Saliency-based Models of Image Content and their Application to Auto-Annotation by Semantic Propagation

Jonathon S. Hare and Paul H. Lewis

School of Electronics and Computer Science,
University of Southampton, Southampton, SO17 1BJ, United Kingdom
{jsh02r, phl}@ecs.soton.ac.uk

Abstract. In this paper, we propose a model of automatic image annotation based on propagation of keywords. The model works on the premise that visually similar image content is likely to have similar semantic content. Image content is extracted using local descriptors at salient points within the image and quantising the feature-vectors into visual terms. The visual terms for each image are modelled using techniques taken from the information retrieval community. The modelled information from an unlabelled query image is compared to the models of a corpus of labelled images and labels are propagated from the most similar labelled images to the query image.

1 Introduction

Searching an image collection can be made intuitive when adequate annotations are available. The keyword terms used for annotation are inherently semantic. By performing text query searches using standard techniques against the keyword terms, images can be found in a manner that will satisfy many users. Of course, this technique can also be combined with visual content search techniques to give the user much more control over the search.

Previous approaches to automatic image annotation have tended to use region-based image descriptions, typically generated by automatic segmentation or through fixed, usually rectangular, shapes. Rectangular regions are a poor choice for image description because they are not be robust to a variety of transformations, such as image rotation. The segmentation approach has a large problem - that of how to perform the segmentation. Over the years many techniques for performing image segmentation have been suggested, although none can really solve the problem of linking the segmented region to the actual object that is being described. Indeed, this shows that the non-naive segmentation problem is not just a bottom-up image processing problem, but also a top-down problem that requires prior knowledge of the true object, before it can be successfully segmented.

Auto-annotation of images has previously been addressed in two separate ways. The first approach has been to define annotation as an unsupervised statistical inference problem. Statistical links between regions and words are discovered by estimating the joint probability distribution between regional image features and words [1].

The second approach clearly separates the textual annotations from the image features, and works by comparing the image similarity at a purely visual level. The approach is basically a supervised learning task. A set of labelled training images is used to associate image features with words, and annotation can take place by comparing visual features, and *propagating* words [2].

In this paper, we propose an approach to image auto-annotation that is not region-based, but instead uses salient interest points. We use a vector-space representation of the local descriptors of salient regions to describe the image in an invariant manner, and a method of semantic propagation to generate the correct annotations for the image.

2 Techniques for Modelling Textual Information

Recent work by Hare and Lewis [3], Sivic and Zisserman [4] and slightly earlier work by Westmacott and Lewis [5], showed a new approach to object matching within images and video footage based on an analogy with classical text retrieval using a vector-space model. This section of the paper briefly describes two models of information; the vector-space model, and a second related model of information called Latent Semantic Indexing or Latent Semantic Analysis [6].

2.1 The Classical Vector-Space Model

Most classical text retrieval systems work in the same general way, by representing a document and query as a set of terms. These terms are represented as axes in a vector-space, using weighted term frequency as the distance along the axis corresponding to that term.

The vector-space model works by first parsing the documents into individual terms. These terms then undergo a process called *stemming*. Words with common stems often have similar meanings. Each of the stemmed words are then represented by a unique identifier for that word. The number of occurrences of each word in the document is counted and a vector of word-frequencies is created to represent the document. The word-frequency vector often has a weighting applied to it. In the vector-space model, documents, V_q and V_d, can be considered to be similar if the angle between their vectors is small. The normalised scalar product (cosine of angle) is used to measure similarity: $\cos(\theta) = \frac{V_q \bullet V_d}{|V_q||V_d|}$. A cosine similarity of 1 implies that the documents are identical, and a similarity of 0 implies they are unrelated.

2.2 The Latent Semantic Indexing Model

The classical approach to modelling text described above depends on a lexical match between the words in the documents for them to be considered similar.

However, there is often diversity in the words used to describe a document, making the lexical methods incomplete and imprecise. Some words can be interchanged in the same context (*synonomy*), and words often have multiple meaning (*polysemy*). Deerwester *et al* [6] suggest that it is possible to take advantage of the implicit higher-order structure in the association of terms with documents by determining the singular value decomposition (SVD) of large sparse term by document matrices. Terms and documents represented by the k largest singular vectors are then matched against user queries. Deerwester calls this retrieval method Latent Semantic Indexing (LSI) because the k subspace represents important associative relationships between terms and documents that are not necessarily evident in individual documents. Comprehensive details of the LSI process can be found in [6].

