Combining Generative And Discriminative Classifiers For Semantic Automatic Image Annotation

Brahim MINAOUI *bra_min@yahoo.fr*
Faculty of Science and Technology,
Computer Science Department,
Sultan Moulay Slimane University.
PO Box. 523, Béni Mellal, Morocco.

Mustapha OUJAOURA *M.Mustapha.Oujaoura@ieee.org*
Faculty of Science and Technology,
Computer Science Department,
Sultan Moulay Slimane University.
PO Box. 523, Béni Mellal, Morocco.

Mohammed FAKIR *fakfad@yahoo.fr*
Faculty of Science and Technology,
Computer Science Department,
Sultan Moulay Slimane University.
PO Box. 523, Béni Mellal, Morocco.

Abstract

The object image annotation problem is basically a classification problem and there are many different modeling approaches for the solution. These approaches can be classified into two main categories such as generative and discriminative. An ideal classifier should combine these two complementary approaches. In this paper, we present a method achieving this combination by using the discriminative power of the neural networks and the generative nature of Bayesian networks. The evaluation of the proposed method on three typical image's database has shown some success in automatic image annotation.

Keywords: Automatic Image Annotation, Discriminative Classifier, Generative Classifier, Neural Networks, Bayesian Networks.

1. INTRODUCTION

Automatic image annotation help to bridge the semantic gap, that exists between low-level visual features and the high-level abstractions perceived by humans, by producing object labels or keyword annotations which are nearer to the high level semantic descriptions needed for good image retrieval.

In order to overcome this semantic gap, a number of current research efforts focus on robust classifiers achieving automatically multi-level image annotation [1-6]. These classifiers can be characterized as generative and discriminative according to whether or not the distribution of the image and labels is modeled.

It was observed that generatively-trained classifiers perform better with very few training examples and provide a principled way of treating missing information, whereas a classifiers trained discriminatively perform better with sufficient training data and provide a flexible decision boundaries [7]. Motivated by these observations, several researchers have proposed a variety of techniques that combine the strengths of these two types of classifiers. These hybrid methods, which have delivered promising results in the domains of object recognition [8-10], scene classification [11-15] and automatic image annotation [16-17], have been explored in different ways: [9] and [11] propose a classifier switching algorithm to select the best classifier (generative or discriminative) for a given dataset and availability of label. [10], [14] and [15] propose a technique for combining the two classifiers based on a continuous class of cost functions that interpolate smoothly between the generative strategy and the discriminative one. [8, 12-13] and [16] propose a hybrid generative-discriminative approach in which the features extracted from a generative model are analyzed by a followed discriminative classifier. [17] devise a hybrid generative-discriminative learning approach that includes a Bayesian Hierarchical model (generative model) trained discriminatively.

In this paper, in an attempt to gain the benefit of both generative and discriminative approaches, we propose an approach which combines in a parallel scheme the Bayesian networks for the generative model and the neural networks for the discriminative classifier to accomplish the task of automatic image annotation. The annotation decision is realized by the vote of combined classifiers. Each classifier votes for a given keyword. The keyword that has the maximum of votes will be considered as the proper keyword for the annotation of an object in a query image.

The rest of paper is organized as follows. The various features used in this study are explained in Section 2. Section 3 presents the Bayesian networks and neural networks classifiers. Section 4 describes the experiences adopted to realize the automatic image annotation using these classifiers. Finally, the conclusion of this work is presented in Section 5.

2. FEATURES EXTRACTION

After dividing the original image into several distinct regions that correspond to objects in a scene by using region growing segmentation algorithm [18], the following descriptors are extracted:

2.1 Color Histogram

Typically, the color of an image is represented through some color model. There exist various color models to describe color information. The more commonly used color models are RGB (red, green, blue), HSV (hue, saturation, value) and Y, Cb, Cr (luminance and chrominance). Thus, the color content is characterized by 3 channels from some color models. In this paper, we used RGB color models. One representation of color image content is by using color histogram. Statistically, it denotes the joint probability of the intensities of the three color channels [19].

