Experimental Studies of Visual Models in Automatic Image Annotation

Ping Guo, Tao Wan, and Jin Ma

Image Processing and Pattern Recognition Laboratory,
Beijing Normal University, Beijing 100875, China
pguo@ieee.org

Abstract. Semantic image annotation can be viewed as a mapping procedure from image features to semantic labels, by the steps of image feature extraction and image-semantic mapping. The features can be low-level visual features, such as color, texture, shape, etc., and the semantic labels can be related to the knowledge of human on the image understanding. However, these linear representations are insufficient to describe the complex natural scene. In this paper, we study currently existing visual models that are able to imitate the way the human visual system acts for the tasks of object recognition and scene interpretation. Therefore, it is expected to bring a better understanding to the image visual content in human cortex will. In the experiments, there are three state-of-the-art visual models are investigated for the application of automatic image annotation. The results demonstrate that with our proposed strategy, the annotation accuracy is improved comparing to the most used low-level linear representation features.

Keywords: Automatic image annotation; Semantics; Visual models; Human visual system; Low-level features.

1 Introduction

In computer vision research field, automatic semantic image annotation is attracted many researchers' interest. Semantic image annotation can be viewed as a mapping procedure from image features to semantic labels, by the steps of image feature extraction and image-semantic mapping. When the image annotation is posed as a classification problem, it becomes a mapping problem from low-level visual feature to high-level semantic label. Low-level visual features (color, shape, texture, edges) are easy to be dealt with computer, while high-level semantic labels are related to the knowledge of human on the image understanding.

In visual feature extraction, most studies use linear representations. The methods, such as Gabor transform, discrete cosine transform and wavelet transform, are adopted. But for a complex natural scene, perceptually distinct image regions produce response patterns that are highly overlapping and cannot be easily distinguished using low-level, linear representations [1]. As we know, human visual system outperforms the best machine vision system in any measurements. A fundamental function of the

J.A. Jacko (Ed.): Human-Computer Interaction, Part I, HCII 2011, LNCS 6761, pp. 562–570, 2011.

visual system is to encode the building blocks of natural scenes that subserve visual tasks such as object recognition and scene understanding. Therefore, it is expected to bring us better understanding to image visual feature through emulating object recognition in human cortex will. In this paper, we will experimental study currently existing visual models, and apply to image annotation problem.

In the experiments, three state-of-the-art visual models are investigated for automated image annotation application problem. These models include the model introduced by Serre *et al*. [2], model proposed by Mutch *et al.* [3], and model proposed by Karklin *et al.* [1], we call them Model-A, Model-B, and Model-C, respectively. The performances of three visual models are evaluated by using various natural images. In our proposed strategy, the neural population acts in the visual models are used as image features, and the support vector machine (SVM) classifier is trained for semantic image annotation.

1.1 Related Work

The recognition of different kinds of object categories with respect to illumination conditions, viewpoints from different positions, and diverse backgrounds is one of the major challenges for computer vision [4]. The experiments also show that human brains outperform the best machine vision system in similar cognitive and detective tasks [5]. Therefore, it is essential for researchers to understand how visual cortex recognizes objects as well as to emulate object recognition procedure.

In 1946, Gabor filter has been proposed to justify the sensory coding in the early vision studies [6] [7]. It has been proved to be a good model of cortical simple cell receptive fields, and the filtered results were used to classify images. Another approaches have focused on examining receptive fields of a more complex type [8], such as position or scale invariance, and constructing different models [5] [9] [10] [11], which learn to code the pixel intensities of a patch of texture or edge. However, much of work has concentrated on fitting mathematical models to image data or has been motivated by the specific computational goals [12]. For example, in a standard model of complex cells [10], the "energy" model including two localized and oriented features (typically 90◦ out of phase Gabor functions) is convolved with the image, and the outputs are squared and summed to give the neuron's response. In addition, the model described in [11] is to maximize the sparseness of locally pooled energies that correspond to complex cell outputs to show emergence of a topographic organization.

