A Improved SML Method for Semantic Auto-annotation of Image

Ouyang Jun-Lin, Zhu Geng-Ming, Wen Xing-Zi, and Zhang Shao-Bo

Hunan University of Science and Technology, Xiang Tan, China, 411201
{Yangjunlin0732,coconut168}@163.com, abcdechowxz@sohu.com

Abstract. Semantic auto-annotation of Image becomes focus in image retrieval Based on content. The two step semantic auto-annotation of image method is proposed. Firstly, a supervised multi-class labeling method (SML) is adopted to coarse annotation for image, then an optimal semantic tag annotation based on Ontology method (OOSTIA) is employed to fine annotate. There are four ways to extend coarse annotation result in the OOSTIA method, which can fully mine ample semantic information in images. The proposed method is compared to others, Experiments result show that the proposed method outperforms others.

Keywords: SML, Ontology, Semantic annotation, Image retrieve.

1 Introduction

Image sematic auto-annotation becomes a focus for image retrieve based on content. Automated image annotation can be divided with respect to the deployed machine learning method into co-courrence models[3], machine translation models[4], classification approaches[1,2], grapic models[7], latent space approaches[5,6] and relevance models[8] ect.

A surpervised multiclass labling(SML)[2] methoed based on without segmentation is proposed. The basic idea is that image is decomposed into a set of overlapping 8*8 regions, extracted with a sliding window that moves by two pixels between consecutive sample, images are simply represented as bags of lacalized feature vectors, a mixture density estimated for each image, and the mixtures(associated with all images annotated) with a common semantic label pooled into a density estimate for the corresponding semantic class. Semantic annotation and retieval are then implemented with a minimum probablitity of error rule, based on these class densities. This kind of method is conceptually simple, computationally efficient, and do not mixture density estimated for each image.

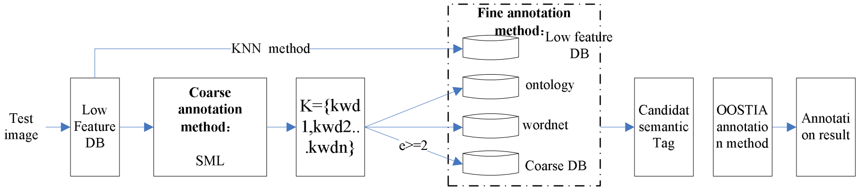

Fig. 1. Two Step Image Semantic Auto-annotation Structure Graphic

G. Yang (Ed.): Proceedings of the ICCEAE2012, AISC 181, pp. 1103–1109.
springerlink.com

By analysing this kind of method, though which get a good efficent in obtaining mostly keywords, a image seems to have thoughts sematic, simple several keywords don't completely describe a image. For example, a image is annotated including {sea, sun, sand}, in fact, mostly semantic isn't annotated such as {ocaan, sunup, sand beach, landscape, scenery, sky, cloud…}. Therefore, a two levels image annotation method is proposed by combining ontology and SML.

2 Surperised Multiclass Labeling (SML) Annotation Methed

2.1 Modling Class without Segmetion

Consider a database T={I_1,…I_N} of images I_i and a semantic vocabulary L={w_1,…w_T} of semantic labels w_i. It is supposed that a set of class-conditional distributions Px|w(x|i), i ∈ {1, ···, T} for the distribution of visual features given the semantic class, it is not difficult to show that both labeling and retrieval can be implemented with a minimum probability of error if the posterior probabilities are available,

$$P_{W|X}(i \mid x) = \frac{P_{X|W}(x \mid i) P_W(i)}{P_X(x)} \tag{1}$$

where Pw(i) is the prior probablity of the ith semantic class. In particular ,given a set of feature vectors X extracted from a (previously unseen) test image I, the minimum probality of an error label for that image is

$$i^*(\chi) = \arg\max_i P_{W|X}(i \mid \chi) \tag{2}$$

2.2 Density Estimation

One efficient alternative to the complexity of model averaging is to adopt a heierarchical density estimation method. This method is based on a mixture hierarchy where children densities consist of different combinations of subsets of the parents' components. In the semantic class densities are their parents. it is possible to estimate the parameters of class mixtures directly from those available for the individual image mixtures, using a two stage procedure. The first stage is the naïve averaging. assuming that each image misture has K components, this leads to a class mixture of DiK components with parameters

$$\left\{\pi_j^k, \mu_j^k, \Sigma_j^k\right\}, j = 1,...Di, k = 1,..., K \tag{3}$$

