Modelling Visual Objects Invariant to Depictive Style

Qi Wu
http://www.cs.bath.ac.uk/~qw219
Peter Hall
http://www.cs.bath.ac.uk/~pmh

Media Technology Research Centre
Department of Computer Science
University of Bath,
Bath, UK

Abstract

Representing visual objects is an interesting open question of relevance to many important problems in Computer Vision such as classification and location. State of the art allows thousands of visual objects to be learned and recognised, under a wide range of variations including lighting changes, occlusion, point of view, and different object instances. Only a small fraction of the literature addresses the problem of variation in depictive style (photographs, drawings, paintings *etc*.), yet considering photographs and artwork on equal footing is philosophically appealing and of true practical significance.

This paper describes a model for visual object classes that is learnable and which is able to classify over a broad range of depictive styles. The model is a graph in which simple shapes label region nodes. We use our model to classify twenty classes in CalTech 256, each class augmented by additional images to increase the variance in style. When compared to a Bag of Words classifier and to a structure only based classifier, our results show a significant increase in robustness to variance in depictive style.

1 Introduction

Humans are able to recognise objects in a seemingly unlimited variety of depictions: in photographs, in line drawings, as cuddly toys, in clouds. Computer Vision classifiers, on the other hand, tend to be restricted to recognising objects in photographs alone. The work in this paper is motivated by a desire to explore this gap, and its broad contribution is to take one step towards closing it.

The fact that objects can be visualised in a wide variety of depictive styles, yet remain recognisable, leads us to the question: *what properties of an object class are invariant to depiction?* This is an important question for Computer Vision, because it directly affects performance in applications such as image retrieval, image matching, and object classification. With few exceptions, the models used in Computer Vision are trained and tested on a single depictive style. Yet models learned exclusively from photographs typically do not generalise well to other depictive styles; it can be said that such models are over-fitted. Such models are necessarily limited in their utility to applications – it becomes difficult to access both photographs and artwork in a library of portraits, for example. Additionally, recognisable objects exist for which there are no photographs (*e.g*. the Gryphon).

We argue that models of visual objects should not be premised, even tacitly, on photo-real appearance or indeed on any particular depictive style at all. Rather, visual object models should be based on quasi-invariant properties of the objects in a class. A similar argument is made by those who advocate part-based representations for image. We go further by saying that such models should generalise across depictive styles. This means that if a model is constructed using images in one style, the same object should also be classifiable even when depicted using a different style.

In this paper, we investigate a method for modelling visual objects classes in a manner that is invariant to depictive style. The assumption we make is that an object class is characterised by the qualitative shape of object parts and their structural arrangement. Hence we use a graph of nodes and arcs in which qualitative shapes such as triangle, square, and circle to label the nodes. More exactly our model is a hierarchy of levels, yielding a coarse-to-fine representation. Each level contains an undirected graph of nodes and arcs. Nodes between levels are connected via parent-child arcs, which are directed. Child nodes are nested inside their parent.

Our technical contribution is to show that *it is possible to learn models of object classes that generalise across depictive styles, in the sense that it is possible to learn a model using one style but classify objects depicted in other styles*. The paper has two main sections:

1. Section 3 explains how to build a hierarchical graph model to represent object classes, with nodes labelled by qualitative shape and edges labelled with displacement vectors.
2. Section 4 describes experiments on a cross-depiction image dataset. The experiments provide empirical evidence that our model is more robust to cross-depiction object classification than an excellent Bag of Words classifier.

We briefly review Related Work in Section 2. The paper concludes, in Section 5, with a discussion of the limitations of our modelling scheme, and points to future developments and applications.

2 Related Work

Of the many approaches to visual object classification, the bag of words (BoW) family is arguably the most popular and successful. Borrowing from techniques in document analysis, BoW methods have featured in Computer Vision since the early to mid 2000s. The classical one for image categorization, used by this paper for comparison, is an extended version of the method proposed first in [5]. Details aside, all in the BoW family model visual object classes via histograms of "visual words". Using a histogram ignores the spatial location of the words, making BoW methods robust to changes in shape, occlusion, lighting, *etc*.