Another type of vision model stressed importance to a quantitative theory of the ventral stream of visual cortex [1] [13]. They extracted image features much like what the simple and complex cells in human brain would do, and then labeled images with the help of these features. Serre *et al*. [2] have improved a cognitive system that closely followed the organization of visual cortex and build an invariant feature representation by alternating between a template matching and a maximum pooling operation. They made the features complex to scale and position after each layer, and used SVM or boosting as the classifier. Mutch and Lowe [3] introduced an object recognition method based on sparse features with limited receptive fields. They made a few improvements on the Serre's model. In recent years, Karklin and Levicki proposed a new model using complex cell properties [1]. It generalized over similar images,

higher-level visual neurons encode statistical variations that characterize local image regions. The subsequent section will give a brief introduction for these models.

1.2 Visual Model Overview

Serre *et al.* [2] introduced a general framework for the recognition of complex visual scenes, which is motivated by the findings in biology. The key element in the approach is a set of scale and position-tolerant feature detectors, which agree quantitatively with the tuning properties of cells along the ventral stream of visual cortex. These features are able to adaptive to the training set. They demonstrate the strength of the approach on a range of recognition tasks from invariant single object recognition in clutter to multiclass categorization problems and complex scene understanding tasks. They also showed a universal feature set learned from a set of natural image unrelated to any categorization task and achieved good performance.

Mutch and Lowe [3] developed an improved approached based on Serre's model by incorporating sparsity and localized intermediate-level features. They first applied Gabor filters at all positions and scales. Feature complexity and position/scale invariance are then built up by alternating template matching and max pooling operations. Several refinement steps are taken to increase the sparsity by constraining the number of feature inputs, lateral inhibition, and feature selection. The model is a partial implementation of the standard model of object recognition in cortex. The experiments have indicated that these modifications can improve classification performance.

Karklin and Lewicki have proposed a hierarchical model in [14, 15], which not only described sparse marginal densities and magnitude dependencies, but also captured a variety of joint density functions that are consistent with previous observations and theoretical conjectures. They also developed a new vision model based on this hierarchical model [15], which summarized the patterns of correlations for a given type of image using the covariance matrix of the data. In this model, every image patch corresponds to a latent variable, which encodes the image distribution consistent with the input image. An important advantage of this approach is that, it learns a general set of features that are determined by the statistical structures in natural images.

The paper is organized as follows. Section 2 describes the image annotation method using these three visual models. The experimental results and discussions are present in Section 3. The conclusions and feature work are summarized in Section 4.

2 Image Annotation with Visual Models

In this work, we experimental study these three state-of-the-art visual models applied to semantic image annotation problem. The basic idea is to adopt the visual model for image feature extraction, then feed these features to a SVM classifier to implement image annotation. The annotation procedure with these three visual models are described in details in the following subsections.

2.1 Image Annotation Using the Model-A

The detail annotation procedure with Model-A proposed by Serre *et al.* [2] is described as follows:

Step 1: Apply feature extraction method to all training and test images.

- S1 units: Apply a set of Gabor filters to each image. The filters are arranged to form a pyramid of scales, spanning a range of sizes from 7×7 to 37×37 pixels in step of two pixels. To keep the number of units tractable, four orientations (0°, 45°, 90°, and 135°) are considered, thus leading to 64 different filters in total (16 scales × 4 orientations).
- C1 units: C1 units pool over afferent S1 units from the previous layer to get the local maximum value with the same orientation and from the same scale band. The parameter setting can be found in [2]. Each scale band contains two adjacent filter sizes (there are eight scale bands for a total of 16 S1 filter sizes). For instance, scale band 1 contains S1 filters with sizes 7×7 and 9×9. The scale band index of the S1 units also determines the size of the S1 neighborhood NS×NS over which the C1 units pool. Again, this process is performed for each of the four orientations and each scale band independently.
- S2 units: In the S2 layer, units pool over afferent C1 units from a local spatial neighborhood across all four orientations. S2 units behave as radial basis function (RBF) units. Each S2 unit response depends in a Gaussian-like way on the Euclidean distance between a new input and a stored prototype. That is, for an image patch X from the previous C1 layer at a particular scale S, the response r of the corresponding S2 unit is given by:

$$R = \exp(-\beta \|X - P_i\|^2). \tag{1}$$

  In the experiment, we take β as 1/800. P_i is one of the N features (center of the RBF units) learned during training. To initialize the P_i in RBF neural network, a simple sampling process is applied that during training, a large pool of prototypes of various sizes and at random positions are extracted from a target set of images. These prototypes are extracted at the level of the C1 layer across all four orientations, i.e., a patch P_0 of size $n \times n$ contains $n \times n \times 4$ elements.
- C2 units: The element of the final feature vector is computed by taking a global maximum value over all scales and positions for each S2 type over the entire S2 lattice, i.e., the S2 measures the match between a stored prototype Pi and the input image at every position and scale; we only keep the value of the best match and discard the rest. As a result, a feature vector of $N \times 1$ dimension is formed for each image.