Color histogram describes the distribution of colors within a whole or within an interest region of image. The histogram is invariant to rotation, translation and scaling of an object but the histogram does not contain semantic information, and two images with similar color histograms can possess different contents.

The histograms are normally divided into bins to coarsely represent the content and reduce dimensionality of subsequent classification and matching phase. A color histogram H for a given image is defined as a vector by:

$$H = \left\{ h\left[i \in \{1,..., k\}\right] = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \delta\left(f(x, y) - C(i)\right)}{M \times N} \quad and \quad (i-1)\times E\left(\frac{256}{k}\right) \le C(i) < i \times E\left(\frac{256}{k}\right) \right\} \tag{1}$$

Where:

- i represent a color in the color histogram;
- E(x) denotes the integer part of x;
- h[i] is the number of pixel with color i in that image;
- k is the number of bins in the adopted color model;

And δ is the unit pulse defined by:

$$\delta(x) = \begin{cases} 1 & si \quad x = 0 \\ 0 & si \quad x \neq 0 \end{cases} \tag{2}$$

In order to be invariant to scaling change of objects in images of different sizes, color histograms H should be divided by the total number of pixels M x N of an image to have the normalized color histograms.

For a three-channel image, a feature vector is then formed by concatenating the three channel histograms into one vector.

2.2 Legendre Moments

In this paper, the Legendre moments are calculated for each one of the 3 channel in a color image. A feature vector is then formed by concatenating the three channel moments into one vector.

The Legendre moments [20] for a discrete image of M x N pixels with intensity function f(x, y) is the following:

$$L_{pq} = \lambda_{pq} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} P_p(x_i)\, P_q(y_j)\, f(x, y) \tag{3}$$

Where $\lambda_{pq} = \dfrac{(2p+1)(2q+1)}{M \times N}$, xi and yj denote the normalized pixel coordinates in the range of [-1, +1], which are given by:

$$\begin{cases} x_i = \dfrac{2x - (M-1)}{M-1} \\ y_j = \dfrac{2y - (N-1)}{N-1} \end{cases} \tag{4}$$

$P_p(x)$ is the pth-order Legendre polynomial defined by:

$$P_p(x) = \sum_{k=0}^{p} \left\{ \frac{(-1)^{\frac{p-k}{2}} (p+k)!\, x^k}{2^p\, k! \left(\frac{p-k}{2}\right)! \left(\frac{p+k}{2}\right)!} \right\}_{p-k=even} \tag{5}$$

In order to increase the computation speed for calculating Legendre polynomials, we used the recurrent formula of the Legendre polynomials defined by:

$$\begin{cases} P_p(x) = \dfrac{(2p-1)x}{p} P_{p-1}(x) - \dfrac{(p-1)}{p} P_{p-2}(x) \\ P_1(x) = x \quad , \quad P_0(x) = 1 \end{cases} \tag{6}$$

2.3 Texture Descriptors

This Several images have textured patterns. Therefore, the texture descriptor is used as feature extraction method from the segmented image.

The texture descriptor is extracted using the co-occurrence matrix introduced by Haralick in 1973 [21]. So for a color image I of size $N \times N \times 3$ in a color space (C_1, C_2, C_3), for $(k, l) \in [1, \cdots, N]^2$ and $(a, b) \in [1, \cdots, G]^2$, the co-occurrence matrix $M_{k,l}^{C,C'}[I]$ of the two color components $C, C' \in \{C_1, C_2, C_3\}$ from the image I is defined by:

$$M_{k,l}^{C,C'}([I], a, b) = \frac{1}{(N-k)(N-l)} \sum_{i=1}^{N-k} \sum_{j=1}^{N-l} \delta(I(i, j, C) - a, I(i+k, j+l, C') - b) \quad (7)$$

Where δ is the unit pulse defined by:

$$\delta(x, y) = \begin{cases} 1 & if \quad x = y = 0 \\ 0 & else \end{cases} \quad (8)$$

Each image I in a color space (C_1, C_2, C_3) can be characterized by six color co-occurrence matrix:

$M^{C_1,C_1}[I]$, $M^{C_2,C_2}[I]$, $M^{C_3,C_3}[I]$, $M^{C_1,C_2}[I]$, $M^{C_1,C_3}[I]$, $M^{C_2,C_3}[I]$.