Although the BoW methods address many difficult issues, they tend to generalise poorly across depictive styles. This means that models trained on photographs will tend to misclassify objects in another depictive style. The explanation for this is the formation of visual words: words are found by clustering low-level features, hence the assumption of low variation in feature appearance is built-in to such classifiers. Others have acknowledged this, and respond by using low-level features that do not depend on photometric appearance. Some use the shape of edgelets [7, 19], others use the shape of regions [11, 12]. We do not use edge data, but do use region shape. However, rather than using complicated shapes for regions (as others do), or just using (a hierarchy of) Gaussian blobs [18], we use a collection of simple shapes (*e.g*. circle, square, triangle) [25]. The idea is that abstracting region shape into one of a few classes brings greater robustness to non-salient variations. Anecdotal support for

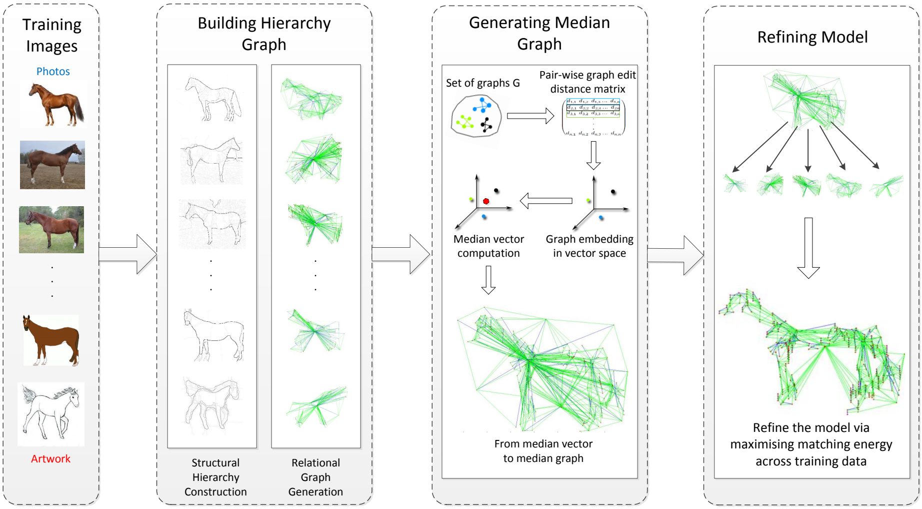

Training Images
Photos
Artwork
Building Hierarchy Graph
Structural Hierarchy Construction
Relational Graph Generation
Generating Median Graph
Set of graphs G
Pair-wise graph edit distance matrix
Median vector computation
Graph embedding in vector space
From median vector to median graph
Refining Model
Refine the model via maximising matching energy across training data

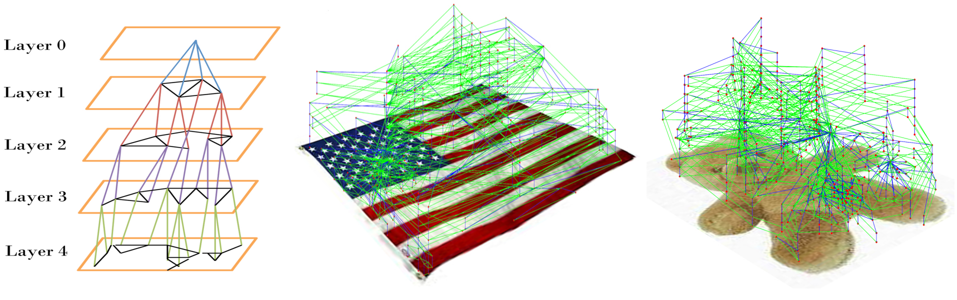
Figure 1: Constructing a class model, from left to right. (a): An input collection (possibly different depictions) used for training. (b): Probability maps for each input image, and graph models for each map. (c): The median graph model for the whole class. (d): The refined median graph as the final class model.

this is found in the fact that many artworks comprise simple shapes, and even sophisticated artists often paint over a skeleton comprising simple shapes.

Our model is a hierarchical graph, in which simple shapes label nodes, as in Figure 3. We are not alone in using a parts based hierarchy to model objects and object classes. Hierarchy of shapes or object regions are used to learn object class models (for example [6, 9, 14, 23, 28]). These build object class models, and most are motivated by a view we share: that such models should reflect the underlying object rather than its appearance. Some emphasise the importance of structural invariance [23, 27], as we agree this is an important property. Many hierarchies make use of spatial data [15, 16], as we do by labelling arcs with displacement vectors. None of the above use a median graph, as we do, to represent a visual object class. We construct a median graph via embedding [8]. Others construct a class specific *graph prototype* [26], but this is not the median graph and is labelled with SIFT features rather than qualitative shape.