3 Representing Images using the Textual Information Models

Salient Regions. In previous work, it has been shown that content-based retrieval based on salient interest points and regions performs much better than global image descriptors [7, 8]. For our algorithm, we select salient regions using the method described by Lowe [9], where scale-space peaks are detected in a multi-scale difference-of-Gaussian pyramid. Peaks in a difference-of-Gaussian pyramid have been shown to provide the most stable interest regions when compared to a range of other interest point detectors [7, 10].

Local Feature Descriptors. There are a large number of different types of feature descriptors that have been suggested for describing the local image content within a salient region; For example colour moments and Gabor texture descriptors [8]. The choice of local descriptor is in many respects dependent on the actual application of the retrieval system; for example some applications may require colour, others may not. In the current implementation of the algorithm, Lowe's SIFT (Scale Invariant Feature Transform) descriptor [9] is used. The SIFT descriptor was shown to be superior to other descriptors found in the literature [11], such as the response of steerable filters or orthogonal filters. The performance of the SIFT descriptor is enhanced because it was designed to be invariant to small shifts in the position of the salient region, as might happen in the presence of imaging noise.

Creating Visual Terms. One immediately obvious problem with taking local descriptors to represent words is that, depending on the descriptor, there is a possibility that two very similar image patches will have slightly different descriptors, and thus there is a possibility of having a massive vocabulary of words to describe the image. A standard way to get around this problem is to apply vector quantisation to the descriptors to quantise them into a known set of descriptors. This known set of descriptors then forms the vocabulary of 'visual' terms that describe the image. This process is essentially the equivalent of the stemming, where the vocabulary consists of all the possible stems. The next problem is that of how to design a vector quantiser. Sivic and Zisserman [4] selected a set of video frames from which to train their vector quantiser, and used the k-means clustering algorithm to find clusters of local descriptors within the training set of frames. The centroids of these clusters become the ‘visual’ words representing the entire possible vocabulary. The vector quantiser then proceeded by assigning local descriptors to the closest cluster.

In this work, a similar approach was used. A sample set of images from the data-set was chosen at random, and feature vectors were generated about each salient region in all the training images. Clustering of these feature descriptors was then performed using the batch k-means clustering algorithm with random start points in order to build a vocabulary of ‘visual’ words. Each image in the entire data-set then had its feature vectors quantised by assigning the feature vector to the closest cluster.

4 Image Auto-Annotation

In this preliminary work, we only look at a very simple method of propagating semantics based on the similarity rank of matching images. The basic idea is intuitive; images that are similar should have similar meaning or semantics.

Using the two models of textual information described in section 2 and applying them to image content as described in section 3, we have all the tools needed to compare and rank documents based on their visual content. By creating a collection or corpus of pre-annotated images, it should be possible to label unannotated images by looking for similar annotated ones. In our simple model of annotation, we just apply, or propagate the labels from the closest M matching images to the unannotated query image.

5 Results and Discussion

Image Dataset. The University of Washington Ground Truth Image Database [12] contains 697 public-domain images that have been semantically marked-up. For example an image may have a number of labels describing the image content, such as “trees”, “bushes”, “clear sky”, etc. We have processed the labels to correct mistakes and fold together terms by merging plurals into singular form (i.e. “trees” became “tree”). The original 287 keywords became 170 terms with these modifications. The average number of keywords per image is 4.8. The empirical keyword distribution across the dataset is shown in Figure 1. For experimentation, the dataset was randomly split into two parts, with one part used for training, and one part used for testing.

Performance Evaluation. Many different measures could be chosen for evaluating the performance of an auto-annotation algorithm, but a number of factors need to be accounted for when choosing a measure. Firstly, the statistics of the

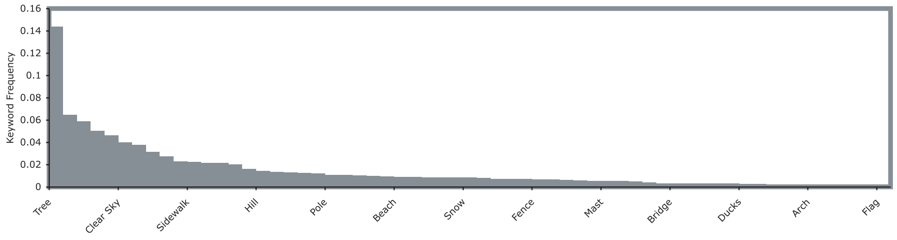

Fig. 1. Plot of empirical keyword distribution in the dataset

vocabulary have to be taken into account. Figure 1 shows the empirical distribution of keywords in the dataset. Because words like 'Tree' occur more often, they are much safer guesses when determining annotations. An auto-annotation technique should therefore perform better than a technique that pseudo-randomly applies labels based on the empirical distribution of keywords in the training set.