Matrix $M^{C_2,C_1}[I]$, $M^{C_3,C_1}[I]$ and $M^{C_3,C_2}[I]$ are not taken into account because they can be deduced respectively by diagonal symmetry from matrix $M^{C_1,C_2}[I]$, $M^{C_1,C_3}[I]$ and $M^{C_2,C_3}[I]$

As they measure local interactions between pixels, they are sensitive to significant differences in spatial resolution between the images. To reduce this sensitivity, it is necessary to normalize these matrices by the total number of the considered co-occurrences matrix:

$$M_{k,l}^{C,C'}([I], a, b) = \frac{M_{k,l}^{C,C'}([I], a, b)}{\sum_{i=0}^{T-1} \sum_{j=0}^{T-1} M_{k,l}^{C,C'}([I], i, j)} \quad (9)$$

Where T is the number of quantization levels of the color components

To reduce the large amount of information of these matrices, the 14 Haralick indices [21] of these matrices are used. There will be then 84 textures attributes for six co-occurrence matrices (14×6).

3. NEURAL NETWORKS AND BAYESIAN NETWORKS CLASSIFIERS

3.1 Neural Networks

Neural networks (or artificial neural networks) learn by experience, generalize from previous experiences to new ones, and can make decisions [22, 23].

A multilayer neural network consists of an input layer including a set of input nodes, one or more hidden layers of nodes, and an output layer of nodes. Fig.1 shows an example of a three layer network used in this paper, having input layer formed by M nodes, one hidden layer formed by L nodes, and output layer formed by N nodes.

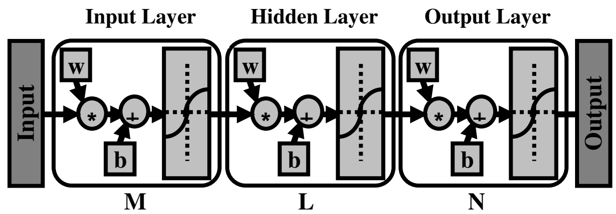


FIGURE 1: The Three Layer Neural Network.

This neural network is trained to classify inputs according to target classes. The training input data are loaded from the reference database while the target data should consist of vectors of all zero values except for a one element, where its index is the class they are to represent. The transfer function used in this tree layer neural network is hyperbolic tangent sigmoid transfer function defined by:

$$f(x) = 2/\left(1+\exp(-2x)\right)-1 \qquad (10)$$

According to authors in [24], the number of neurons in the hidden layer is approximately equal to:

$$L = E\left(1+\sqrt{M(N+2)}\right) \qquad (11)$$

Where:

- E(x) denotes the integer part of x.
- M and N are respectively the number of neurons in the input and output layers.

3.2 Bayesian Networks

The Bayesian networks are based on a probabilistic approach governed by Bayes' rule. The Bayesian approach is then based on the conditional probability that estimates the probability of occurrence of an event assuming that another event is verified. A Bayesian network is a graphical probabilistic model representing the random variable as a directed acyclic graph. It is defined by [25]:

- $G=(X, E)$, Where X is the set of nodes and E is the set of edges, G is a Directed Acyclic Graph (DAG) whose vertices are associated with a set of random variables $X=\{X_1, X_2, \cdots, X_n\}$;
- $\theta=\{P(X_i \mid Pa(X_i))\}$ is a conditional probabilities of each node X_i relative to the state of his parents $Pa(X_i)$ in G.

The graphical part of the Bayesian networks indicates the dependencies between variables and gives a visual representation tool of knowledge more easily understandable by users. Bayesian networks combine qualitative part that are graphs and a quantitative part representing the conditional probabilities associated with each node of the graph with respect to parents [26].
Pearl and all [27] have also shown that Bayesian networks allow to compactly representing the joint probability distribution over all the variables:

$$P(X)=P(X_1, X_2, \cdots, X_n)=\prod_{i=1}^{n} P(X_i \mid Pa(X_i)) \qquad (12)$$

Where $Pa(X_i)$ is the set of parents of node X_i in the graph G of the Bayesian networks.