Step 2: Apply a SVM classifier with RBF kernel to the feature vectors above to get the separating surfaces, then assign semantic label to corresponding image.

2.2 Image Annotation Using the Model-B

The Model-B [3] is implemented and summarized as follows:

Step 1: For the image layer, we convert the image to grayscale and scale the shorter edge to 140 pixels while maintaining the aspect ratio, and then we create an image pyramid of 10 scales, each a factor of $2^{1/4}$ smaller than the last.

Step 2: For the Gabor filter (S1) layer, the S1 layer is computed from the image layer by centering 2D Gabor filters with a full range of orientations at each possible position and scale. The base model follows [2] and uses 4 orientations. The Gabor filters are 11×11 in size, and are described by

$$G(x, y) = \exp(-(\mathbf{X}^2 + \gamma^2 \mathbf{Y}^2) / (2 \times \delta^2)) \times \cos(2\pi \mathbf{X} / \lambda), \tag{2}$$

where $\mathbf{X} = x \cos \theta - y \sin \theta$ and $\mathbf{Y} = x \sin \theta + y \cos \theta$. x and y vary between -5 and 5, and θ varies between 0 and π. The parameters γ (aspect ratio), δ (effective width), and λ (wavelength) are all taken from [2] and are set to 0.3, 4.5, and 5.6 respectively.

Step 3: Local invariance (C1) layer pools nearby S1 units to create position and scale invariance over larger local regions, and as a result can also subsample S1 to reduce the number of units. For each orientation, the S1 pyramid is convolved with a 3D max filter, 10×10 units across in position and 2 units deep in scale. A C1 unit's value is simply the value of the maximum S1 unit (of that orientation) that falls within the max filter. Due to the pyramidal structure of S1, we are able to use the same size filter for all scales. The resulting C1 layer is smaller in spatial extent and has the same number of feature types (orientations) as S1.

Step 4: In intermediate feature (S2) layer, at every position and scale in the C1 layer, we perform template matches between the patch of C1 units centered at that position/scale and each of d prototype patches. These prototype patches represent the intermediate-level features of the model.

Step 5: In global invariance (C2) layer, we create a *d*-dimensional vector that is classified using the linear SVM classifier.

2.3 Image Annotation Using the Model-C

In Model-C, Karklin assumed that the individual image patch **x** is with multivariate Gaussian probability distributions [1],

$$p(\mathbf{x} | \mathbf{y}) = N(0, \mathbf{C}), \tag{3}$$

where covariance matrix **C** is represented by a set of basis functions $\mathbf{A}_j$,

$$\log \mathbf{C} = \sum y_j \mathbf{A}_j, \tag{4}$$

where $\mathbf{A}_j$ is a symmetric matrix of the same size with the covariance matrix **C**, and can be illustrated as:

$$\mathbf{A}_j = \sum_k w_{jk} \mathbf{b}_{jk} \mathbf{b}_{jk}^T. \tag{5}$$

Every patch has a latent variable y that has a different set of weights w_{jk}, corresponding to an expansion or contraction along vector $\mathbf{b}_{jk}$. Different image patches

correspond to different latent variables y_j, which is regarded as neural population acts. In the experiment, we take y_j as the feature vectors.

There are three main steps to build the Model-C [1]:

Step 1: Image pre-processing is performed to ensure that all the image patches are with Gaussian distribution. To emulate the transformation at the retinal cone cells [14] [15], the original images are transformed to grayscale. Then after pixel intensities log10-transformed, the images are filtered with a Gaussian low-pass filter. Then, the images are down-sampled with the rate of 2:1. Each image in the database is divided into non-overlapping 20×20 image patches. Every patch is transferred into a column vector. The desirable patches are selected as the training data to form a large matrix. The mean luminance value is calculated for the large matrix and subtracted from each patch. This is "whitened" [16] all image patches to remove global correlations and to normalize the variance.