Secondly, the training dataset itself might not contain *correct* keywords for some of its images. For comparative purposes, this is not a problem because all of the algorithms have to deal with the same data, however, in an absolute sense, the reported performance is likely to be worse than if perfect data were used.

Thirdly, the performance measure needs to account for the number of incorrect words. An ideal auto-annotation system should choose the correct number of keywords required to describe the image content. Barnard *et al* [1], suggest the use of the *normalised score* measure, $E_{NS}^{(model)} = \frac{r}{n} - \frac{w}{N-n}$, where r is the number of correctly predicted words, n is the actual number of keywords in the query image, w is the number of wrongly predicted words, and N denotes the number of words in the vocabulary. The score gives a value of 1 if the image is annotated exactly correctly, a value of 0 for predicting both everything or nothing, and -1 if the exact complement of the actual word set is predicted. The use of the normalised score is not without problems however. If we are to believe that the measure used should choose the correct number of keywords, then the normalised score is not a good measure as it does not sufficiently weight incorrect guesses. For example, Monay and Gatica-Perez [2] report that in their test database, with an average of 18.5 keywords per image, the normalised score is maximised when their annotation algorithms return about 40 keywords per image. This implies that even if the annotation algorithm is selecting all of the correct labels, it is selecting even more incorrect ones, thus making for very noisy annotations.

In order to address this problem, we have chosen to use precision and recall as our measures for evaluation, although we do also include the normalised score for comparison. Using the same terminology as above, precision and recall are defined as:

$$Recall = r/n,\ Precision = r/(r+w)\ . \tag{1}$$

The interpretation of the precision and recall measures for evaluation of auto-annotation are a little different from the evaluation of retrieval systems. In retrieval, the aim is to get a high precision for all values of recall. However in annotation, the aim is to get both high precision (high proportion of correctly guessed labels to the number guessed) and high recall (high overall proportion of correct labels).

Experimental Results. A number of experiments were performed to ascertain the performance of the two annotation methods and also to provide comparison of their performance against annotation using randomly selected labels, and labels selected based on the empirical frequency distribution in Figure 1. The experiments were performed using a randomly selected 50 : 50 mix of images from the dataset to provide a set of training images and a set of query images. The number of visual terms was set to 3000 [13]. The word-occurrence vectors for both the vector-space and LSI models were unweighted. The optimal number of dimensions of the semantic space, K, for the LSI model was found to be about 40 with respect to maximising the precision, recall, and normalised score.

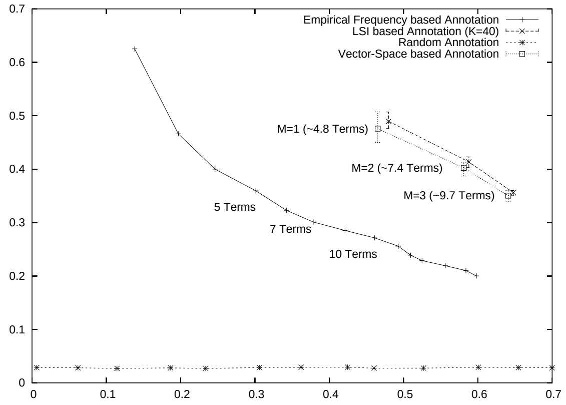

Fig. 2. Precision-Recall curves for each of the auto-annotation methods. Error bars show range of precision over repeated runs

Figure 2 shows the precision-recall curves for each of the annotation methods and the results are summarised in Table 1. The precision-recall curves for the LSI and Vector Space models were generated by increasing the number of images considered for the annotation propagation, M. As would be expected, as M increases, recall also increases due to the increasing number of correctly predicted terms, but precision decreases due to the increased number of incorrect predictions. The curves for the random and frequency distribution based methods were generated by choosing increasing numbers of keywords for annotation. Figure 3 shows some example images together with their true and estimated annotations.

The results clearly show that auto-annotation by simple keyword propagation outperforms choosing labels by choosing words based on the frequency distribution of terms. In addition, the LSI based model marginally outperforms the straight vector-space model.