This joint probability could be actually simplified by the Bayes rule as follows [28]:

$$\begin{aligned} P(X) &= P(X_1, X_2, \cdots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)) \\ &= P(X_n | X_{n-1}, \cdots, X_1) \times P(X_{n-1} | X_{n-2}, \cdots, X_1) \times \cdots \times P(X_2 | X_1) \times P(X_1) \\ &= P(X_1) \times \prod_{i=2}^{n} P(X_i | X_{i-1}, \cdots, X_1) \end{aligned} \tag{13}$$

The construction of a Bayesian network consists in finding a structure or a graph and estimates its parameters by machine learning. In the case of the classification, the Bayesian network can have a class node C_i and many attribute nodes X_j. The naive Bayes classifier is used in this paper due to its robustness and simplicity. The Fig 2 illustrates its graphical structure.

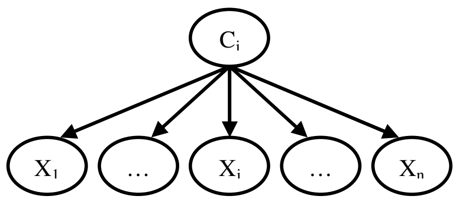


FIGURE 2: Naive Bayes Classifier Structure.

To estimate the Bayesian networks parameters and probabilities, Gaussian distributions are generally used. The conditional distribution of a node relative to its parent is a Gaussian distribution whose mean is a linear combination of the parent's value and whose variance is independent of the parent's value [29]:

$$P\left(X_i = x_i | Pa\ (X_i)\right) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{ -\frac{1}{2\sigma_i^2}\left(x_i - \left(\mu_i + \sum_{j=1}^{n_i} \frac{\sigma_{ij}}{\sigma_j^2}(x_j - \mu_j) \right)\right)^2 \right\} \tag{14}$$

Where,

- $Pa(X_i)$ Are the parents of X_i;
- μ_i , μ_j , σ_i *and* σ_j are respectively the means and variances of the attributes X_i and X_j without considering their parents;
- n_i is the number of parents of X_i;
- σ_{ij} is the regression matrix of weights.

After the parameter and structure learning of a Bayesian networks, The Bayesian inference is used to calculate the probability of any variable in a probabilistic model from the observation of one or more other variables. So, the chosen class C_i is the one that maximizes these probabilities [30]:

$$P\left(C_i \middle| X\right)=\begin{cases} P\left(C_i\right)\prod_{j=1}^{n} P\left(X_j \middle| Pa\left(X_j\right), C_i\right) & \text{if } X_j \text{ has parents .} \\ P\left(C_i\right)\prod_{j=1}^{n} P\left(X_j \middle| C_i\right) & \text{else .} \end{cases} \quad (15)$$

For the naive Bayes classifier, the absence of parents and the variables independence assumption are used to write the posterior probability of each class as given in the following equation [31]:

$$P\left(C_i \middle| X\right)=P\left(C_i\right)\prod_{j=1}^{n} P\left(X_j \middle| C_i\right) \quad (16)$$

Therefore, the decision rule d of an attribute X is given by:

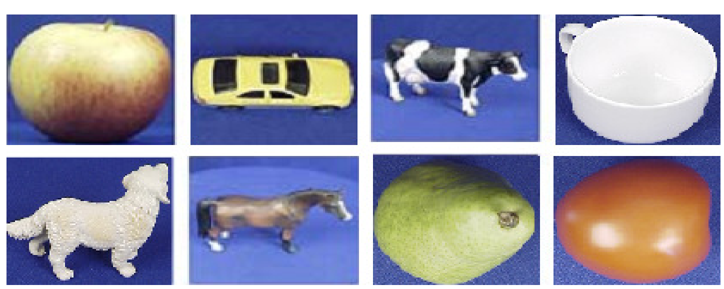
$$d(X)=\underset{C_i}{\arg\max}\ P\left(C_i \middle| X\right)=\underset{C_i}{\arg\max}\ P\left(X \middle| C_i\right)P\left(C_i\right)=\underset{C_i}{\arg\max}\ P\left(C_i\right)\prod_{j=1}^{n} P\left(X_j \middle| C_i\right) \quad (17)$$

The class with maximum probability leads to the suitable keyword for the input image.

