Feature Selection for Graph-Based Image Classifiers

Bertrand Le Saux and Horst Bunke

Institut für Informatik und Angewandte Mathematik,
University of Bern, Neubrückstrasse, 10, CH-3012, Bern, Switzerland
{lesaux, bunke}@iam.unibern.de

Abstract. The interpretation of natural scenes, generally so obvious and effortless for humans, still remains a challenge in computer vision. We propose in this article to design binary classifiers capable to recognise some generic image categories. Images are represented by graphs of regions and we define a graph edit distance to measure the dissimilarity between them. Furthermore a feature selection step is used to pick in the image the most meaningful regions for a given category and thus have a compact and appropriate graph representation.

1 Introduction

How can one construct computer programmes in order to understand the content of scenes? Such programmes would satisfy needs in image retrieval and computer vision, and could possibly be applied to a wide range of areas, including security, digital libraries and web searching. We propose in this article to design binary classifiers capable to recognise some generic image categories.

Previously, image classification has been performed by using directly support vector machines on image histograms [1] or hidden Markov models on multi-resolution features [2]. These methods do not take into account that human description of an image content is rarely global but often specific to an image part. To include local information, attributed relational graphs [3] and image blocks [4] were proposed. Such approaches rely on the ability of the classifiers to distinguish between complex features, so they are prone to over-fit when the concept to learn has a large variance.

Our approach segments images into regions and index each image by a graph of regions. For a given type of scene, only image parts that are meaningful in that case are selected in order to make easier the task of the classifiers. This allows to define an efficient comparison scheme between the graphs that represent the images.

This paper is organised as follows. In Sect. 2, we explain how to describe the images and how to select the meaningful regions. The graph matching procedure and the classification process are described in Sect. 3. Finally, we present some experiments and discuss their results in Sect. 4.

(a) (b) (c)

Fig. 1. The original image (a) is first segmented (b), and then we keep only the regions (c) the type of which has a large mutual information with the scene to predict. When indexing this image with respect to the *countryside* label, the sky and the buildings are considered as non-informative and are discarded.

2 Image Representation

2.1 From Images to Regions

Images are first segmented into regions by using the mean shift algorithm [5]. This is a simple non-parametric technique for maximisation of the probability density. It basically performs a density gradient ascent.

To perform colour segmentation, the mean shift procedure is applied at various start locations, then the obtained high density colours are mapped to the image plane to keep only those belonging to large enough regions. Typically, this technique gives results as shown on Fig. 1: there are less than 10 regions per image, that are not necessarily connected but correspond more or less to the main semantic areas since colour is an important visual cue for generic images.

2.2 Feature Selection

Region Lexicon. The region lexicon consists of a list of the region types that occur in an image data set. Such a data set is built by gathering various generic images. Once they are segmented, these images are assumed to provide a good representation of the possible image regions that occur in the real world.

We cluster this data set of image regions using techniques previously proposed to find clusters of visually similar images in image databases [6] and based on fuzzy clustering methods [7]. The resulting clusters contain visually similar image regions and thus define implicitly a region type. Each of them is included in the region lexicon.

Selection of Meaningful Regions. For a segmented image, we can determine the type of each region simply by computing the distance (based on the region descriptor) to the cluster centroids and choosing the closest one. Let $\mathcal{I}$ denote the set of images, and X a random variable on $\mathcal{I}$ standing for the distribution of images. We can build a set of features $F = \{f_1, \ldots, f_N\}$ which are mappings from $\mathcal{I} \rightarrow \{0, 1\}$. In the experiments those features are indicators of the presence - or

absence - of a given region type in the image. We denote $F_1 = f_1(X), \ldots, F_p = f_p(X)$ the boolean random variables associated with those features.

In order to understand which region types are meaningful to recognise a concept, a filtering phase based on feature selection [8] is applied as in [9]. The most standard ways to select features consist in ranking them according to their individual predictive power, that may be estimated by mutual information [10].

Information theory [11] provides tools to assess the available features. The entropy measures the average number of bits required to encode the value of a random variable. For instance, if we denote Y a boolean random variable standing for the class to predict (i.e. the concept to associate with the image), its entropy is $H(Y) = -\sum_y P(Y = y)\log(P(Y = y))$. The conditional entropy $H(Y|F_j) = H(Y, F_j) - H(F_j)$ quantifies the number of bits required to describe Y when the feature F_j is already known. The mutual information of the class and the feature quantifies how much information is shared between them and is defined by:

$$\begin{aligned} I(Y, F_j) &= H(Y) - H(Y|F_j) \\ &= H(Y) + H(F_j) - H(Y, F_j) \end{aligned} \tag{1}$$

The features f_j are ranked according to the information $I(Y, F_j)$ they convey about the class to predict, and those with the largest mutual information are chosen. In the image, we keep only the regions that have a region type among the selected ones (cf. Fig. 1). They are the most meaningful ones to recognise the concept.