Cross-depiction problems are little studied. Their importance is high for applications such as content based retrieval using sketches to index real video [4]. Matching static images has been addressed using self-similarity descriptors [17], and HoG features form the basis of a support vector machine is also capable of cross-depiction matching [20]. Neither of these are used in classification so far as we know, and the latter is computationally expensive – it requires thousands of images, we need just a few. Others use only structure for cross-depiction classification [27], which is a base for us to test against in our experiments. The method relies on spectral graph theory: it embeds a graph into a pattern space via the first few eigenvalues of the graph's Laplacian matrix, a fully supervised Gaussian Mixture Model is used as a classifier.

In summary, the problem of cross-depiction classification is little studied. We use a



(a) (b) (c)

Figure 2: *(a)* Relational graph model in schematic form. *(b)* A graph model of American flag. *(c)* A graph model of teddy bear. Parent-child arcs in blue, neighbour arcs in green.

hierarchical model of visual object class with nodes labelled by qualitative shape, such a model is unique so far as we know. We now describe the visual class model in greater detail: how to create them, and their value to the problem of classification.

3 Learn one model for each visual object class

We learn visual class models from input images, each labelled with the object they contain. There are three major steps: (i) build an "image graph" for each image in the training set; (ii) compute the class model as the median graph of the image graphs, and (iii) refine the class model by maximising classification performance over the training set. Figure 1 presents a framework of the proposed method. The steps are now discussed in detail.

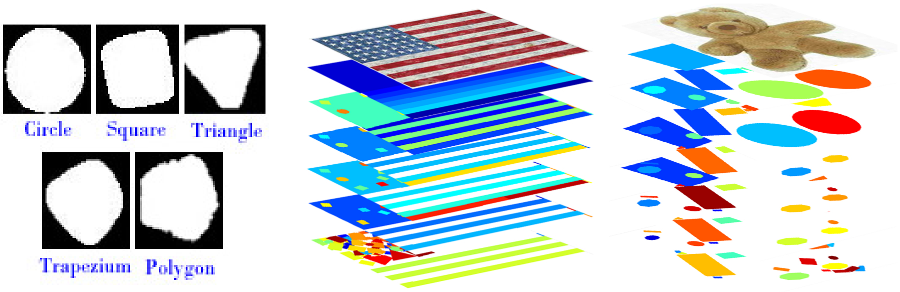
3.1 Build Image Graphs, one for each image.

Our modeller uses a state of the art segmentation algorithm from Berkeley that automatically yields a hierarchical description of an input image [1]. This outputs a sequence of segmentations indexed by thresholding over a probability map over region boundaries. The segmentations are ordered coarse-to-fine; smaller regions are nested inside larger ones.

We build an "image graph" from this in which regions label nodes. Undirected arcs at a given level (threshold value) denote touching neighbours. Directed arcs link a parent region in one level to the children it contains in the next level. There can be several hundred layers, but their number can be reduced to about ten or so, without loss of information, by a graph based filtering process [21]. This reduced graph is our starting point. Typical examples of reduced graph models can be seen in Figure 2.

We label graph arcs with displacement vectors between region centroids. We label nodes using qualitative shapes from $\mathbb{S} = \{circle, polygon, square, trapezium, triangle, random\}$. The first five of these shapes have been shown to explain around 80% of regions in photographs *up to an affine transform*; *random* is used when a region cannot be classified. See [25] for details on shape classification. "Polygon" captures pentagons, hexagons, *etc*.

More exactly, we label nodes with probability vectors over $\mathbb{S}$. The shape classifier is a mixture model over a feature space of (the absolute value of) Zernike moments. Each mixture component is itself a Gaussian Mixture Model. For each shape class $S \in \mathbb{S}$ we specify a GMM $(N_S, \{\alpha_{si}, \mu_{Si}, C_{Si}\}_{i=1}^{N})$, with N_S the number of GMM components, and α_{Si},



(a) (b) (c)

Figure 3: *(a)* Primitive shape classes (other than *random*)[25] *(b)* An American flag broken in primitive shapes. *(c)* A teddy bear likewise decomposed.

