

Classifying Textile Designs using Region Graphs

Wei Jia
weijia@computing.dundee.ac.uk
Stephen J. McKenna
stephen@computing.dundee.ac.uk
Annette A. Ward
award@computing.dundee.ac.uk
Keith Edwards
kedwards@computing.dundee.ac.uk

School of Computing
University of Dundee
Scotland, UK

Abstract

Markov random field pixel labelling is often used to obtain image segmentations in which each segment or region is labelled according to its attributes such as colour or texture. This paper explores the use of such a representation for image classification. In particular, the problem of classifying textile images according to design type is addressed. Regions with the same label are treated as a group and each group is associated uniquely with a vertex in an undirected, weighted graph. Each region group is represented as a bag of shape descriptors. Edges in the graph denote either the extent to which the groups' regions are spatially adjacent or the dissimilarity of their respective bags of shapes. Series of unweighted graphs are obtained by removing edges in order of weight. Finally, an image is represented using its shape descriptors along with features derived from the chromatic numbers or domination numbers of the unweighted graphs and their complements. Experimental results are reported on a challenging classification task using images from a textile design archive.

1 Introduction

Image recognition has often been based on descriptors of image regions that result from segmentation [1, 6, 7]. This paper explores the use of region shape descriptors resulting from segmentation by pixel labelling using a Markov random field (MRF) model. As well as partitioning an image into regions, MRF labelling identifies groups of regions that have the same label values. The approach proposed here seeks to represent relationships between these groups of regions that might be useful for tasks such as classification. This leads to a weighted graph in which vertices are associated with region groups and edge weights indicate relationships between the groups. The edge weights can encode information on spatial adjacency or alternatively the dissimilarity of the regions in the groups. Such a representation differs from traditional region adjacency graphs since the vertices correspond to *groups* of regions rather than individual regions. Consequently, these graphs tend to have far fewer vertices. A method for obtaining feature vectors of fixed dimensionality from these graphs is

proposed so that standard machine learning methods can be used. This involves generating series of unweighted graphs from the weighted graphs via edge removal. Feature vectors can then be based on graph-theoretic measures such as chromatic number and domination number.

This method is applied to image classification using examples from a commercial textile archive owned by Liberty Fabric Ltd. The collection holds over 13,000 images, primarily textile swatches but also prints and other original artwork. The classification problem based on this database is challenging because of large intra-class variations and because images from different classes often have much in common. Furthermore, the textile swatches span more than a century of design history and are often damaged, degraded or partially occluded by labelling.

2 Overview of Method

2.1 Graph construction

Figure 1(a) shows an example of an image segmented into groups of regions by assigning each pixel a label; the label image is shown in the centre. Given such a labelling, the image can be represented as a *bag of shapes* by computing shape descriptors for each connected component [9]. However, a bag of shapes model ignores relationships between the groups of regions. In order to retain information about these relationships, we construct undirected weighted graphs as shown in Figure 1(b). Each group of regions is represented as a bag of shapes. Edges in the graph encode relationships between the groups. For example, edges can encode the extent to which the groups' regions are spatially adjacent, or the dissimilarity of their respective bags of shapes.

An undirected graph $G = (V, E)$ consists of a set of vertices V and a set of edges E . Each vertex $v \in V$ is associated with a group of regions (bag of shapes). Two ways in which edge weights can be assigned are:

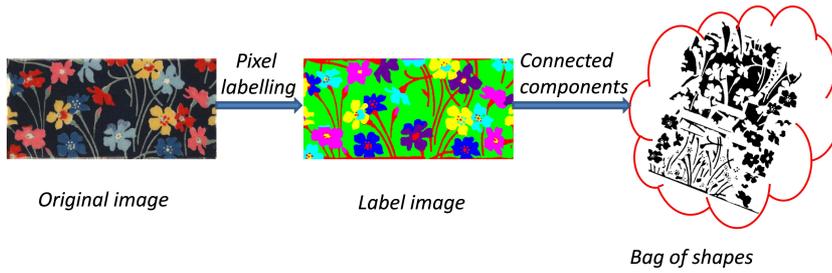
1. A weight is assigned the arc length of the common boundary shared by the groups of regions. This is the extent to which two groups' regions are spatially adjacent.
2. A weight is assigned a measure of dissimilarity of the groups of regions.

