I2R ImageCLEF Photo Annotation 2009 Working Notes

Jiquan Ngiam and Hanlin Goh
Institute for Infocomm Research, Singapore, 1 Fusionopolis Way, Singapore 138632
{jqngiam, hlgoh}@i2r.a-star.edu.sg

Abstract

This paper describes the method that was used for our two submission runs for the ImageCLEF Photo Annotation task.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 [**Database Managment**]: Languages—*Query Languages*

General Terms

Measurement, Performance, Experimentation

Keywords

Photo annotation, Global and local features, Support Vector Machines (SVM), feature selection

1 Introduction

The ImageCLEF Photo Annotation 2009 task involved 53 concepts spanning from abstract concepts (aesthetics, blur) to visual elements (trees, people). Although an ontology was provided, our method did not rely on the ontology heavily, except for handling disjoint cases.

Our method follows the framework in [9] , involving Support Vector Machines and extended Gaussian Kernels over the χ^2 distance. We use a variety of global features, novel local region selectors and simple greedy feature selection.

2 Image Features

2.1 Global Features

The following global features were each computed over the entire image. Each feature essential provides a histogram for the image.

In features where a quantized HSV space was used, the following quantization parameters were employed: 12 Hue Bins, 3 Saturation Bins, 3 Value Bins. Bins were of equal width in each dimension. This results in a total of 108 bins. The choice of these parameters was motivated by [6]

2.1.1 HSV Histogram - 108 dim

The quantized HSV histogram (as above) was used as a feature vector.

2.1.2 Color Auto Correlogram (CAC) - 432 dim

We computed the CAC over a quantized HSV space with 4 distances 1,3,5,7. This gives us a feature vector of 108*4 = 432 dimensions.

For each color & distance pair (c, d), we computed the probability of finding the same color at exactly distance d away. Refer to [6] for details.

2.1.3 Color Coherence Vector (CCV) - 216 dim

We computed the CCV over a quantized HSV space. Since there are two states - coherent and incoherent, this gives us a feature vector of 108*2 = 216 dimensions. We set the tau parameter to 1% of the image size. Refer to [7] for details.

2.1.4 Census Transform (CT) - 256 dim

The CT histogram is a simple transformation of each pixel into a 8-bit value based on its 8 surrounding neighbors (two states, either ¿= or ¡ its neighbor). This provides a feature vector of 256 dimensions (histogram of the CT values). Refer to [8] for details.

2.1.5 Edge Orientation Histogram - 37 dim

We used the LTI-Lib's Canny Edge detector to compute the edge orientation histogram. Each pixel is assigned to either an edge (with orientation) or non-edge. Orientations are quantized into 5 degree angle bins, giving a total of 36 bins for 180 degrees. An additional bin is concatenated for non-edges. This gives a final vector of 37 dimensions.

2.1.6 Interest Point Based SIFT - 500 dim

We used the SIFT binary provided by David Lowe [5] to compute SIFT descriptors. The descriptors were quantized into 500 visual words. A visual words dictionary was computed using k-Means clustering.

2.1.7 Densely Sampled SIFT - 1500 dim

We densely sampled SIFT points at 10 pixel spacings, 4 scales (4, 8, 12, 16 px radius) and 1 orientation. The points are similarly quantized into 1500 visual words by k-Means. This scheme follows that of [1].

2.2 Local Region Features

For a number of concepts, the classification problem can be framed in the Multiple Instance Learning (MIL) framework. In essence, a concept (e.g. Mountain) exists *if and only if* a region within the image demonstrates the concept. Hence, one is motivated to consider whether it is possible to improve performance by finding appropriate region(s) to consider in an image.

We define a local region to be a bounding box. Given a bounding box in an image, one can compute image features similar to the global features. Hence, we define a local region feature to be a feature vector that is extracted based only on region in a bounding box. Therefore, the problem of finding local region features is reduced to one of finding good bounding boxes for each image.

2.2.1 Local Region Selectors

To find good local region selectors (bounding boxes), we frame the problem in a MIL setting. In this setting, each image is a bag-of-regions and regions are considered to be true iff they contain the target concept. Furthermore, a bag is true iff it contains a true region.