All image patches are formulated in a matrix called ***Pool***, which is a 400×N matrix (N is the number of image patches). Let ***mPool*** be the mean luminance value of ***Pool***, then the covariance **C** is calculated as,

$$\mathbf{C} = \boldsymbol{rePool} * \boldsymbol{rePool'}, \tag{6}$$

where ***rePool*** is the matrix ***Pool*** minus its mean value, and

$$\boldsymbol{rePool} = \boldsymbol{Pool} - \boldsymbol{mPool}. \tag{7}$$

Then the eigenvalues ***EPool*** and eigenvectors ***DPool*** are computed using the covariance matrix. The whitening is done by this,

$$\boldsymbol{whitex} = 1/\mathrm{sqrt}(\boldsymbol{DPool}) \times \mathbf{C}^{-1/2} \times \mathbf{x}, \tag{8}$$

where **x** is a column vector in ***rePool***, and ***whitex*** represents the whitened patch.

Fig. 1. Sample images used in the experiment

Step 2: In the feature extraction stage, the visual model parameters are estimated. After the training process, the latent variable y is computed for every image patch based on the perceptual model in the dataset. These latent variables of image patches are regarded as neuron population acts to objects in image, we take them as feature vectors. The part of feature vectors are used as training data for the SVM classifier and others are the testing data.

Step 3: The SVM classifier is used to perform the annotation task on images.

3 Experimental Results and Discussions

3.1 Summary of the Experiments

The evaluation of an image annotation system requires three components: an image database with manually produced annotations, which is used as training and verification dataset, a strategy to construct annotation system, and a set of measures to verify annotation performance. In the experiment, the image database is consisted of 351 images partly selected from the Caltech 101 object categories. There are 3 classes in the semantic label set, including "Bonsai", "Car side", and "Hksbil". Some images used in the experiments are shown in Fig. 1.

In the experiment, each image is assigned with a caption of only one label. In the Model-A and Model-B, the Gabor filters are applied to each image and it is decomposed into a set of regions. The filters are arranged to form a pyramid of scales, spanning a range of sizes from 7×7 to 37×37 pixels in steps of 2 pixels. To keep the number of units tractable, we consider four orientations (0°, 45°, 90°, and 135°), thus leading to 64 different filters total (16 scales × 4 orientations).

In the Model-C, pixel intensities are log-transformed, corresponding roughly to the transformation at the retinal cone cells, and the images are low-pass filtered. Taking the same experiment setup as described in [15], 20×20 pixel image patches are extracted from the entire dataset without using Gabor filter. The mean luminance value is subtracted from each patch, thus speeding up the model training process without significant influence on the performance. All the image patches are whitened in order to remove global correlations and normalize the variance. This allows the model to encode only the deviations of each image distribution from the global statistics. After the pre-processing approach, the neuron population acts y are calculated by training the model with image patches. In training stage, the values of y are estimated with EM algorithm. With obtained neuron coding y as the input of the SVM classifier, image is annotated using class label of SVM output. In this step, about half of samples are taken as the training samples and the remains as test samples. Finally, the annotation precision of every image concept in the test set is computed as:

$$\text{precision} = w_c / w_{auto}, \tag{9}$$

where w_{auto} is obtained from the annotation system, and w_c is the SVM classifier output.

3.2. Experimental Results and Discussions

Three visual models have been developed and evaluated on a variety of natural images. In Table 1, we can see that for different classes these three visual models produced various results. For example, in some classes such as "Car side", "Hawksbill", some visual models can achieve good results, however, in some other classes like "Bonsai", the performance is not satisfied. It shows that the performance of the visual models is affected by the factors of the complexity of the shape and texture of the target object.

In the "Bonsai" class, the Model-A obtains the better precision rate than the Model-B and Model-C. While for "Car side" class, the Model-B gets the highest annotation precision. From the experiments, it can be found that the Model-B tends to produce more stable results in comparison with the Model-A and the Model-C. The reason lies in that the Model-B uses the Gabor filter with the multiple orientations comparing to the one used in the Model-A with four orientations. There are some additional improvements in the Model-B to further enhance the performance.