The complement of a graph G is the graph G' that has the same vertex set and an edge set consisting of the edges not present in G [17]. The graph sum $G + G'$ is the complete graph in which every pair of distinct vertices is connected by an edge.

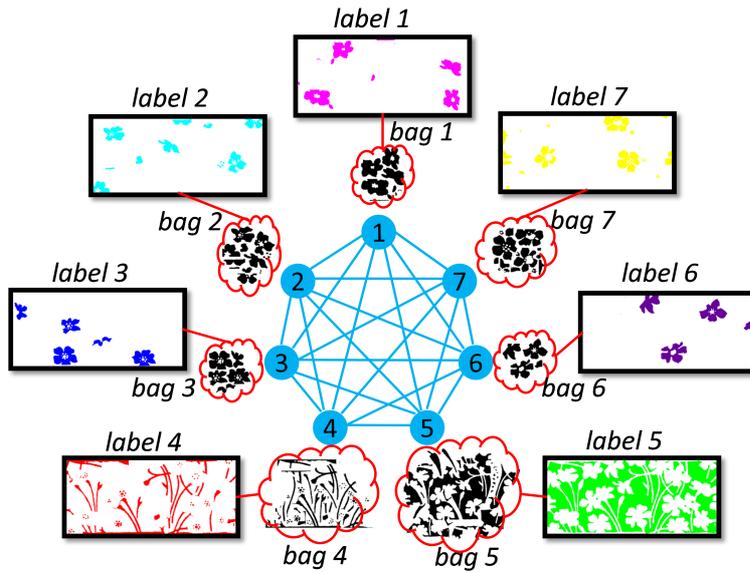
2.2 Chromatic number and domination number sequences

In graph theory, the chromatic number of a graph, G , is the smallest number of colours needed to colour the vertices without adjacent vertices sharing the same colour [17]. Consider toy examples with three groups of regions as vertices of a graph, fully connected to each other. The edge weights are assigned values proportional to the arc lengths of the common boundaries shared by the groups of regions. Two such graphs are as follows.

1. Figure 2(a) shows an example in which the groups of regions are equally adjacent to each other so that all edges are assigned the same weight. Deleting edges in order of weight generates the chromatic number sequence $3 \rightarrow 1$. This is an extreme case.



(a)



(b)

Figure 1: (a) An image is segmented into labelled regions. It can then be represented using a *bag of shapes*. (b) Alternatively, a weighted graph can be constructed in which each vertex is associated with a group of regions that share the same label. Each group of regions is represented as a *bag of shapes*. Edge weights encode relationships between the groups.

2. Figure 2(b) shows an example in which edges are assigned different weights. Deleting edges in order of weight generates the chromatic number sequence $3 \rightarrow 2 \rightarrow 2 \rightarrow 1$

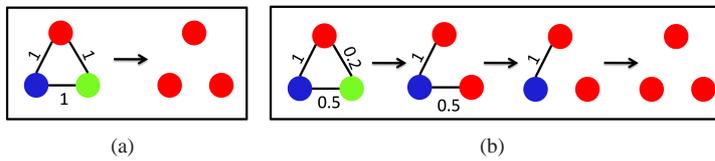


Figure 2: Graph minimal colouring sequences obtained by deleting edges in the order of weight.

These examples illustrate that deleting edges by weight results in sequences of chromatic numbers that depend on the adjacency relationships of the region groups. Similarly, sequences of chromatic numbers can be computed from the complement graphs. Calculating chromatic numbers is NP-complete [17]. However, the graphs obtained have relatively few vertices so they can be computed quickly. A graph colouring algorithm based on backtracking was used [4].