The second is an extension of EM which clusters the Gaussian components into an M-component mixture, where M is the number of components desired at the class level. Denoting by$\left\{\pi_c^m, \mu_c^m, \Sigma_c^m\right\}, m = 1,..., M$ the parameters of the class mixture, the formula can be found in paper [2].

2.3 Algorithm Description

In this section, we describe the two algorithms used in this work, namely, training and annotation. For the training algorithm, we assume a training set D={(I_1,w_1),…, (I_D,w_D)} of image-caption pairs, where $I_i \in T_D$ with $T_{D=}${I_1,…, I_D},and $w_i \in$L,with L={ $w_{1,\ldots}w_T$}.The steps of training algorithm are:

For each semantic class w∈L,

a. Build a training image set $\bar{T}_D \subset T_D$,where $w \in w_i$ for all $I_i \in \bar{T}_D$,

b. For each image $I_i \in \bar{T}_D$

i. Decompose I into a set of overlapping 8*8regions, extracted with a sliding window that moves by two pixels between consecutive samples (note that, in all experiments reported in this work, images were represented in the YBR color space)

ii. Compute a feature vector , at each location of the tree YBR color channels, by the application of the discrete cosine transform(DCT).Let the image be represented by

$$B = \left\{ \left[x^Y, x^B, x^R \right]_1, \left[x^Y, x^B, x^R \right]_2, \ldots, \left[x^Y, x^B, x^R \right]_M \right\}$$

where $\left[x^Y, x^B, x^R \right]_m$ is the concatenation of DCT vectors extracted from each of the YBR color channels at image location m∈{1,…,M}.Note that the 192-dimensional YBR-DCT vectors are concatenated by interleaving the values of the YBR feature components. This facilitates the application of dimensionality reduction techniques due to the well-known energy compaction properties of the DCT. To simplify notation, we hereafter replace $\left[\mathrm{x}^{\mathrm{Y}}, \mathrm{x}^{\mathrm{B}}, \mathrm{x}^{\mathrm{R}} \right]_{\mathrm{m}}$ with x.

iii. Assuming that the feature vectors extracted from the regions of Image I are sampled independently,find the mixture of eight Gaussians that maximizes their likelihood using the EM algorithm.This produces the following class conditional distribution for each image:

$$P_{X|W}(x \mid I) = \sum_{k=1}^{8} \pi_I^k G(x, \mu_I^k, \sum{}_I^k) \tag{4}$$

where π_I^k 、 μ_I^k 、 $\sum{}_I^k$ are the maximum like-lihood parameters for image I and mixture component k.

c. Fit Gaussian mixture of 64 components by applying the hierarchical EM algorithm of [17-20]to the image-level mixtures of [21].this leads to a conditional distribution for class w of

$$P_{X|W}(x \mid w) = \sum_{k=1}^{64} \pi_w^k G(x, \mu_w^k, \sum\nolimits_w^k) \tag{5}$$

The annotation algorithm processes test images It $I_t \notin T_D$, executing the following steps:

1. step(b-i) of the training algorithm
2. setp(b-ii) of the training algorithm.
3. For each class wi∈L,compute

$$\log P_{W|X}(w_i \mid B) = \log P_{X|W}(B \mid w_i) + \log P_W(w_i) - \log P_X(B) \tag{6}$$

Where B is the set of DCT features extracted form image I_t,

$$\log P_{X|W}(B \mid w_i) = \sum_{x \in B} \log P_{X|W}(x \mid w_i) \tag{7}$$

$P_W(w_i)$ is computed from the training set as the proportion of images containing annotation wi ,and $P_X(B)$ is a constant in the computation above across different wi∈L.