μ_{Si}, C_{Si} being the prior, mean, and covariance of each. For a region x we denote the density of the shape class by $p(x|S)$, which is readily computed using the standard form for a GMM,

$$p(x|S) = \sum_{i=1}^{N_S} p(x|\mu_{Si}, C_{Si}) \alpha_{Si}. \tag{1}$$

We label the corresponding graph node with a 6 elements vector of MAP estimate of shape-class membership:

$$p(S|x) = \frac{p(x|S)p(S)}{\sum_{T \in \mathcal{S}} p(x|T)p(T)}. \tag{2}$$

If an application requires a single shape, we use $S^* = \arg\max_S p(x|S)$. The shape-class prior, $p(S)$ is taken to be the relative number of shapes classified as shape S. All parameters used are provided by the shape classifier after training on about 40000 regions [25]. Figure 3 illustrates the shape classes we use, and the shape classes used to label nodes at each level of a hierarchy. This completes our construction of an image graph.

3.2 Compute an Initial Visual Class Model.

Given a set of image graphs, the next step is to compute the median graph model as the visual class model. The median graph, introduced into structural pattern recognition by Jiang *et al.* [13], is a useful concept that can be used to represent a set of graphs. A single prototype is extracted from a collection of graphs.

Let $\mathbb{G} = \{G_1, ...G_n\}$ be a set of graphs and let $d(G_i, G_j)$ be some distance function to measure the dissimilarity between graphs G_i and G_j. A simple approach to finding a median graph is to find the graph $G_k \in \mathbb{G}$ that minimises the sum of $d(.,.)$ over $\mathbb{G}$. A better approach is to choose the median graph, $\bar{G}$ from the set of all graphs that can be constructed from all combinations of all subgraphs of all graphs $G \in \mathbb{G}$. This vast set is denoted $\mathbb{U}$, and the median graph we use is defined using it:

$$\bar{G} = \arg\min_{H \in \mathbb{U}} \sum_{G_i \in \mathbb{G}}^{n} d(H, G_i). \tag{3}$$

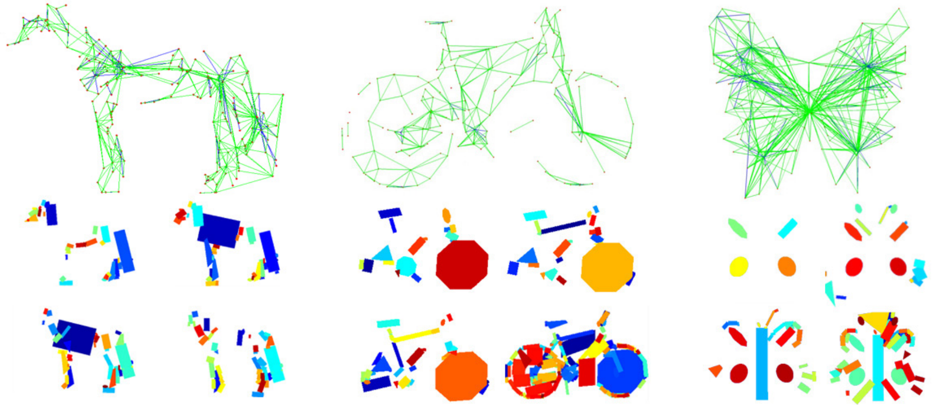


Figure 4: Examples of three graph models generated from 3 categories of objects, which are horses, bicycles, and butterflies. The visualization shows of selected levels below the corresponding model, with the simple shapes fitted. Child-parent arcs are in blue, adjacencies between the nodes in the same level are green.

This is far too large a problem to solve directly. In this paper we use an approximate algorithm for median graph computation proposed in [8].