2.3 From Regions to Graphs

Definition 1. *A graph G is a 4-tuple $G = (V, E, \mu, \nu)$ where*

- *V is the set of vertices;*
- *$E \subseteq V \times V$ is the set of edges;*
- *$\mu : V \to L_V$ is a function assigning labels to the vertices;*
- *$\nu : E \to L_E$ is a function assigning labels to the edges.*

Two different alternatives to represent images by a graph are investigated in this paper. In either case, each region constitutes a vertex of the graph. The vertex labels are the colour histograms that characterise the corresponding region. In the first type of graph representation, only vertices corresponding to adjacent regions (i.e. with at least one point of contact) are linked by an edge, with no label. In the second graph representation, all vertices are linked to all the other ones, with a label defined proportionally to the common boundary length (CBL). Both types of graphs are undirected.

3 Image Classification

Classification of images implies to be able to measure the similarity between the graphs representing the images. Moreover in the case of images, data are

usually corrupted by noise and strongly depend on illumination conditions. Error correcting methods for graph matching have been proposed to cope with these problems. Among them, the graph edit distance is particularly popular. It defines a set of possible edit operations and assigns a cost to each of them. The distance of two graphs is then the minimum cost of all sequences of edit operations that transform a graph into the other. To compute the graph edit distance, we use the A^* algorithm [12] as described for graph matching in [13]. A look-ahead procedure [14] is used to speed up the matching process. Last, a k-Nearest-Neighbour (k-NN) classifier is used to classify the images. Next sections describe the edit operations and their associated cost.

3.1 Graph Edit Operations

Let p be a mapping between the vertices of two graphs $G_1 = (V_1, E_1, \mu_1, \nu_1)$ and $G_2 = (V_2, E_2, \mu_2, \nu_2)$. We assume that G_1 and G_2 are such that $\text{Card}(V_1) \leq \text{Card}(V_2)$. This mapping consists of elementary mappings (v, w), $v \in V_1$ and $w \in V_2$ such that each vertex is used only once. The \$ element denotes a missing vertex in graph G_2. For each couple (v, w) in p, the possible vertex edit operations are defined as follows:

- vertex label substitution: if $w \neq \$$ the mapping implies the substitution of $\mu_1(v)$ by $\mu_2(w)$.
- vertex deletion: if $w = \$$, it implies the deletion of v from G_1.

For each pair of elementary mappings (v, w) and (v', w') in p, the possible edge edit operations are defined as follows:

- edge label substitution: if $\exists$ an edge $e_1 = (v, v') \in E_1$ and an edge $e_2 = (w, w') \in E_2$, the mapping implies the substitution of edge label $\nu_1(v, v')$ by $\nu_2(w, w')$.
- edge deletion: if $\exists$ an edge $e_1 = (v, v') \in E_1$ and there is no edge $(w, w') \in E_2$, it implies the deletion of e_1 from E_1.
- edge insertion: if there is no edge $(v, v') \in E_1$ but $\exists$ an edge $e_2 = (w, w') \in E_2$, then $e_1 = (v, v')$ has to be inserted in E_1.

3.2 Graph Edit Costs

Different sets of graph edit costs are defined for the two graph representations of images defined in Sect. 2.3. In both cases, the vertices convey the visual information about the image regions, so the vertex edit operations have the same cost:

Definition 2. *Vertex edit costs for both graph representations:*

- *vertex label substitution: the cost of the substitution of $\mu_1(v)$ by $\mu_2(w)$ is the Euclidean distance between the labels (i.e. the colour histograms of the image regions):* $c(\mu_1(v) \rightarrow \mu_2(w)) = ||\mu_1(v) - \mu_2(w)||_2$.

– *vertex deletion: to make the deletion easier on large graphs than on small ones:* $c(v \rightarrow \$) = \frac{1}{\mathrm{Card}(V_1)}$.