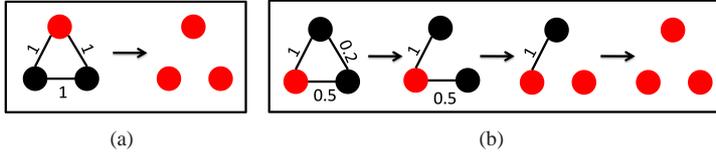


Figure 3: Graph dominating set sequences (red dots) obtained by deleting edges in the order of weight.

Another measure from graph theory that can reflect relationships between the vertices (groups of regions) is the domination number [8]. A dominating set of a graph is a subset D of V such that every vertex in $V - D$ is joined to at least one vertex in D by some edge. Figure 3 shows dominant sets (red nodes) of groups obtained by deleting edges in order of weight. The number of red nodes is the domination number. Thus, in the example of Figure 3(a) the domination number sequence is $1 \rightarrow 3$ and in the example of Figure 3(b) the domination number sequence is $1 \rightarrow 1 \rightarrow 2 \rightarrow 3$. Similarly, we can compute sequences of domination numbers from complement graphs. Finding the domination number is also an NP-complete problem [5]. The domination number was calculated using the Bron-Kerbosch algorithm [2].

The sequence of (normalised) weights of those edges whose removal changes the chromatic number constitutes a feature vector. Similarly, a feature vector can be computed based on changes in domination number. Figure 4 plots chromatic number against normalized weights for one of the examples above. The resulting feature vector is $[1.0, 0.2, 0.0, \dots, 0.0]$.

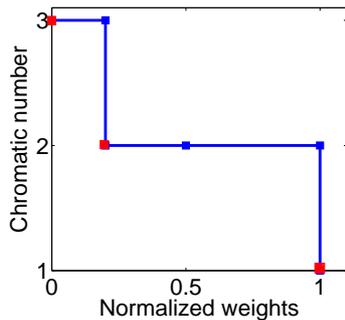


Figure 4: The change in chromatic number as edges are removed in order of weight for the example in Figure 2(b). The resulting feature vector is $[1.0, 0.2, 0.0, \dots, 0.0]$.

3 MRF pixel labelling

In order to obtain label images such as the one shown in Figure 1(a), MRF pixel multi-labelling was used. This method alternated between estimating Gaussian distributions in RGB space and minimising energy in an MRF model with the RGB distributions as likelihood functions. Such an energy function has the form

$$E(f) = \sum_p \sum_q \lambda \cdot (1 - \delta(f_p - f_q)) - \sum_p \ln P(x_p | f_p), \quad (1)$$

where f is an image labelling, f_p denotes the label assigned to a pixel p , and pixels denoted q are neighbours of pixel p . The first term rewards spatial coherence and the second term rewards a good fit for the RGB distributions. The parameter $\lambda \geq 0$ specifies the penalty for assigning different labels to neighbouring pixels. Optimization was performed using α -expansion steps with $\lambda = 4$. Further details can be found elsewhere [10].

4 Bags of shapes

The shape of each region (connected component) was described using generic Fourier descriptors (GFD) [19]. Specifically, a 2D Fourier transform was applied to a polar representation $f(r, \theta)$ of each binary region image (see Figure 5):

$$F(\rho, \phi) = \sum_r \sum_\theta f(r, \theta) \exp[-j2\pi(r\rho + \theta\phi)], \quad (2)$$

where $0 \leq r < 1$ and $0 \leq \theta < 2\pi$. The generic Fourier descriptor (GFD) is:

$$\mathbf{d} = \left(\frac{|F(0,0)|}{\text{area}}, \frac{|F(0,1)|}{|F(0,0)|}, \dots, \frac{|F(m-1,n-1)|}{|F(0,0)|} \right) \quad (3)$$

Setting $m = 4$ and $n = 12$ gave a 48-dimensional feature vector. This representation is rotation, translation and scale invariant.



