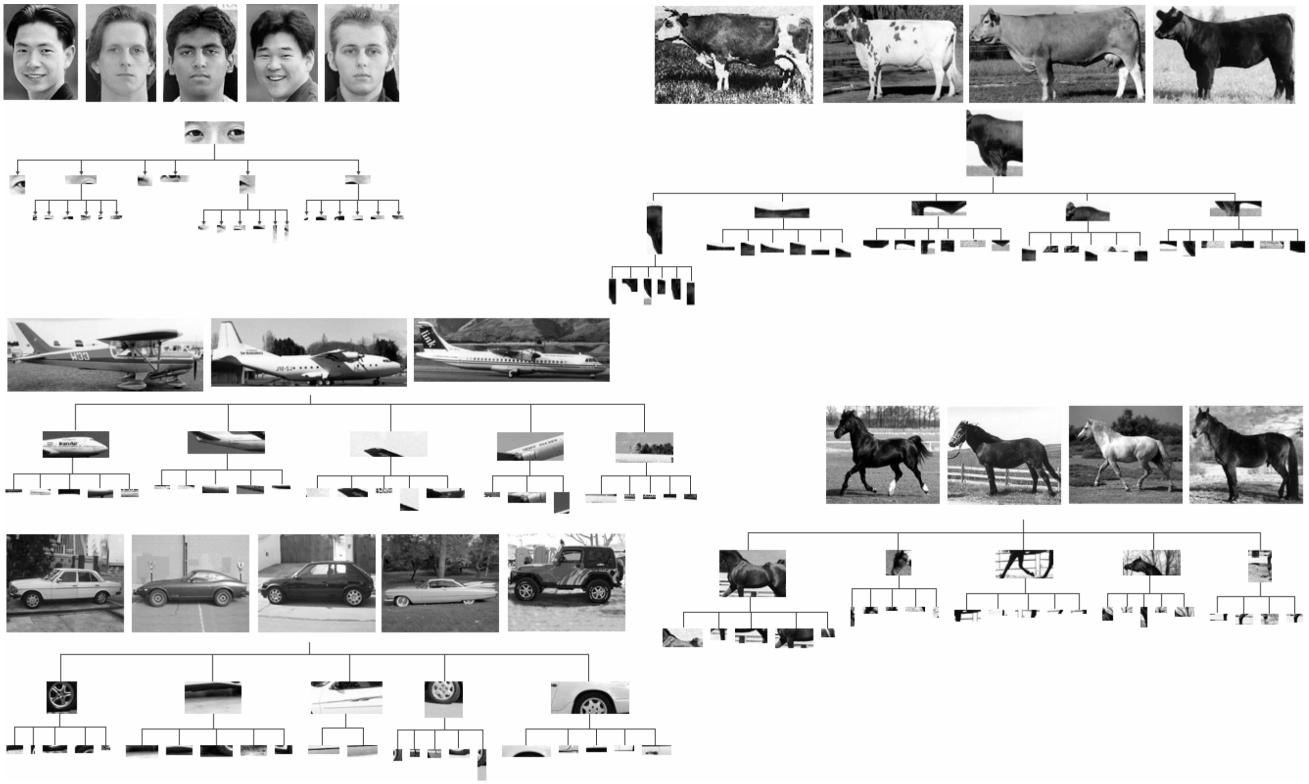

Figure 5: Examples of class images and computed hierarchies from 5 classes.