Structure Is a Visual Class Invariant

Bai Xiao, Yi-Zhe Song, Anupriya Balika, and Peter M. Hall

Computer Science Department, University of Bath, U.K.

Abstract. The problem of learning the class identity of visual objects has received considerable attention recently. With rare exception, all of the work to date assumes low variation in appearance, which limits them to a single depictive style usually photographic. The same object depicted in other styles — as a drawing, perhaps — cannot be identified reliably. Yet humans are able to name the object no matter how it is depicted, and even recognise a real object having previously seen only a drawing.

This paper describes a classifier which is unique in being able to learn class identity no matter how the class instances are depicted. The key to this is our proposition that topological structure is a class invariant. Practically, we depend on spectral graph analysis of a hierarchical description of an image to construct a feature vector of fixed dimension. Hence structure is transformed to a feature vector, which can be classified using standard methods. We demonstrate the classifier on several diverse classes.

1 Introduction and Background

The work in this paper was motivated by the desire to classify objects no matter how they are depicted. For example, it is perfectly normal to label a picture of a bird as "bird", whether it is seen for real, is photographed, is a drawing, or is depicted in any other way. Moreover, it is common human experience to be able to recognise a real instance of some object after being shown only a drawing of it. For example, flori — catalogues of plants — allow botanists to identify plants; flori support text descriptions using drawings rather than photographs, possibly because drawings capture the class as a whole more faithfully than a single photograph of one plant. Clearly, learning to identify object categories through pictures of all kinds is an important problem. It is this problem we address, in contrast to most previous work which assumes a single depictive style, most commonly photographs.

Consider a thought experiment using four images. Three images show faces, one little more than a scribble by a young child, one the face of a clown with crosses for eyes, the other a photograph. The fourth image is a photograph of a car. Which is the odd one out? We have run this experiment in practice, and were not at all surprised to find the car was always selected as the odd one out. Yet the variance between the faces is profound. Eyes can be any shape, so can the mouth. The draw face may or may not have an outline, and if it does exist can be any shape. Children tend to draw the eyes at the top of the head when

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 329–338, 2008.

in fact they are in the middle. Given so much variation in shape, in position, in color, it is difficult to see any invariant except the structural arrangement of facial parts. The natural question following these observations is *what class property is invariant across depictions?* Our answer, which we believe is novel, is stated as a proposition: *the topology of an object's parts is a class property invariant to depiction.* It is this proposition that this paper sets out to test.

We learn objects classes, and so our work relates to is preceded by a considerable body of research that also learns object classes. Recent work, *circa* 2000 to date, uses localized image patches; there is too much to review here, but [2,8,14,13,19] provide typical examples. Many problems have been addressed, including recognition under occlusion [5] and from different points of view, eg [16], and learning from just a few examples [11]. Despite the many successes of this work it is all premised on the assumption that corresponding patches exhibit low variation.However, drawings and photographs would raise the variance above acceptable limits. Indeed, some deliberately filter non-photographs from their database used to build models of appearance [7]. Yet an object's class does not depend on its depiction; faces can be drawn, photographed, painted and so on.

By comparison, there is little research on classifying objects independently of depiction type. Some use curves [15], which does not necessarily assume photographic depiction, but does require low shape variance which we have observed cannot be assumed when using artwork. Schectman and Irani [18] show it is possible to match pairs of images in different depictive styles. They use the spatial relation between different patterns, which they call local self-similarities. For example, flowers might represent eyes in one picture, but photographed in another. Provided matching is consistent, Schectman and Irani [18] can match them. It might seem that matching many pairs would allow classes to be built, but as the number of representational instances grow so does the variation in depective style. This means there is no satisfactory charactisation in terms of appearance alone.

Fidler and Leonardis [6] uses a large data base of image to learn very simple primitives, which are combined into ever more specific objects until the highest layers are category-specific. The system is then able to detect categories across depictive styles. In contrast we need just a few examples of each class in order to learn. Also, Fidler and Leonardis [6] do not explicitly consider the topological relationship, as do we and Ahuja and Todorovic [1]. But they [1] cluster parts on the basis of similar geometric, photometric and topological properties. For them, topological similarity is defined via tree isomorphisms.