Figure 5: A region in polar coordinates

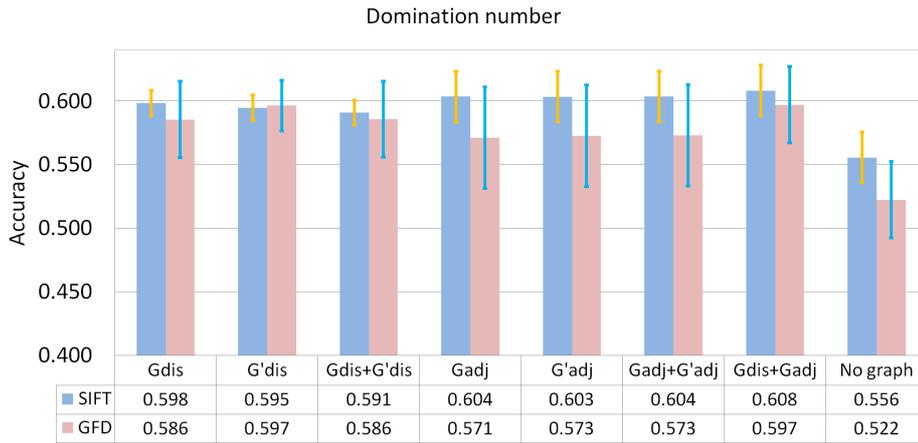
Given a collection of shapes from training images, a codebook was calculated by running k-means [12] on the shape descriptors. Codewords were defined as the centres of the clusters [13]. A given shape can be assigned to the nearest codeword. A set of shapes can be represented as a histogram of the codewords.

5 Experiment

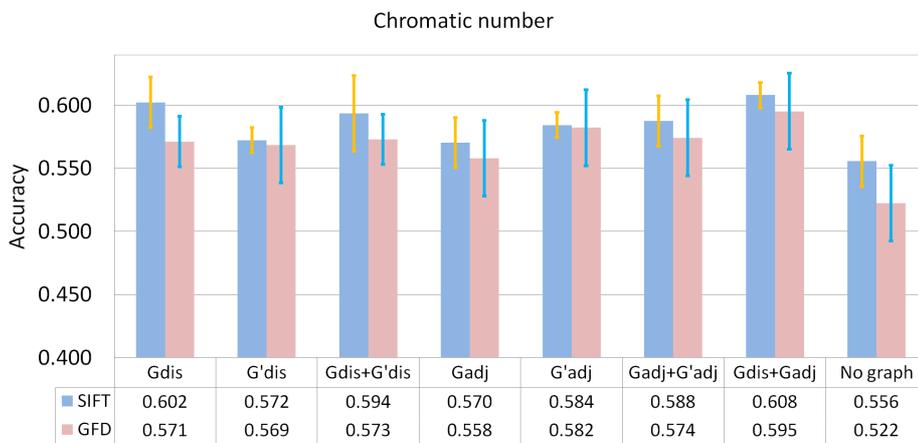
The dataset contained 490 images manually categorised into 7 classes with 70 examples per class. Figure 4 shows some examples of each class. The images were categorised based on first level text descriptors that were generally termed "style" within the archive database.

	Correctly classified examples		Misclassified examples	
Floral				
Paisley				
Stripe				
Leaf				
Geometric				
Spot				
Check				

Figure 6: Example images from the seven categories. Images on the left were correctly classified. Images on the right were sometimes misclassified. The wrongly assigned class labels are shown above these images.



(a)



(b)

Figure 7: Classification results obtained using cross-validation with different feature sets. See main text for details.

It should be noted, however, that one of the descriptors (geometric) defined a broad design style that included images with more variable content but with geometric attributes (e.g., angles, lines, simple shapes). Some terms (floral, paisley, and leaf) referred to the design content of the image in familiar textile design terms (especially in the cases of floral and paisley); others (stripe, spot, and check) provided both a description of style and visual content. This classification problem is challenging because of large intra-class variations and because images from different classes often have much in common.