4. EXPERIMENTS AND RESULTS

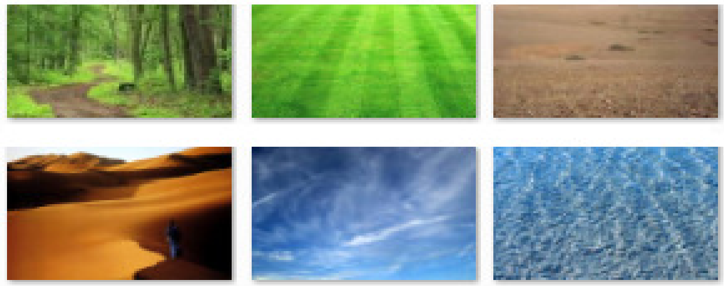
After In this section, we study and compare the performance of discriminative and generative classifiers for automatic image annotation using in first time each classifier alone and in second time the combination of the two different classifiers [31].

In order to achieve this goal, we conduct two experiments on three image databases ETH-80 [32], COL-100 [33] and NATURE created in this work. The Fig.3 shows some examples of image objects from these three image databases used in our experiments.



ETH-80

COIL-100



NATURE

FIGURE 3: Some objects images from ETH-80, COL-100 and NATURE databases.

In the phase of learning and classification, we used a training set of 40 images and a test set of 40 images for each image databases.

In all experiments, the features described in Section 2 are extracted after image segmentation by region growing. For each region that represent an object, 10 components of Legendre moments (L00, L01, L02, L03, L10, L11, L12, L20, L21, L30) and 16 elements for RGB color histograms are extracted from each color plane namely R, G and B. The number of input features extracted using Texture extraction method is 14 Haralick indices multiplied by 6 co-occurrence matrices. This gives 84 textures attributes.

4.1 Experiment 1

In this experience, we provide comparative results of image annotation between the two classifiers: discriminative (neural networks) and generative (Bayesian networks). The experimental method adopted in this experience is represented by the figure 4.

In first time, we have used three neural networks classifiers to annotate images of all databases. Each neural networks, receiving as input one of the three extracted descriptors, votes for a given keyword. The keyword that has the maximum of votes is considered as the proper keyword for the annotation of an object in a query image.

In second time, we repeated the same operation with Bayesian networks classifier as shown in figure 4.

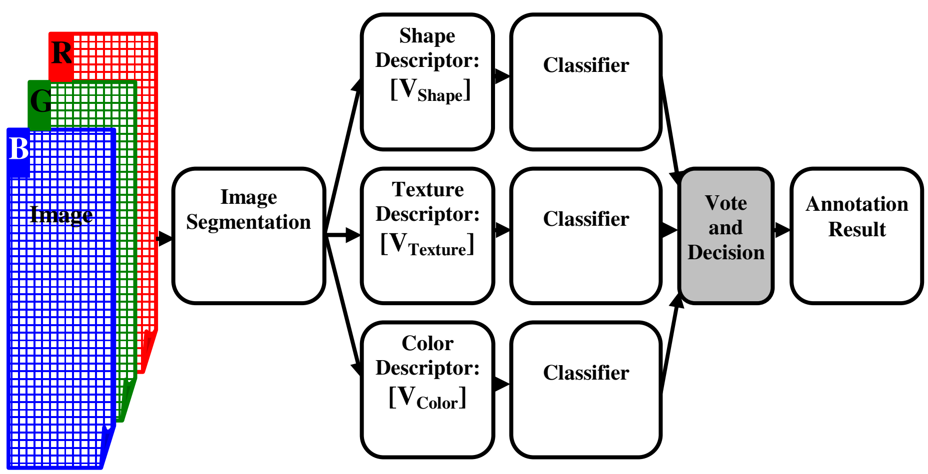


FIGURE 4: Experimental method adopted for image annotation.