By using spectral graph theory [4] to encode structure as vectors we not only avoid the graph isomorphism problem and gain robustness to noise, but also can form clusters. In practice we must extract an object's structure with a method robust to wide variations in depictive style. The outline of the paper is as follows. In Section 2, we introduce a hierarchical image description by building a tree description through agglomerative merging of image primitives. Section 3 explains how a a modified form of graph energy is used to segment the tree into sub-trees, yielding a rough segmentation of objects. The objects are subject to exactly the same form of analysis, yielding their parts. The topological relation

between parts is then characterized with a feature vector. These features are clustered using standard methods, in Section 4. Section 5 shows the experiments. Finally, we give the conclusion in Section 6.

2 A Hierarchical Image Description

Forming a hierarchical image description is important to us, but our claim should not depend on a particular algorithm. We can use any algorithm to produce an image description, provided that: (i) gives a covering or partition of the image, so the whole is described; and (ii) operates effectively over images of all depictive styles, so any image can be described. Unfortunately, for reasons given below, these criteria, but especially the second, rule out well known approaches. Unfortunately, we were therefore forced to develop an approach of our own. Although this exhibits some novelty, we would not wish the reader to be distracted by it. We emphasis that we make just two claims for it: (a) it operates over different dipictions; (b) it is strong enough to allow us to demonstrate our proposition that structure is class invariant.

Maximally stable extremal regions [12] and sieves [3] both naturally produce a hierarchical image description. But neither can be used because they build no tree for binary images — but line drawings are often binary images. The patch based approaches common to many visual object classifiers (see Introduction, above) depend on low variation in appearance, but using line drawings, paintings, photographs ensures wide appearance variation. Additionally patches are usually filtered to keep just the most "salient". But there is a problem with databases which include drawings and photographs, as shown in the first column of Figure 1. In the second column of Figure 1 potential patches are located via the first extremum of a difference of Gaussian (DoG) filter, which is reported to give excellent results [10]. The size of a DoG patch is set to just cover a local feature. Figure 1 filters DoG patches using a threshold on the DoG signal strength. This threshold is not set absolutely, but is set as a fraction of the strongest signal. This must be set high to include on salient feature in a photograph, but the same threshold leads to missed features in the drawing – the second column in Figure 1. Lowering the threshold includes the missing features in the drawing, but at the expense of including all patches in the photograph – the third column in Figure 1.

We initially solved this problem using watershed regions as image primitives, which we then grouped to form a hierarchical description. This paper, though, uses image primitives which are DoG patches. In this case we filter DoG patches on the basis of size, using a greedy algorithm. We keep the largest DoG patch and reject any other patch whose center falls inside it. We then keep the largest remaining patch, and simply iterate until all patches are kept or rejected. The result is a set of DoG patches, each one a circular patch. These patches can overlap, as seen in the fourth column of Figure 1. The next step is to merge neighboring image primitives to create a hierarchical tree description.

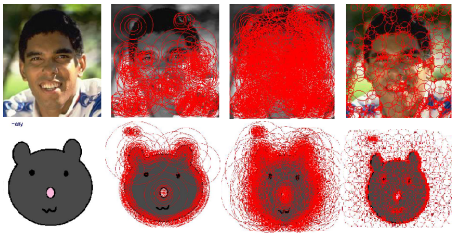

Fig. 1. Interest patches (oriented to local gradient): the first column are original images, the second column are top 50% of patches, the third column are top 90%. These patches can give poor coverage of some images, unless many are used. Notice that the eyes of the drawing are not covered until 90% of patches are used — but the photographed face is then swamped with patches. For comparison, the fourth column shows greedy filtering – the primitive we use.

At each merging iteration we merge the most similar pair of neighboring regions. The similarity measure must allow for the fact that regions can be of arbitrary shape, so we use simple statistical properties. Specifically, the distribution of color and intensity gradient of a region is modeled as a single Gaussian, written $\Omega = (N, \mu, \mathbf{C})$, where N is the number of pixel in the region, μ is the mean of the property vectors at each pixel, and $\mathbf{C}$ the covariance. When two regions, i and j, are merged it is easy to compute a new representation for them. If the regions do not overlap then the new model, $\Omega = \Omega_i + \Omega_i$, is

$$\Omega = \left(N_i + N_j, \frac{N_i \mu_i + N_j \mu_j}{N_i + N_j}, \frac{N_i \mathbf{C}_i + N_j \mathbf{C}_j}{N_i + N_j} + e \right) \tag{1}$$

The "error" term, e, compensates the eigenspace for the difference between the means of the model and is easily shown to be

$$e(i,j) = \frac{N_i N_j}{N_i + N_j} (\mu_i - \mu_j)^T (\mu_i - \mu_j) \tag{2}$$

We allow for spatially overlapping regions simply by partitioning the pixels into sets "i alone", "j alone" and "i and j", then adding the results as above. Agglomeration continues until there is just one region left. Finally note that the adjacency matrix is weighted using the error measure, so if primitives i and j are connected the weight is $w(i,j) = \exp(-e(i,j))/e_{\max}$, in which $e_{\max}$ is the maximum weight over all connected pairs. We are now ready to analyze this tree.