For a set of image graphs generated as the section 3.1, $\mathbb{G} = \{G_1, G_2, ...G_n\}$, we first compute the graph edit distance (equal to the cost of a sequence of optimal edit operations, see section 3.2.1) between every pair of graphs in $\mathbb{G}$. Hence, an $n \times n$ distance matrix will be generated. Then, each row/column of the matrix can be seen as an n-dimensional vector, corresponding to each graph in $\mathbb{G}$. This embeds graphs into an n-dimensional feature space. Secondly, a median vector will be generated by computing the *Euclidean Median* of all the data points in the feature space. Finally, we transfer this median vector to a graph representation. This transformation process involves a *triangulation procedure*, which can be found in [8]. The result is our first approximation of the visual class model.

3.2.1 Graph edit distance

The graph edit distance, $d(G_1, G_2)$, of two graphs is equal to the cost of an optimal *ecgm* (error-tolerant graph matching) [3]. Formally, let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ to be two graphs, and the number of vertices of two graphs is not necessary equal. An error-correcting graph matching (*ecgm*) from G_1 to G_2 is a bijective mapping $X : \hat{v}_1 \rightarrow \hat{v}_2$, where $\hat{v}_1 \in V_1$ and $\hat{v}_2 \in V_2$, so the number of vertices of two matched sub-graph $|\hat{v}_1| = |\hat{v}_2|$. Then,

$$d(G_1, G_2) = c(X^*) \tag{4}$$

The cost function $c(X^*)$ here is the sum of distance of the edit operations implied by X^*, which is the optimal *ecgm* mapping and can be obtained in the following process. We follow Torresani *et al.* [22], who proposed a graph matching method based on global optimization. Given a pair of graphs, G_1 and G_2, the graph matching problem consists in finding a correspondence between nodes of G_1 and G_2 that maximise the following score of global consistency given as

$$E(X; G_1, G_2) = \sum_{i \in V_1, j \in V_2} x_{ij} \Phi_{i,j} + \sum_{i_1, i_2 \in V_1, j_1, j_2 \in V_2} x_{i_1 j_1} x_{i_2 j_2} \Theta_{e_1 e_2}, \tag{5}$$

where each X is a binary matrix that denotes the node-node correspondence and $e_1 = (i_1, i_2) \in E_1$, $e_2 = (j_1, j_2) \in E_2$. Maximising E gives an optimal edit path between two graphs, X^*. The two similarity matrices, Φ and Θ, measure the similarity of each node and each pair of edge respectively.

We specify Φ as the probability that two segmented regions are the same underlying simple shape. Suppose we have region i in graph G_1 and region j in G_2, then $p(S|i,j)$ denotes the probability that both are simple shape S. We specify

$$\Phi_{ij} := \max_{S \in \mathbb{S}} p(S|i,j) = \max_{S \in \mathbb{S}} p(S|i)p(S|j), \tag{6}$$

which assumes that regions are iid. This makes it easy to compute Φ_{ij} via equation 2.

The similarity of a pair of edges from two graphs, Θ, is obtained by evaluating how well the edge e_1 in graph G_1 matches the edge e_2 in graph G_2, in terms of both length and direction. Following [22] we specify edge similarity as

$$\Theta_{e_1 e_2} := \eta(\exp(\delta^2_{e_1 e_2}/\sigma^2_\iota) - 1) + (1-\eta)(\exp(\alpha^2_{e_1 e_2}/\sigma^2_\alpha) - 1) \tag{7}$$

in which, using p_1, p_2 and q_1, q_2 to denote region centroid of i_1, i_2, j_1, j_2:

$$\delta_{e1,e2} = \frac{|||p_1 - q_1|| - ||p_2 - q_2|||}{||p_1 - q_1|| + ||p_2 - q_2||} \text{ and } \alpha_{e1,e2} = \arccos\left(\frac{p_1 - q_1}{||p_1 - q_1||} \cdot \frac{p_2 - q_2}{||p_2 - q_2||}\right). \tag{8}$$

The parameter η is a scalar value trading off the importance of preserving distance versus preserving directions, we set $\eta = 0.5$. Variance values σ^2_ι and σ^2_α could (in principle) be learned from ground truth correspondences, but we set $\sigma^2_\iota = 0.5$ and $\sigma^2_\alpha = 0.9$ as the initialized value given by [22].

3.3 Refine the Visual Class Model.

The median graph contains nodes and arcs that derive from visual clutter in background of images in the training set. Hence, we developed a cleaning algorithm to remove such elements, and so refine the visual class model (*vcm*).