4.1.1 Results

Table I summarizes the results of automatic image annotation for each type of classifier and Figures 5,6,7,8, 9 and 10 shows the confusion matrix.

Database	Classification Approach	Average Annotation Rate	Error Rate
ETH-80	neural networks	87.50%	12.50%
	Bayesian networks	90.00%	10.00%
COIL-100	neural networks	82.50%	17.50%
	Bayesian networks	85.00%	15.00%
NATURE	neural networks	90.00%	10.00%
	Bayesian networks	93.33%	6.77%

TABLE 1: Average annotation rate and error rate.

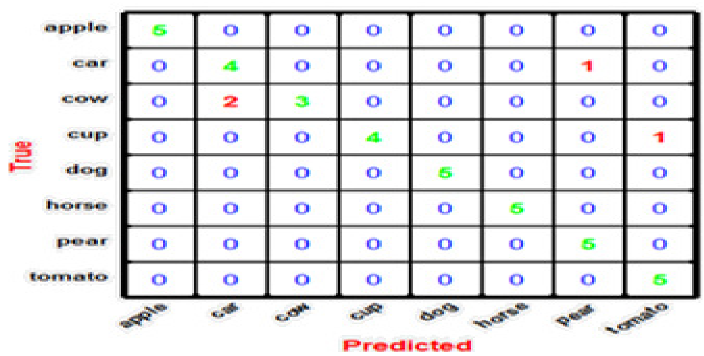


FIGURE 5: Confusion matrix for images of database ETH-80 by using Bayesian networks.

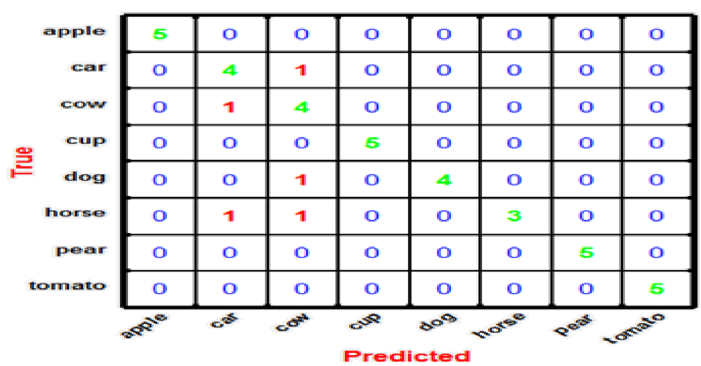


FIGURE 6: Confusion matrix for images of database ETH-80 by using neural networks.

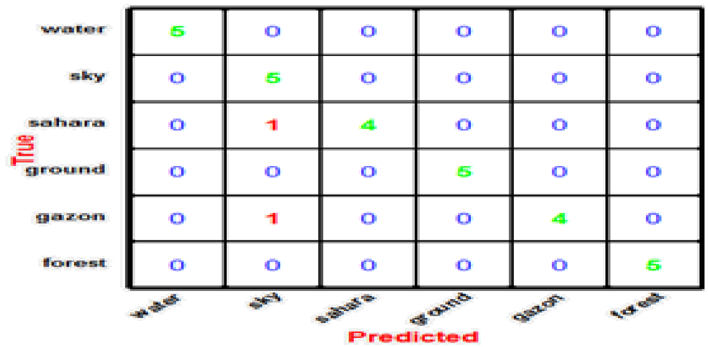


FIGURE 7: Confusion matrix for images of database NATURE by using Bayesian networks.