3 Object and Object Part Identification

Our aim now is to partition a hierarchical description of an image into objects, and then partition objects into parts. This process can not depend in any way on the visual representation, because that is assumed to be unknown. Rather,

the process must depend only on the structure of tree and the graph of primitives. Figure 2 exemplifies what we mean by object structure. This figure shows structure can be successfully obtained in a simple way and one same structure is general to many depictive styles. Appearance within a class varies significantly, because of depictive style, yet images from the same object class share roughly similar structures. For example, all the flowers contain a center; faces comprise eyes and a nose; for cows and horses we can extract the legs, the body and the head. Intuitively, these structures are a property of the object; which is as it should be, for we have proposed that structure is a property of an object's class.

Fig. 2. Examples objects and their parts, found using graph energy analysis, taken from our experimental training and test sets. Facial parts are identified; four legged animals have their head and legs separated from their body; flowers have their centers extracted.

Consider first the problem of identifying a salient object in the tree. As with graph cuts [17], we want high connectivity within an object and low connectivity between objects. But we do not use graph cuts, we use graph energy instead. We make this choice because: (i) graph cuts cannot decide the number of objects there are in an image, whereas our graph energy based method can decide this quantity; and (ii) the graph energy we use shows minima for graphs of homogeneous degree. This includes circuits (polygons) and cliques (fully connected objects), both of which are visually salient. Any node in the tree can be used to identify a collection of image primitives, which in turn correspond to a subgraph the graph of neighbors. The root node corresponds to the whole image, and we compute the graph energy of this. Then we can remove the root node to reveal the lowest branches, which form a covering of the image and which partition the adjacent matrix into subgraphs. The energy of each subgraph is computed and their sum is defined as the energy of the whole. We continue in this way, moving down through the tree, computing a graph energy at each step.

We use borrow the idea of the Laplacian graph energy [9]. The Laplacian matrix is defined as $L = D - A$, in which D is a degree matrix, and A an adjacency matrix. Laplacian graph energy has the following standard definition: for a general graph $G = (V, A)$, with arc weights $w(i, j)$ the Laplacian energy is

$$\mathcal{E}(G) = \sum_{i=1}^{|V|} \left| \lambda_i - 2\frac{m}{|V|} \right| \tag{3}$$

In which: the λ_i are eigenvalues of the Laplacian matrix; m is the sum of the arc weights over the whole graph, or is half the number of edges in an unweighted graph; $|V|$ is the number of nodes in graph. Note that $2m/|V|$ is just the average (weighted) degree of a node. Now, the Laplacian energy of a graph can rise or fall; our tests

show that this rise and fall is strongly correlated with the variance in the degree matrix D. This means local minima tend to occur when the graph is regular.

The standard definition of Laplacian graph energy makes no explicit account for a change in the number of connected components. We would like this to happen, because connected components are partitions of a general adjacency matrix. Now the adjacency matrix we have has just one connected component, it is constructed that way. But the tree nodes do partition our adjacency matrix into subgraphs that are connected only with arcs weaker in strength than any arc in the subgraph, so it is reasonable to consider each tree branch (object) as a connected component in an adjacency matrix, Figure 3 shows a typically energy curve, as a function of the height of the tree.

We compute graph energy $\mathcal{E}(G)$ for each distinct object. We combine these by defining the *normalized* Laplacian energy for a graph of N connected components over n nodes (image primitives/leaves)

$$\xi = \frac{n}{N} \sum_{i=1}^{N} \frac{\mathcal{E}(G_i)}{|V_i|} \tag{4}$$

The term $\mathcal{E}(G)/|V|$ is the average connection energy per node — this is rather like normalising spatial measures (such as image moments) by area. Since the partition energy, ξ, is a function of tree height, so we should write $\xi(k)$. This novel definition coincides with $\mathcal{E}(G)$ exactly if there is just one connected component, but otherwise is bounded above by $\mathcal{E}(G)$. The significant difference, though, is that ξ rises steeply when two large connected components are merged, less steeply when a large component absorbs a smaller one, and gives local minima where connected components are close to regular graphs.