Table 1. Summary of Results

Method	M	Number of Words	Precision	Recall	E_{NS}
Vector-Space	1	~ 4.8	0.476	0.465	0.450
	2	~ 7.42	0.402	0.581	0.554
	3	~ 9.70	0.350	0.641	0.602
LSI (K=40)	1	~ 4.8	0.490	0.480	0.466
	2	~ 7.42	0.414	0.588	0.561
	3	~ 9.70	0.356	0.648	0.609
Empirical	1	-	0.329	0.343	0.323
	2	-	0.288	0.425	0.394
	3	-	0.241	0.509	0.463
Random	1	-	0.028	0.031	0.001
	2	-	0.026	0.037	-0.004
	3	-	0.029	0.063	0.004

6 Conclusions and Future Work

The results shown in the previous section show promise for our relatively simple auto-annotation approach. The results show that both the vector-space and LSI based annotation algorithms are much better than by just picking keywords based on the empirical frequency distribution. In addition, the LSI based method performs marginally better than the plain vector-space approach. This confirms the findings of [13] which showed LSI based retrieval outperforms vector-space retrieval using a similar method.

The current approach is in some ways deficient because it is unable to select individual terms. This needs to be addressed in future work. Also, the actual cosine distance between matching images has not been taken into account, and better models that actually take into account the distance when selecting which images to propagate labels from can likely be chosen. We also need to consider more powerful image descriptors, such as ones that include colour.

7 Acknowledgements

We are grateful to the EPSRC and Motorola UK Research Laboratory for their support of this work.

References

[1] Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. J. Mach. Learn. Res. **3** (2003) 1107–1135

[2] Monay, F., Gatica-Perez, D.: On image auto-annotation with latent space models. In: MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia, ACM Press (2003) 275–278

	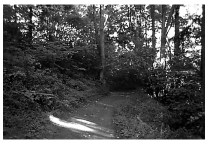	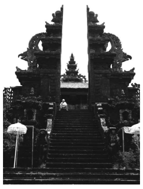	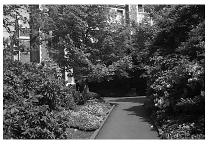
True Annotations	Tree, Bush, Sidewalk	Temple, Sky	Flower, Bush, Tree, Sidewalk, Building
Empirical Annotations	Tree, Building, People, Bush, Grass	Tree, Building, People Bush, Grass	Tree, Building, People, Bush, Grass
Vector-Space Annotations	Tree, Bush	Tree, Building, Grass, Sidewalk, Pole, People, Clear Sky	Flower, Bush, Tree, Building, Partially Cloudy Sky
LSI Annotations	Tree, Bush, Grass, Sidewalk	Steps, Wall	Flower, Bush, Tree, Ground

Fig. 3. Example Annotations

[3] Hare, J.S., Lewis, P.H.: Content-based image retrieval using a mobile device as a novel interface. In Lienhart, R.W., Babaguchi, N., Chang, E.Y., eds.: Proceedings of Storage and Retrieval Methods and Applications for Multimedia 2005, San Jose, California, USA, SPIE (2005) 64–75

[4] Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: International Conference on Computer Vision. (2003) 1470–1477

[5] Westmacott, M., Lewis, P.: An inverted index for image retrieval using colour pair feature terms. In: Proceedings of the SPIE Image and Video Communications and Processing Conference. (2003) 881–889

[6] Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41** (1990) 391–407

[7] Hare, J.S., Lewis, P.H.: Salient regions for query by image content. In Enser, P., Kompatsiaris, Y., O'Conner, N.E., Smeaton, A.F., Smeulders, A.W.M., eds.: Image and Video Retrieval: Third International Conference, CIVR 2004, Dublin, Ireland, Springer (2004) 317–325

[8] Sebe, N., Tian, Q., Loupias, E., Lew, M., Huang, T.: Evaluation of salient point techniques. Image and Vision Computing **21** (2003) 1087–1095

[9] Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110

[10] Mikolajczyk, K.: Detection of local features invariant to affine transformations. PhD thesis, Institut National Polytechnique de Grenoble, France (2002)

[11] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: International Conference on Computer Vision & Pattern Recognition. Volume 2. (2003) 257–263

[12] University of Washington: Ground Truth Image Database. http://www.cs.washington.edu/research/imagedatabase/groundtruth/ (2004)

[13] Hare, J.S., Lewis, P.H.: On image retrieval using salient regions with vector-spaces and latent semantics. To appear in Image and Video Retrieval: Third International Conference, CIVR 2005. Singapore. Springer (2005)