We begin by matching the median graph back into each training image, to count the number of times a given node in the model appears in the training data. This frequency count indicates the relevance of a node to the visual class. Next, we delete all nodes below a frequency threshold – we compute the matching score (using equation 5) between the edited *vcm* and each image in the training set. The threshold is then incremented, and the process repeats until the total match score is maximised. The nodes that remain define the final *vcm*. Figure 4 shows some final results.

4 Experiments and Results

Our visual class model (*vcm*) has the potential to be used in many applications, here we use classification – and cross-depiction classification in particular. Like any classification task, ours consists of two main steps, training and testing. Training comprises building a *vcm*, as described in Section 3. The testing process involves matching each *vcm* (one for each visual class under consideration) into the image graph that corresponds to an input test image. More

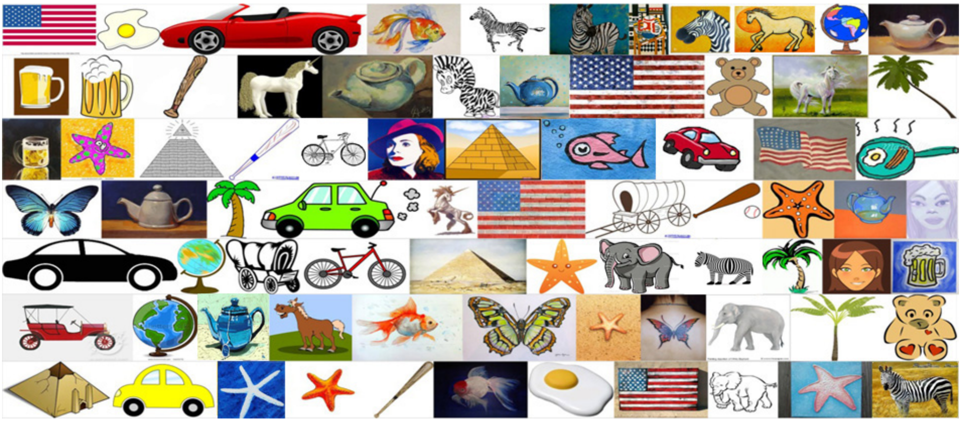


Figure 5: Some example pictures from our own dataset that augments CalTech 256.

formally, for n categories of object, we have a set of *vcms* with index set $\mathbb{N} = \{i\}_1^n$. Given an input image I we have its image graph $G[I]$, we compute

$$i = \arg\max_{i \in \mathbb{N}} E(X; G[I], G_i) \tag{9}$$

as the index number of the class to which the query image belongs. The similarity measure function is given by equation 5; notice that it ignores clutter nodes in $G[I]$.

To our best knowledge, there is not a published database and benchmark for this kind of cross-style object classification task. Therefore, we have augmented the Caltech-256 Object Category Dataset[10] with a parallel database that widens the variation in depictive style, see Figure 5. There are 20 categories of object in our own dataset now and more classes will be added. The dataset can be downloaded from here.

Using our expanded version of CalTech-256 we conduct experiments designed to test how well a visual class model generalised from across depictive styles. Specifically we: (1i) train on photographs alone and test on photographs; (1ii) train on artwork alone and test on artwork; (2i) train on photographs alone and test on artwork; (2ii) train on artwork alone and test on photographs; (3i) train on photo and test on both photographs and artwork; (3ii) train on artwork and test on both photographs and artwork; (4) train on both and test on both.

For comparison with alternative visual class models we conduct the above experiment using not just our *vcm* but with three others also. The first is a BoW classifier, chosen because it performs well and will help us assess the performance of such a popular approach to the problem of cross-depiction classification. The BoW we use is proposed in [24], it uses PHOW features [2] (dense multi-SIFT descriptors) and K-means for visual word dictionary construction. Finally, it uses an internal SVM for classification. Here we set the number of words in vocabularies at 600. And the other parameters are the same as [24] used, which can achieve 64% performance on Clatech101 dataset. The second is a shape-based method proposed in [7]. We first learn a shape model for each class by using the local *PAS features*. Then we compare each testing image with each shape model with a scoring function provided in [7] to decide which class it belongs to. The parameters we use are same as the paper provided except to change dissimilarity threshold γ to 10. The third *vcm* alternative we experiment with uses structure alone as a model [27] and is relevant because it explicitly sets out to classify in a cross-depiction domain. It uses the first few eigenvalues of the Laplacian matrix of the object structure as the feature vector, which embeds graphs in a pattern space. A GMM is employed as the classifier. Experimental results are shown in the following section.