As mentioned local minima of graph energy occur when (Sub)graphs exhibit homogeneous degree. Figure 3 shows that there is more than one such local minimum. We currently select the tallest tree minimum to obtain a partition

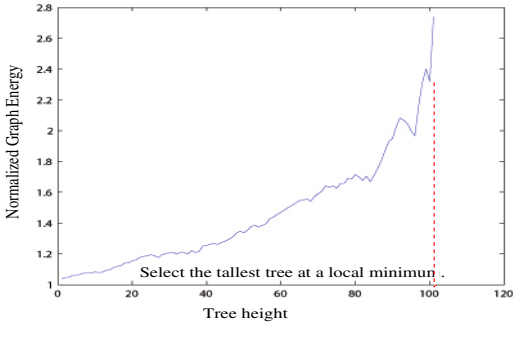

Fig. 3. An example of normalized graph energy as a function of tree height. We identify an object using the local minimum with tallest tree.

with the fewest number of large objects. We differentiate between foreground and background objects by having the user touch the foreground.

Having identified objects we turn attention to the second goal of breaking them into parts. Objects are just images, with a tree and neighbor matrix inherited from the full image. We find an object's parts by recursive application of the technique described — we just find the last local minimum in the graph energy curve for the object-image. The result is a set of objects and their parts, as seen in Figure 2. It is these structures we use to characterize the objects, as explained next.

4 Structure Vectors as Object Features, and Classification

In this section, we explain how objects are assigned feature vectors that are descriptors of their structure, and how these can be used to classify. The analysis of the tree yields an object made of parts, and the way those parts are structured. This is no more than a graph of nodes — the parts — and arcs — their connections. We follow Ping and Wilson [20], who advocate the use of eigenvalues from the signless Laplacian matrix. This is defined as $L = D + A$, where as before A is the adjacency matrix and D is the degree matrix. So, we form a feature vector, $\mathbf{f}$, as the eigenvalues of $D + A$.

We fixed $\mathbf{f}$ into a space of eight dimensions, padding with zeros if necessary. We used eight dimensions because that was the largest graph we observed, so we were sure to retain all information. The clusters visualized in Figure 5 are in two dimensions, yet remain largely separable; how best to choose the dimension of structural feature space remains an open question.

A structure vector $\mathbf{f}$ is no more than a particular kind of feature vector, which allows us to use any standard classification algorithm. After visually examining a two dimensional scatter plot of twelve different kinds of object we decided to

Fig. 4. The set of training images. Each column is a class (but horses and cows are both four legged animals). Many depictive styles are included in each class.

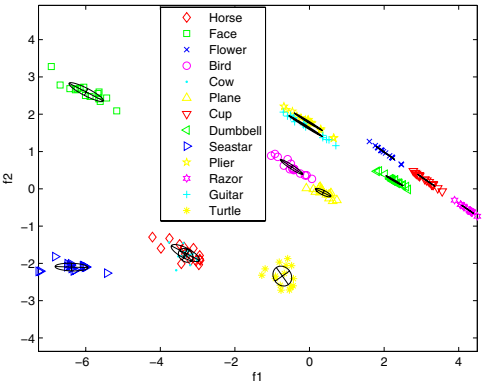

Fig. 5. Classes as components in a Gaussian Mixture Model

opt for supervised training. An eigenmodel comprising the number of point, their mean and covariance was fitted to the features of each training class, which was manually labeled hence the supervision. The individual eigenmodels were then collected into a single Gaussian Mixture Model (GMM). Figure 4 shows some of the pictures in our training set. We need only a few images per class, we trained with 10 examples per class. All sets contains drawings and paintings as well as photographs, yet form discernible clusters, as Figure 5 shows. The spread of the cluster is explained by noise that can arise from slightly different segmentations.

Notice we include both horses and cows in the training set, but these form a single cluster. This shows that the equivalence classes generated from structure are broad. We included these "four legged animals" because we predicted they would classify as one — but "four legged animals" is a perfectly legitimate class, and horses and cows can be separated other means, if the application demands. Notice that a system that always discriminates cows from horses might never be able to recognise the class of four legged animals.

5 Empirical Verification

To verify our classifier we used a set of test images, with 50 images from each class which are different from training images. Test images are highly varied in their depictive style(full test images are available at https://cs.bath.ac.uk/ xb202/data). For a given test object we constructed a feature vector, which was then input to our GMM which estimated the posterior probability of the test object belonging to each training class. The test object was assigned the class for which the posterior probability was largest. In this experiment, we constructed a histogram of how the test set distributes over the training classes, and each histogram is a row in Table 1. It is clear that most objects can be correctly classified with high probability, suggesting our GMM is a reasonable discriminative model.