In the first graph representation, graphs have only edges corresponding to adjacent regions and the corresponding costs are defined as:

Definition 3. *Edge edit costs for set* #1*:*

– *edge label substitution: by definition there is a perfect match so there is no cost:* $c(\nu_1(e_1) \rightarrow \nu_2(e_2)) = 0$.
– *edge deletion: to take into account the size of the graph and have comparable costs:* $c(e_1 \rightarrow \$) = \frac{1}{\mathrm{Card}(V_1)}$.
– *edge insertion: by symmetry:* $c(\$ \rightarrow e_1) = \frac{1}{\mathrm{Card}(V_1)}$.

A second way to define the edges is based on the the common boundary length (CBL) of two regions. The edge label could be defined as the CBL itself, or a normalised value based on the CBL, for example $\max(\frac{CBL}{BL_{\mathrm{reg1}}}, \frac{CBL}{BL_{\mathrm{reg2}}})$ or $\mathrm{avg}(\frac{CBL}{BL_{\mathrm{reg1}}}, \frac{CBL}{BL_{\mathrm{reg2}}})$ where $BL_{\mathrm{reg}i}$ is the boundary length of region i. For such graphs, since there exist edges between all pairs of vertices, there is no need anymore for edge deletion or insertion operations:

Definition 4. *Edge edit costs for set* #2*.*

– *edge label substitution: for any pair of edges* e_1 *and* e_2*,*

$$c(\nu_1(e_1) \rightarrow \nu_2(e_2)) = ||\nu_1(e_1) - \nu_2(e_2)||_2$$

4 Experiments

4.1 Data Set

The data set is composed of 200 images collected from the web. Four classes contain instances of a particular scene type: *snowy, countryside, streets* and *people*. A fifth one consists of various generic images aimed to catch a glimpse of the possible real scenes and thus used as negative samples for the classifiers. In the experiments, training categories of 30 instances are extracted randomly from the data set and error rates are averaged on 25 runs. Some examples are shown in Fig. 2.

4.2 Graph Matching Classification

The edit cost sets proposed in Sect. 3.2 are compared in Table 1. The quality of the considered set of edit costs depends on the complexity of the underlying scenes. For class *snowy* which is rather easy to recognise, the second set of edit costs is superior to the first one. However, for class *people* which is rather difficult, the situation is just the opposite. Since we intend to build some generic classifiers able to recognise different types of scenes, they have to satisfy an overall criterion

Fig. 2. The meaningful regions correspond to the region types that have a high mutual information with the label to predict. The upper row shows the original images and the lower row shows only the meaningful regions in these images.

including both the smallest average error and the smallest standard deviation. The last graph representation has the best test error rate on average, but shows large disparities between scenes. We observe that edge labels based on the simple adjacency between the regions result in the best overall performance.

Figure 3-a illustrates the influence of the number of neighbours in the classifier on the test error rate. For the complex scenes, the graphs indicate there exists an optimal value (around 15 neighbours). This is less obvious on simpler scenes like *country*, for which error rates are rather constant, with a slight trend to increase with the number of neighbours. The number of neighbours is then set to 15 for the complex scenes and 5 for the simple ones.

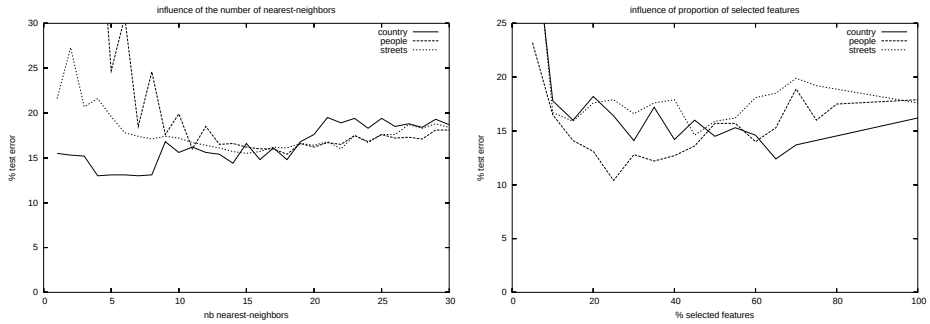
4.3 Influence of the Feature Selection

For each type of scene, the feature selection allows to pick out the meaningful parts of the image. Figure 2 shows the selected regions are consistent with what can be expected intuitively. The influence of the proportion of selected features on the test error rate is presented in Fig. 3-b. The graphs show that lower error rates can be obtained by selecting roughly between a third and a half of the region types: the optimal values are then chosen as a tuning reference for each concept. Table 2 compares the error rates with and without region selection: performance is improved for each category.