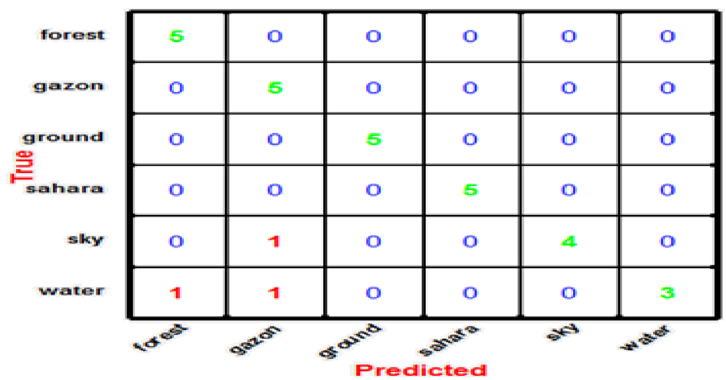


FIGURE 8: Confusion matrix for images of database NATURE by using neural networks.

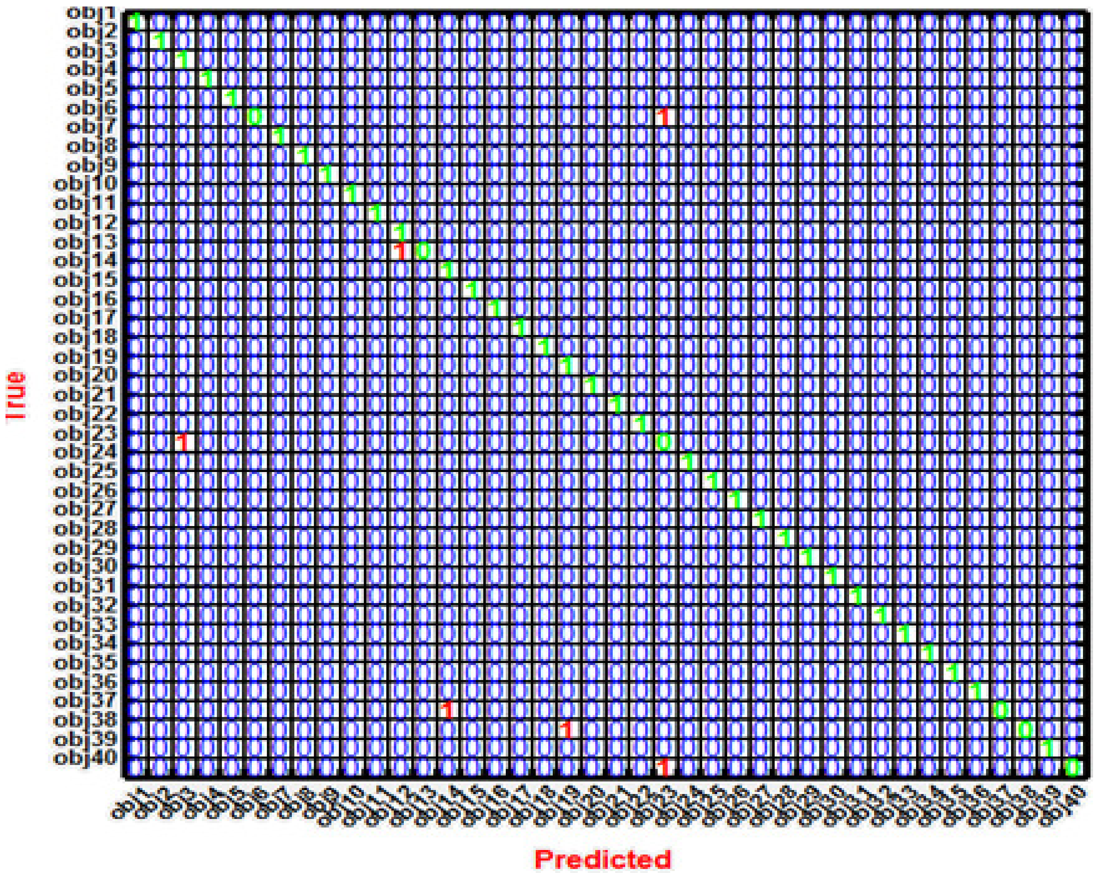


FIGURE 9: Confusion matrix for images of database COIL-100 by using Bayesian networks.