Table 1. A confusion matrix; each row shows the probability a test class is classified as a given training class. The classes are: (1) Four legged animals; (2) Faces; (3) Flowers; (4) Birds; (5) Plane; (6) Cup; (7) Dumbbells; (8) Star fish; (9) Pliers; (10) Razors; (11) Guitars; (12) Turtles.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.94								0.03			0.03
2		0.94		0.06					0.03			0.03
3			0.86				0.14					
4		0.06		0.91		0.03						
5					0.86	0.14						
6			0.06			0.89					0.05	
7							1.00					
8	0.06							0.94				
9									0.97		0.03	
10						0.06				0.94		
11							0.09			0.03	0.88	
12	0.03											0.97

6 Discussions and Conclusions

This paper claims structure can be used to classify, just as other features like texture, shape are used. We have provided experimental evidence in support of this claim. Our means of forming hierarchical descriptions requires relatively simple images, but does operate over a wide range of depictive styles. Because of this we eschew any other claims regarding our method, except that is is strong enough to allow empirical support to our claim.

The characteristic that structure brings into classification is that of broad classes — sufficiently broad to cover objects depicted in a wide range of styles: photographs, drawings, and so on, yet sufficiently discriminative to be meaningful. The broadness of the classes we generate are illustrated not just by the inclusion of many depictive styles, but also by the fact cows and horses classify as one; this is not necessarily a disadvantage — an application might require "four legged animals" as a class, and subdivision on other measures is always possible.

Changing the segmentor and the classifier will not affect our conclusion, although it will make the system more robust. Similarly, there is more work to do to answer questions of scale, point of view, occlusion and so on. Again, answering these questions will not change our conclusion, but will act to make the system more robust, which provides amply future work. Also, comparative evaluation against other systems is for future work: exactly how to compare a classifier based on structure against one based on appearance is an open issue. Here we have shown that structure can be used as a feature, which was our intention and the contribution of this paper.

References

1. Ahuja, N., Todorovic, S.: Discovering hierarchical taxonomy of categories and shared subcategories in images. In: ICCV (2007)
2. Leibe, B., Schiele, B.: Interleaved object categorization and segmentation. In: BMVC
3. Andrew Bangham, J., Moravoc, K., Harvey, R., Fisher, M.: Scale-space trees and applications as filters, for stereo vision and image retrieval. In: Pridmore, T., Elliman, D. (eds.) British Machine Vision Conference (August 1999)
4. Chung, F.R.K.: Spectral graph theory. American Mathematical Society (1997)
5. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 1, 91–110 (2004)
6. Fidler, S., Leonardis, A.: Towards scalable representations of object categories: Learning a hierarchy of parts. In: ICCV (2007)
7. Schroff, F., Criminisi, A., Zisserman, A.: Havesting image databases from the web. In: ICCV (2007)
8. Grauman, K., Darrell, T.: Unsupervised learning of catagories from sets of partially matching image features. In: CVPR (2006)
9. Gutman, I., Zhou, B.: Laplacian energy of a graph. Linear Algebra and its Applications 44, 29–37 (2006)
10. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. TPAMI 2005 27(10), 1615–1630 (2005)
11. Fei-Fei, L., Fergus, R., Perona, P.: A bayesian approach to unsupervised one-shot learning. In: ICCV (2003)
12. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC (2002)
13. Mikolajczyk, K., Leibe, B., Schiele, B.: Multiple object class detection with a generative model. In: CVPR (2006)
14. Mutch, J., Lowe, D.: Multiclass object recognition with sparse, localized features. In: CVPR (2006)
15. Pinz, A., Opelt, A., Zisserman, A.: A boundary-fragment-model for object detection. In: ECCV (2006)
16. Yan, P., Khan, S.M., Shah, M.: 3d model based object detection in an arbitrary view. In: ICCV (2007)
17. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: International Conference on Computer Vision and Pattern Recognition (1997)
18. Schechtman, S., Irani, M.: Matching local self-similarities across images and videos. In: CVPR (2007)
19. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for visual object class recognition
20. Zhu, P., Wilson, R.C.: Stability of the eigenvalues of graphs. In: Gagalowicz, A., Philips, W. (eds.) CAIP 2005. LNCS, vol. 3691, pp. 371–378. Springer, Heidelberg (2005)