Table 3 compiles the computing times of the graph matching process and the image classification task (performed on a computer with a 800 MHz processor) for various proportions of selected features. Since the algorithm complexity is exponential with the number of vertices, feature selection appears as an intelligent way to greatly speed up the process.

Table 1. Error rates for various keywords: comparison of various edit cost sets.

keyword	edit cost set #1		edit cost set #2 ($\nu = CBL$)		edit cost set #2 ($\nu = max(\frac{CBL}{BL_1}, \frac{CBL}{BL_2})$)	
	training error	test error	training error	test error	training error	test error
snowy	8.4 %	11.4 %	9.4 %	8.2 %	9.1 %	7.9 %
country	14.5 %	16.3 %	16.5 %	15.8 %	15.4 %	14.4 %
people	12.8 %	15.6 %	19.1 %	20.9 %	17.5 %	19.4 %
streets	14.6 %	17.3 %	16.9 %	16.0 %	17.3 %	15.3 %
mean		15.15 %		15.22 %		13.75 %
deviation		2.25 %		4.54 %		4.15 %



(a) (b)

Fig. 3. (a) The number of neighbours has more influence on the complex scenes for which an optimal value can be found with roughly 15 neighbours. (b) A feature selection rate between a third and a half of the features allows to obtain the best error rates.

5 Conclusion

In this article we have presented a new approach for image classification, which is based on a graph representation of the images. The classifier is a k-Nearest-Neighbour algorithm and uses a graph edit distance for which we have evaluated different sets of edit costs to find the most appropriate one for image analysis.

Furthermore, we have shown that a region selection by maximisation of the mutual information between the region types and the class to predict greatly improves the recognition rates while reducing the complexity of the graph matching. This allows the classifier to offer competitive computing times.

Other existing methods in the literature stress different features in the image. For instance [1] or [9] lead to more or less comparable results, but what is more, our method performs better on the type of scenes that are difficult for them. Further work will investigate how our approach can be combined with these ones to achieve a better overall performance.

Table 2. Influence of the feature selection on the error rates.

keyword	with all regions		with region selection	
	training error	test error	training error	test error
snowy	8.4 %	11.4 %	11.1 %	10.9 %
country	14.5 %	16.3 %	14.5 %	12.4 %
people	12.8 %	15.6 %	12.7 %	10.4 %
streets	14.6 %	17.3 %	17.1 %	14.6 %

Table 3. Influence of the feature selection on the computational costs.

	nodes	graph distance	classif.
all features	5.6	74.8 ms	7659.1 ms
66% features	3.9	5.1 ms	497.0 ms
50% features	3.0	1.6 ms	117.8 ms
33% features	2.3	0.3 ms	47.6 ms

References

1. Chapelle, O., Haffner, P., Vapnik, V.: SVMs for histogram-based image classification. IEEE Transactions on Neural Networks **10** (1999) 1055–1065
2. Li, J., Wang, J.Z.: Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Trans. PAMI **25** (2003) 1075–1088
3. Beretti, S., Del Bimbo, A., Vicario, E.: Efficient matching and indexing of graph models in content-based retrieval. IEEE Trans. PAMI **23** (2001) 1089–1105
4. Minka, T., Picard, R.: Interactive learning using a society of models. Pattern Recognition **30** (1997) 565–581
5. Comaniciu, D., Meer, P.: Robust analysis of feature spaces: Color image segmentation. In: Proceedings of CVPR, San Juan, Porto Rico (1997) 750–755
6. Le Saux, B., Boujemaa, N.: Unsupervised robust clustering for image database categorization. In: Proceedings of ICPR, Quebec, Canada (2002) 259–262
7. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New-York, N.Y. (1981)
8. Guyon, I., Elisseff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3** (2003) 1157–1182
9. Le Saux, B., Amato, G.: Image recognition for digital libraries. In: ACM Multimedia/International Workshop on Multimedia Information Retrieval. (2004)
10. Battiti, R.: Using mutual information for selecting features in supervised neural network learning. Neural Networks **5** (1994) 537–550
11. Gray, R.M.: Entropy and Information Theory. Springer-Verlag, New York, N.Y. (1990)
12. Nilsson, N.J.: Principles of Artificial Intelligence. Tioga, Palo Alto, CA (1980)
13. Messmer, B.: Graph Matching Algorithms and Applications. PhD thesis, University of Bern (1995)
14. Wong, E.: Three-dimensional object recognition by attributed graphs. In Bunke, H., Sanfeliu, A., eds.: Synctatic and Structural Pattern Recognition - Theory and Applications. (1990) 381–314