4. Annotate the test image with the five classes wi of largest posterior probablility , $\log P_{W|X}(w_i \mid B)$.

3 Semantic Image Annotation Method Based on ontology

By using the result above method to coause annotation for image, in this section,a optimal semantic tag annotation based on Ontology method (OOSTIA) is employed to fine annotate. A ontology dimension D_h, h=1,…,M, is a tree-structured, also called semantic tags. More precisely, a semantic tag st_j is a path in D_h , $st_j = n_0 / n_1 ... / n_k \in D_h$, where each n_i is a node of the ontology. Each image I_i can be concisely represented as I_i ={ P_i , F_i ,K_i ,ST_i }, P_i is represented to image , F_i is represented to image feature, K_i is represented to keyword based on SML annotation. ST_i is represented to semantic tag of image. In this course of fine annotation include two steps: firstly, produces candidate semantic tag, secondly, organizes the candidates into a candidate tree CT, ranks them, and returns the top-m ones.

3.1 Candidate Semantic Tag Predicting Method

For any couse annotation images, which need to fine annotation, defines relevance keyword variable RK and candidate semantic tag variable CST.There are four ways to produce candidate semantic tag, such as figure 2.

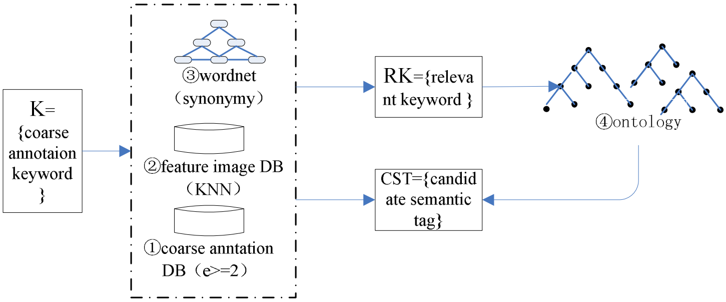

Fig. 2. Four expanded methods of candidate semantic tag

(1) we exploit query keywords K by applying a co-occurrence search on DB image keywords. The search provides a set of images that share at least e terms with K. We rank the iamges on the base of the co-occurrence value and, for the top-p images only, their keywords are added to a set RK of relevant keywords,and all the semantic tags are used to initialize CST. For example, if K={beach , sea , sun}, e=2. and there is an image Ii with K i ={beach, sea, sky}, and STi ={landscape/water/sea}, then sky is added to RK and landscape/water/sea to CST. （2）according to low feature of image F,seach relevant image by using similar degree measure method in the DB , the top-g images which contained at least one keyword is used as candidate image, and the keyword and semantic tag is added to the RK and CST respectively.(3) Expands all relevant keyword RK according to wordnet. For example, if RK has {sea}, synonymous {ocean} is added to RK, and keywords and semantic tag which are contained in the image DB are added to RK and CST respectively.(4) At last, all RK are expanded by domain ontology, and all path contained RK are added to CST.

3.2 Candidate Semantic Tag Annotation Method

The candidate semantic tags CST are bulit into a candidate tree, and then computes the overall top-m as annotation results. Ranking is based on weghts. The weight w_j of st_j is computed as $w_{j=}$ $freq_j * util_j$, where $freq_j$ is the requency of st_j and $util_j$ is so-called utility of st_j wrap all other candidates st_i, defined as:

$$util_j = \sum_{st_i \in ST_h, i \neq j} \frac{len(st_j \cap st_i)}{\max P_h} \tag{8}$$

Where $len(st_j \cap st_i)$ is the length of the common path between st_j and st_i, where $\max P_h$ is the maximum path length within the dimension D_h. Uility measures the amount of overlap between st_j and all other st_i's.