It is worth noticing that the Model-C has the highest precision values than the other two models for "Hawksbill" class. This illustrates that Model-C is suitable to encoding those image with rich texture information. Original, the main purpose of the Model-C is to describe the natural scene images with texture features. For those image, such as the in-door images, Model-A is more suitable to encoding the features compare to Model-C.

In general, from experiments it is found that annotation accuracy is improved comparing to the most used low-level linear representation features [17].

Table 1. Annotation precision using three visual models

Concept	Model-A	Model-B	Model-C
Bonsai	57.5	51.0	55.8
Car side	88.0	94.6	58.2
Hawksbill	70.0	52.8	85.0

4 Conclusions and Feature Work

In this paper, we explore three novel visual models that are able to imitate the way of the high-level human perception to interpret the image contents. These three models have been applied to the image annotation problem and show the superior results in comparison with the conventional low-level linear representation features. The proposed methodology could serve as a critical part in a content-based image retrieval system to enhance the search query performance in terms of human judgment. In future, we would like to study most these three visual models with respect to visual descriptors. Also we are interested in investigating the extension of our work to the semantic image retrieval system. Experimental results demonstrate that with our proposed strategy, the annotation accuracy is improved comparing to the most used low-level linear representation features.

Acknowledgments. The research work described in this paper was fully supported by the grants from the National Natural Science Foundation of China (Project No. 90820010, 60911130513). The authors would like to express their thanks for Mr. Rukun Hu for his work in parts of experiment.

References

1. Karklin, Y., Levicki, M.S.: Emergence of Complex Cell Properties by Learning to Generalize in Natural Scenes. Nature 457, 83–86 (2009)
2. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust Object Recognition with Cortex-Like Mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 411–426 (2007)
3. Mutch, J., Lowe, D.G.: Object Class Recognition and Localization Using Sparse Features with Limited Receptive Fields. International Journal of Computer Vision 80, 45–57 (2008)
4. Gabor, D.: Theory of Communication. J. IEE 93, 429–459 (1946)
5. Field, D.: What is the goal of sensory coding? Neural Computation 6, 559–601 (1994)
6. Hubel, D.H., Wiesel, T.N.: Receptive Fields, Binocular Interaction and Functional Architecture in The Cat's Visual Cortex. J. Physiology 160, 106–154 (1962)
7. Cavanaugh, J.R., Bair, W., Movshon, J.A.: Nature and Interaction of Signals from The Receptive Field Center and Surround in Macaque V1 Neurons. J. Neurophysiology 88, 2530–2546 (2002)
8. Heeger, D.J., Simoncelli, E.P., Movshon, J.A.: Computational Methods of Cortical Visual Processing. Proc. Natl Acad. Sci., 623–627 (1996)
9. Hyvarinen, A., Hoyer, P.: A Two-layer Sparse Coding Model Learns Simple and Complex Cell Receptive Fields and Topography from Natural Images. Vision Res. 42, 241–2423 (2001)
10. Hubel, D., Wiesel, T.: Receptive Fields and Functional Architecture in Two Nonstriate Visual Areas of the Cat. J. Neurophysiology 28, 22–289 (1965)
11. Bruce, C., Desimone, R., Gross, C.: Visual Properties of Neurons in a Polysensory Area in the Superior Temporal Sulcus of the Macaque. J. Neurophysiology 46, 369–384 (1981)
12. Comon, P.: Independent Component Analysis, a New Concept? Signal Processing 36, 287–314 (1994)
13. Ullman, S., Vidal-Naquet, M., Sali, E.: Visual Features of Intermediate Complexity and Their Use in Classification. Nature Neuroscience 5, 68–687 (2002)
14. Karklin, Y., Levicki, M.S.: A Hierarchical Bayesian Model for Learning Nonlinear Statistical Regularities in Nonstationary Natural Signals. Neural Computation 17, 397–423 (2005)
15. Karklin, Y.: Hierarchical Statistical Models of Computation in the Visual Cortex. PhD thesis, Carnegie Mellon University (2007)
16. Perlibakas, V.: Distance Measures for PCA-based Face Recognition. Pattern Recognition Letters 25, 711–724 (2004)
17. Hu, R.K., Shao, S., Guo, P.: Investigating Visual Feature Extraction Methods for Image Annotation. In: Proceedings of 2009 IEEE International Conference on Systems, Man, and Cybernetics, pp. 3211–3216 (2009)