Seven-fold cross validation (leave-ten-out) was used to evaluate the accuracy of classification. We compared different feature sets based on a linear SVM classifier [3]. The regularisation parameter was set to 0.5. The number of labels used for MRF pixel labelling of each image was decided based on image metadata [10]. Each image was represented by a bag of shapes using 48-dimensional generic Fourier descriptors. A shape codebook with 500

codewords was used. Chromatic numbers were calculated using the TORSICHE Scheduling Toolbox [18]. The implementation for calculating domination numbers was based on the Matgraph toolbox [16].

Graphs were constructed for each image as described in Section 2.1. In the case of using dissimilarity between groups of regions, L_2 distance was used as the distance between code-word histograms. For an example image, Figures 8(a) and 8(c) plot the chromatic numbers and domination numbers of the graph and its complement against the weights of edges removed. Note that these are monotonic functions and that the weights are normalized in the range $[0, 1]$. 12-dimensional feature vectors were derived in the case of both chromatic numbers and domination numbers. The corresponding feature vectors are shown in Figures 8(b) and 8(d).

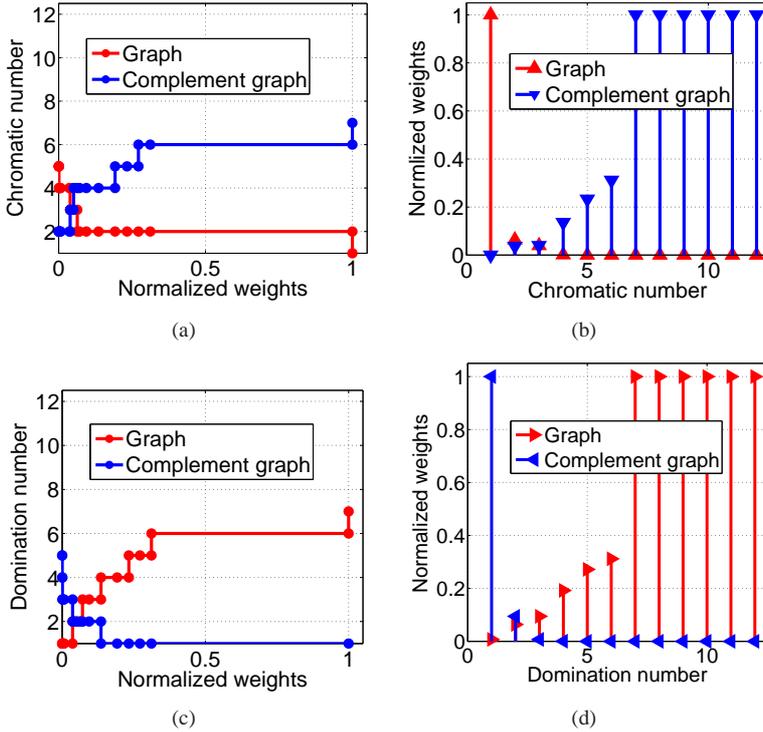


Figure 8: (a) Chromatic numbers of G and G' plotted against normalized edge weights. (b) Features derived from chromatic numbers. (c) Domination numbers of G and G' plotted against normalized edge weights. (d) Features derived from domination number.

Since representing images as bags of local patch descriptors [13] such as SIFT [14] is popular, we also ran the experiment using SIFT features instead of GFD. 500 local patches were randomly sampled from each image. The patch diameters were sampled at random in the range 10 to 30 pixels. Each patch was described as a 128-dimensional SIFT feature vector using an existing implementation [15].