4 Experiment Result

Our experiments is implemented in the environment of Intel-p4 2.8G CPU , XP system. Visual C++6.0 is used as development environment, image DB uses the popular test image DB Corel5K,which contained 5000 images and 50 class, very class contain 100 images. We use 20 image in very class as test example. We compare with four methods: BMRM method [8], information bottleneck cluster annotation method (IBCA)[1], SML[2] and our proposed method (SML-OOSTIA), the annotation result is listed as FIG 3.

Image Method				
BMRM	Elephant tree grass	House grass	Flower leaf rose	Sky sea tree
IBCA	Elephant tree grass	House tree grass	Flower leaf rose	Sky sea tree landscape
SML	Elephant tree grass	House tree grass	Flower leaf rose	Sky sea tree hill landscape
SML-OOSTIA	Elephant animal plant tree grass Branch plain forest landscape	House tree animal plant leaf branch grass plain landscape	Flower leaf rose plant tree branch landscape	Sky universe cloud sea ocean water plant tree branch hill mountain landscape

Fig. 3. Annotation result for four different methods

Fig.4 describe the time performance for the above method. The time complexity can be described by O(DR), where D represent the number of train image, R represent feature of image. the time complexity of SML can be represented by O(TR), where T represent the number of the semantic class of train image. Our method need more time than SML ,but our method is better than MBRM and IBCA.

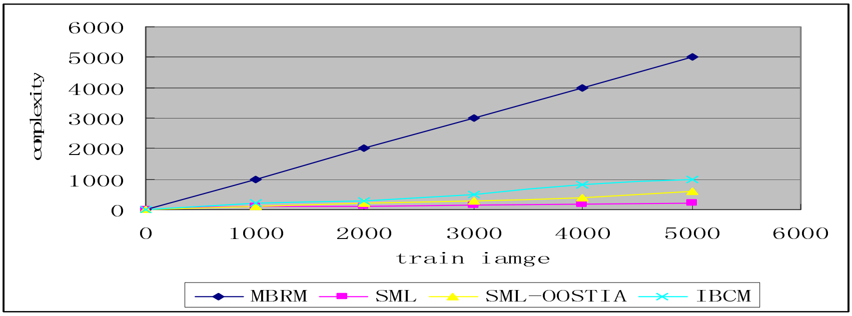

Fig. 4. Performance compared for four different methods

5 Conclusions

A two levels image semantic annotation method is proposed in this paper. Our method compares with other three methods, experiment result show that our method is better than other methods in the whole performance aspect. In the next step, we continue to study the fine annotation methods.

Acknowledgments. This work was supported by scientific research fund of Hunan Provincial Education Department (No:10C0691).

References

[1] Xia, L.-M., Tan, L.-Q., Zhong, H.: Semantic Annotations of Image Based on Information bottleneck method. Pattern Recognition and Artificial Intelligence 21(6), 1199–1205 (2008)

[2] Carneiro, G., Chan, A.B., Moreno, P.J.: Supervised Learning of Semantic Classes for Image Annotation and Retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 394–410 (2007)

[3] Mori, Y., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with word. In: International Workshop on Multimedia Intelligent Storage and Retrieval Management (MISRM) (1999)

[4] Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.: Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)

[5] Monay, F., Gatica-perez, D.: On image auto-annotation with latent space models. In: Proceedings of the 11th International ACM Conference on Multimedia, pp. 275–278. ACM, New York (2003)

[6] Monay, F., Gatica-perez, D.: Plsa-based image auto-annotation:constraining with latent space models. In: Proceedings of the 12th International ACM Conference on Multimedia, pp. 348–351. ACM, New York (2004)

[7] Liu, J., Li, M., Liu, Q., Lu, H., Ma, S.: Image annotation via graph learning. Pattern Recognition 42, 218–228 (2009)

[8] Feng, S.L., Manmatha, R., Lavrenko, V.: Multiple Bernoulli relevance models for image and video annotation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1002–1009 (2004)