4.1 Results and Discussion

Classification accuracy of different methods in various Training/Test cases, shown in table 1 (the deeper the color, the better the performance). The training and test images were selected to show objects on uncluttered backgrounds, which is also a limitation of our current work. The numbers of images in the table are *per-class* figures, the rates are averaged over 20 classes. In total our test used 800 images, including our extension to CalTech 256.

case 1: Training	5p	5a
case 1: Testing	15p	15a
Dense SIFT [24]	70%	59%
Shape Model [7]	25%	33%
Structure Only [27]	16%	19%
Proposed Method	61%	62%

case 2: Training	8p	10p	8a	10a
case2 : Testing	15a	15a	15p	15p
Dense SIFT [24]	43%	47%	49%	51%
Shape Model [7]	33%	35%	34%	34%
Structure Only [27]	19%	23%	22%	25%
Proposed Method	63%	64%	64%	67%

case 3: Training	3a	5a	3p	5p
case 3: Testing	30m	30m	30m	30m
Dense SIFT [24]	46%	50%	50%	54%
Shape Model [7]	27%	30%	24%	27%
Structure Only [27]	13%	16%	14%	16%
Proposed Method	58%	61%	56%	61%

case 4: Training	6m	10m
case 4: Testing	30m	30m
Dense SIFT [24]	60%	61%
Shape Model [7]	32%	34%
Structure Only [27]	21%	24%
Proposed Method	62%	65%

Table 1: Classification accuracy for different cases. From top to bottom, left to right: (a) single domain task, (b) single cross depiction task, and (c) single to mixture depiction task, (d) mixture cross depiction task. The character 'p' is 'photos', 'a' is 'art' and 'm' is 'mixture'. More detailed results for each single experiment can be found in supplementary material.

The table shows that our method outperforms the shape or structure only method in all cases. Our explanation is (i) structure-only method is not sufficiently rich, and (ii) we use more complex structures than the original [27]. We outperform BoW in all cases except case 1i, when photographs are used in both training and testing (or when the training set is very small, see supplementary material). Our rates are considerably higher than BoW for cross-depiction problems (cases 2 and 3). We are a little higher for mixed problems (case 4), more so when the number of training images rises.

Our explanation is that word formation in BoW favours features that exhibit low variance; photo-features will exhibit lower variation than art-features (thinking of all the ways to depict an eye). Thus BoW words are in some sense over-fitted and the method is biased towards a particular depictive style. The table (cases 2 and 3) suggests that our class model is able to generalise to depictions which has not been trained on, and that exhibits improved performance when the training set is mixed. When all cases are taken into account, our method is much more stable in performance (from 58% to 67%) compared to BoW (43 % to 70%). Our rates compare favourably to CalTech 256 benchmarks using only photographs (see `http://www.vision.caltech.edu/Image_Datasets/Caltech256/` and `http://www.vlfeat.org/applications/apps.html`.) We are taking a first step towards widening the classification problem.

5 Conclusion

The ability to generalise to new depictive styles is important, not least because the number of depictive styles is seemingly unbounded. No training procedure can capture them all and so a class model that is able to generalise to unseen depictive styles is of value. Experiments show that our proposal method performs better than the traditional visual appearance based method in cross-depiction problems (including to unseen depictive styles), in mixed problems, and in art-only problems.

However, there are still some limitations of this current system, for example, the methods fail when object is relatively small with complicated background, and the dataset is small, especially the number of positive examples. We do not yet localise objects in images, such an ability would improve our ability to learn. Our class exemplars exhibit a complex structure that would benefit from further simplification, *e.g.* using *graph prototypes* rather than median graphs. Additional labelling (for example texture on nodes, and affine maps) may also improve classification performance. We cannot model objects that exhibit high variation in structure and/or shape, *e.g.* buildings as a general class, such broad classes are a challenge to many classifiers. Our method depends on matching and so can be slow, faster algorithms – perhaps via a hierarchy of classes – are desirable.