We used EM-Diverse Density [10] together with Efficient Subwindow Search (ESS) [4] to search for a target concept with good diverse density. We note that since ESS is able to consider all possible rectangular subwindows, the algorithm essentially considers all possible bounding boxes. However, multiple restarts are required since the algorithm is susceptible to local minimas.

For each concept, we learned a local region selector based on the densely sample SIFT features. From each local region, we extract only the *HSV*, *interest point SIFT* and *densely sampled SIFT* histograms. These three histograms form the (concept-specific) local features for each image.

3 Learning and Feature Selection

3.1 Support Vector Machines

For the final classification, we used LIBSVM [2] with probability estimates (provided with the software). Each concept was treated separately as a individual classification task. Following [9], we used extended Gaussian kernels with the χ^2 distance.

$$K(S_i, S_j) = \sum_{f \in features} \frac{1}{\mu_f} \chi^2(f(S_i), f(S_j))$$

Both local and global features were treated in the same manner. μ_f is the average χ^2 distance for a particular feature; we used it to normalize the distances across different features.

We also performed experiments on cost-sensitive SVMs but the results did not vary that much.

3.2 Feature Selection

Unsurprisingly, different features work well with different concepts. To combine different features, one could incorporate weighting into the kernel function. This method is adopted by the INRIA group in their VOC2007 submission [3]. However, learning these weights is non-trivial and one often resorts to ad-hoc methods such as genetic algorithms.

We chose a simpler method for feature selection in which a greedy algorithm is used. Furthermore, we do employ any partial weighting scheme for the features. Using equal error rate (ERR) as out performance measure, the algorithm is described as follows:

Greedy Algorithm

1. F = all global features
2. For each feature $f \in F$: Compute error rate if f is removed
3. Remove the feature which results in best improvement
4. Repeat (2-3) until removing any feature results in worse performance
5. Consider each feature $f \in All\ Features - F$: Compute error rate if f is added
6. Consider each feature $f \in F$: Compute error rate if f is removed
7. Add or remove the feature which gives best improvement
8. Repeat (5-7) until local optima is reached
9. Return F

The appendix contains a list of concepts and the features that were selected for each of them.

3.3 Hierarchical Evaluation

While there was a new hierarchical measure introduced for the task, we did not specifically optimize for it. The lack of the annotator agreement values also made it more difficult to optimize for this new measure.

However, for the classes that were specified as disjoint, we did simple post processing on the probability estimates to ensure that exactly 1 of the concepts is ≥ 0.5. This was achieved by simply moving the probability estimates to $0.5 \pm \epsilon$.

3.4 Observations

We noticed that the SVM was robust to optional concepts and class imbalance. For optional concepts like Sunny, the SVM was able to correctly classify many “unlabelled” data points correctly (these were classified as negative in a validation set).

The local features were useful for the following concepts: Partylife, Snow, Spring, Autumn, No_Visual_Season, Trees, Mountains, Macro, Portrait, Small_Group, Animals, Vehicle, Overall_Quality, Fancy.

4 Submissions and Results

Two final submissions were made for this task. One used all the features available while the other used only the global features. The official results of the two runs in terms of Average Equal Error Rate (Avg. EER), Average Area Under Curve (Avg. AUC), Average Annotation Score with Annotator Agreement (Avg. AS with AA) and Average Annotation Score without Annotator Agreement (Avg. AS without AA) are reported in Table 1.

Based on the Avg. EER measure, our run with all features was ranked 6 out of 74 submitted runs, while the run using only global features was ranked 11. Comparing the best runs from each group, were reported to be third in the list of twenty participating groups. Evaluating our performance based on the Average Annotation Score with and without the use of Annotator Agreement, our runs were ranked second (global features only) and third (all features).

Submission Run	Avg. EER	Avg. AUC	Avg. AS with AA	Avg. AS without AA
All Features (CVIUI2R_22_2_1244628714641.txt)	0.253296	0.813893	0.82751185	0.8080815
Global Features Only (CVIUI2R_22_2_1244629050173.txt)	0.255945	0.811421	0.8276921	0.8082839

Table 1: Results of Runs evaluated with EER, AUC and the hierarchical measures

References

[1] Anna Bosch, Andrew Zisserman, and Xavier Muoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):712–727, 2008.