The accuracy of classification using different feature sets is shown in Figure 7. G_{dis} denotes the graph features obtained by constructing graphs using dissimilarity as edge weights. G'_{dis} denotes features obtained from the complement graphs. G_{adj} denotes the graph features obtained by constructing graphs using arc length of the common boundary shared by

the groups of regions. G'_{adj} denotes features obtained from the complement graphs. The graph features used in Figures 7(a) and 7(b) were calculated from domination numbers and chromatic numbers respectively. Error bars denote \pm one standard deviation estimated over 10 runs of seven-fold cross validation. No matter what kind of graph features were used, the results suggest that classification accuracy was better than using GFD or SIFT features alone. In general, SIFT appears to be slightly more accurate than GFD. However, GFD has lower dimensionality. The combined use of graphs with edges representing adjacency and dissimilarity gave classification accuracy at least as good as other feature sets.

Figure 9 shows a confusion matrix obtained using GFDs and G_{dis} graph features derived from domination number.

floral	74.3%	11.4%	0	10%	4.3%	0	0
paisley	22.9%	57.1%	0	11.4%	0	0	8.6%
strip	0	0	75.7%	8.6%	2.9%	0	12.8%
leaf	30%	8.5%	5.8%	60%	5.7%	0	0
geometric	5.7%	0	10%	14.3%	47.1%	8.6%	14.3%
spot	0	0	0	10%	14.3%	57.1%	8.6%
check	0	0	15.7%	10%	12.9%	5.7%	55.7%
	fl	ps	st	lf	gm	sp	ck

Figure 9: A confusion matrix obtained using GFD combined with domination features, G_{dis} .

6 Conclusion and Recommendations

The experiments have demonstrated that an improvement in classification accuracy was obtained by combining bags of shapes with the proposed graph-based feature vectors. In order to test the effectiveness of the algorithms, single text keywords assigned to the images were used to compare human labelling with machine classification. These varied ways of describing the images should be further refined and developed in future work. An alternative to deriving feature vectors from a graph is to employ graph kernel methods [7, 11]. Given that the graphs used in this paper have relatively few vertices, kernels could be designed without so much regard to computational complexity as is needed when dealing with large graphs. It would be interesting to compare this approach.

References

- [1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009.
- [2] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16 (9):575–577, 1973.

- [3] C. C. Chang and C-J. Lin. LIBSVM: A library for support vector machines. Software available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
- [4] J. Demel. *Graphs and their Applications*. 1st Prague: Academia, 2002.
- [5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [6] C. H. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, 2009.
- [7] Z. Harchaoui. Image classification with segmentation graph kernels. In *CVPR*, 2007.
- [8] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Domination in Graphs-Advanced Topics*. New York: Dekker, 1998.
- [9] W. Jia and S. J. McKenna. Classifying textile designs using bags of shapes. In *International Conference on Pattern Recognition (ICPR)*, 2010.
- [10] W. Jia, S. J. McKenna, and A. A. Ward. Extracting printed designs and woven patterns from textile images. In *Int. Conf. Computer Vision Theory and Application (VISAPP)*, 2009.
- [11] H. Kashima, K. Tsuda, and A. Inokuchi. *Kernels for Graphs*. MIT Press, 2004.
- [12] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43 (1):29–44, 2001.
- [13] F. F. Li and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.
- [14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60 (2): 91–110, 2004.
- [15] K. Mikolajczyk and C. Schmid. Region Descriptors software available at : www.robots.ox.ac.uk/~vgg/research/affine/descriptors.html.
- [16] E. R. Sheinerman. MATGRAPH: A Matlab toolbox for graph theory. Software available at: <http://www.ams.jhu.edu/~ers/matgraph/>, June 2007.
- [17] S. Skiena. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Reading, MA: Addison-Wesley, 1990.
- [18] P. Šůcha, M. Kutil, M. Sojka, and Z. Hanzálek. TORSCHE scheduling toolbox for Matlab. In *IEEE Computer Aided Control Systems Design Symposium (CACSD'06)*, pages 1181–1186, Munich, Germany, October 2006.
- [19] D. S. Zhang and G. J. Lu. Shape-based image retrieval using generic Fourier descriptor. *Sig. Proc.: Image Communication*, 17:825–848, 2002.