Classification is just one application, future work may yet see further developments in areas such as non-photorealistic rendering as well as more traditional Computer Vision problems such as content based retrieval, and object localisation. Nonetheless, our results are a first step towards depiction invariant modelling.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011. ISSN 0162-8828.

[2] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.

[3] H. Bunke. Error-tolerant graph matching: A formal framework and algorithms. In Adnan Amin, Dov Dori, Pavel Pudil, and Herbert Freeman, editors, *Advances in Pattern Recognition*, volume 1451 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 1998. ISBN 978-3-540-64858-1.

[4] JP Collomosse, Graham McNeill, and Yu Qian. Storyboard sketches for content based video retrieval. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 245–252. IEEE, 2009.

[5] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and CÃľdric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[6] P.F. Felzenszwalb, Girshick R.B., McAllester D., and Ramanan D. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010.

[7] Vittorio Ferrari, Frederic Jurie, and Cordelia Schmid. From images to shape models for object detection. *International Journal of Computer Vision*, 87(3):284–303, 2010.

[8] Miquel Ferrer, Ernest Valveny, Francesc Serratosa, Kaspar Riesen, and Horst Bunke. An approximate algorithm for median graph computation using graph embedding. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[9] S. Fidler, M. Boben, and A. Leonardis. Similarity-based cross-layered hierarchical representation for object categorization. In *Computer Vision and Pattern Recognition*, 2008.

[10] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech - 256 object category dataset. 2007.

[11] Chunhui Gu, Joseph J Lim, Pablo Arbeláez, and Jitendra Malik. Recognition using regions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1030–1037. IEEE, 2009.

[12] Wei Jia and Stephen J. McKenna. *Classifying textile designs using bags of shapes*, pages 294–297. International Conference on Pattern Recognition. IEEE Computer Society, 2010. ISBN 9780769541099.

[13] X. Jiang, A. Munger, and H. Bunke. An median graphs: properties, algorithms, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23 (10):1144–1151, 2001. ISSN 0162-8828.

[14] J.J. Lim, P. ArbelaÌĄez, Chunhui Gu, and J. Malik. Context by region ancestry. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1978–1985, 2009.

[15] Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. Stacked hierarchical labeling. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, pages 57–70, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15566-9, 978-3-642-15566-6.

[16] Alessandro Perina, Nebojsa Jojic, Umberto Castellani, Marco Cristani, and Vittorio Murino. Object recognition with hierarchical stel models. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, pages 15–28, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15566-9, 978-3-642-15566-6.

[17] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007.

[18] Ali Shokoufandeh, Lars Bretzner, Diego Macrini, M Fatih Demirci, Clas Jönsson, and Sven Dickinson. The representation and matching of categorical shape. *Computer Vision and Image Understanding*, 103(2):139–154, 2006.

[19] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multiscale categorical object recognition using contour fragments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1270–1281, July 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.70772.

[20] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Data-driven visual similarity for cross-domain image matching. *ACM Transactions on Graphics (TOG)*, 30(6), 2011.

[21] Yi-Zhe Song, Pablo Arbelaez, Peter Hall, Chuan Li, and Anupriya Balikai. Finding semantic structures in image hierarchies using laplacian graph energy. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 694–707, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15560-X, 978-3-642-15560-4.

[22] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. *Computer Vision–ECCV 2008*, pages 596–609, 2008.

[23] N. Trinh and B. Kimia. Skeleton search: Category-specific object recognition and segmentation using a skeletal shape model. *International Journal of Computer Vision*, 94:215–240, 2011.

[24] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[25] Qi Wu and Peter Hall. Prime shapes in natural images. In *Proceedings of the British Machine Vision Conference*, pages 45.1–45.12. BMVA Press, 2012. ISBN 1-901725-46-4.

[26] Shengping Xia and Edwin R Hancock. Learning class specific graph prototypes. In *Image Analysis and Processing–ICIAP 2009*, pages 269–277. Springer, 2009.

[27] Bai Xiao, Song Yi-Zhe, and Peter Hall. Learning invariant structure for object identification by using graph methods. *Computer Vision and Image Understanding*, 115(7): 1023–1031, 2011.

[28] Long Zhu, Yuanhao Chen, A. Torralba, W. Freeman, and A. Yuille. Part and appearance sharing: Recursive compositional models for multi-view. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1919–1926, 2010.