[2] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. `http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html`.

[4] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[5] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[6] T Ojala, M Rautiainen, E Matinmikko, and M Aittola. Semantic image retrieval with hsv correlograms. In *12th Scandinavian Conference on Image Analysis*, pages 621 – 627, 2001.

[7] Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, pages 65–73, New York, NY, USA, 1996. ACM.

[8] Jianxin Wu and James M. Rehg. Where am i: Place instance and category recognition using spatial pact. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008.

[9] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vision*, 73(2):213–238, 2007.

[10] Qi Zhang and Sally A. Goldman. Em-dd: An improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems*, pages 1073–1080. MIT Press, 2001.

A Appendix: Selected Features

Concept	Features[1]
0	gdsift1500 gsift500 gcac ldsift1500
1	gdsift1500 gsift500 gcac gccv gcthist
2	gsift500 gcac gccv gcthist
3	gdsift1500 gsift500 gcac gcthist
4	gdsift1500 gsift500 gcac ldsift1500
5	gdsift1500 gsift500 gcac ghsv
6	gdsift1500 gsift500 gcac gcthist
7	gdsift1500 gsift500 gcac
8	gdsift1500 gcac gccv ghsv gcthist
9	gdsift1500 gsift500 gcac ghsv gcthist ldsift1500
10	gsift500 gcac gccv ghsv gcthist
11	gdsift1500 gsift500 gcac gccv ghsv lsift500 ldsift1500 gedgehist
12	gdsift1500 gsift500 ghsv gcthist
13	gdsift1500 gsift500 ghsv ldsift1500
14	gdsift1500 gsift500 ghsv gcthist
15	gdsift1500 gsift500 gccv
16	gdsift1500 gsift500 gccv gcthist
17	gdsift1500 gsift500 gcac
18	gdsift1500 gsift500 gcac ghsv gedgehist
19	gdsift1500 gsift500 ldsift1500
20	gdsift1500 gcac
21	gdsift1500 gsift500 gccv
22	gdsift1500 gsift500 gccv gcthist gedgehist
23	gdsift1500 gsift500 gcac gccv
24	gsift500 gcac ghsv gcthist
25	gdsift1500 gcac gccv ghsv gcthist
26	gdsift1500 lsift500 ldsift1500 lhsv
27	gdsift1500 gsift500 gcac
28	gsift500 gcac gccv ghsv gcthist
29	gdsift1500 gsift500 gcac
30	gdsift1500 gsift500 gcac gccv gcthist
31	gdsift1500 gsift500 gcac gccv
32	gdsift1500 gsift500 gcthist
33	gdsift1500 gsift500 gcac gcthist
34	gdsift1500 gccv gcthist lsift500
35	gdsift1500 gsift500 gcac lhsv gedgehist
36	gdsift1500 gsift500 gccv ghsv
37	gdsift1500 gsift500 gccv ghsv
38	gdsift1500 gsift500 gccv
39	gdsift1500 gsift500 gcac
40	gdsift1500 gsift500 gccv ghsv gcthist
41	gdsift1500 gsift500 gcthist
42	gdsift1500 gsift500 gcthist
43	gdsift1500 gsift500 gcac gccv
44	gdsift1500 gsift500 gcthist gedgehist lsift500 lhsv
45	gdsift1500 gsift500 gccv gcthist
46	gdsift1500 gsift500 gcac
47	gdsift1500 gsift500 gcac ldsift1500 gedgehist
48	gdsift1500 gsift500 gcac gccv gcthist gedgehist
49	gdsift1500 gsift500 gcthist ldsift1500
50	gdsift1500 gsift500 gcac gcthist
51	gdsift1500 gsift500 gcac gccv ldsift1500 lsift500
52	gdsift1500 gsift500 gcac gccv ghsv ldsift1500 gedgehist

[1] 'g' indicates global feature and 'l' indicates local feature.