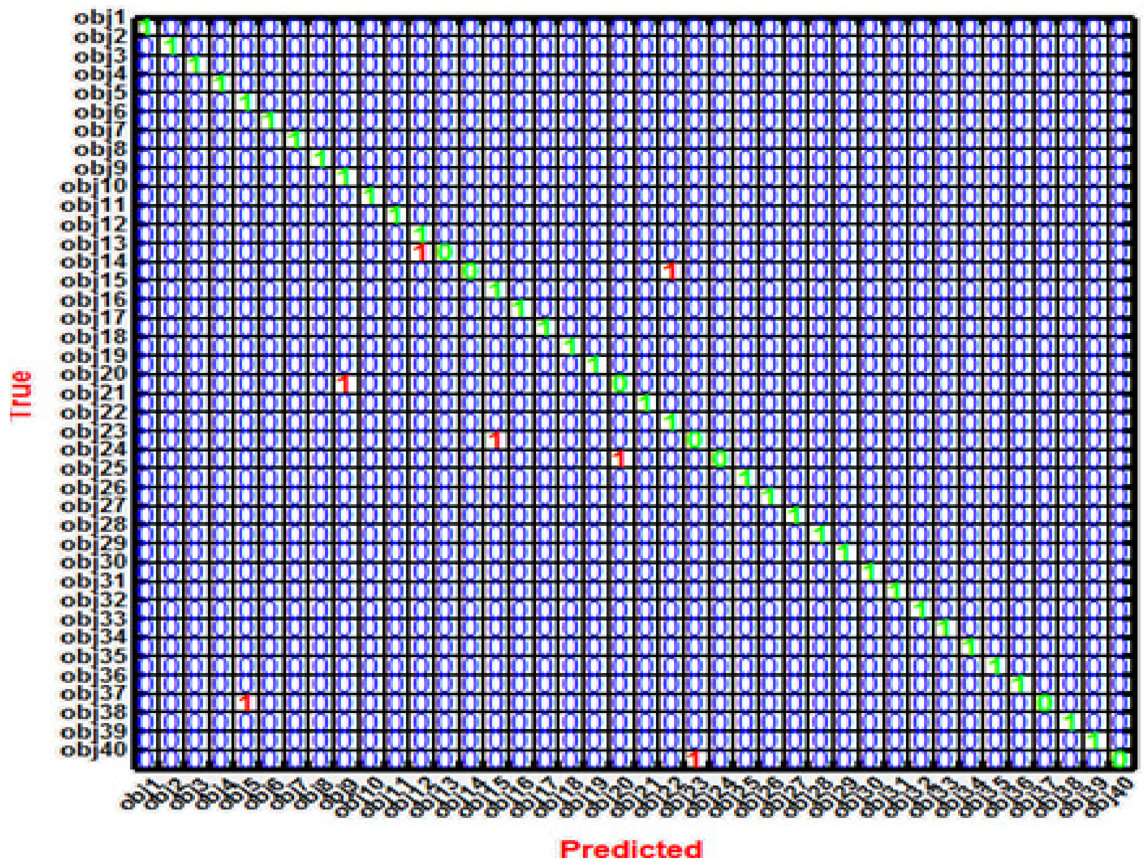


FIGURE 10: Confusion matrix for images of database COIL-100 by using neural networks.

4.2 Analysis of Results

As can be observed from Table 1, Bayesian networks produce the better average annotation rates for all the tree images databases. However, analysis of confusion matrix presented by the Figures 5,6,7,8, 9, 10 shows that the individual annotation rate obtained for some objects (cow, cup, object 6, Sahara and Gazon) with neural networks can be better than those obtained with Bayesian networks. So it appears from these remarks that the combination of these two classifiers will improve the average annotation rates. This constitutes the aim of the experiment 2.

4.3 Experiment 2

Based on the remarks released in the previous two experiments, we combined in this experiment, in addition to descriptors, neural networks and Bayesian networks in order to gain the benefit of the complementarity of these two approaches of classification (discriminative and generative). The principle of this combination is illustrated by the block diagram shown in Fig 11. Thus, with the combination of the three types of descriptors described in Section 2 and the 2 considered types of classifiers, there will be a maximum of votes equal to 3 x 2 = 6. Each classifier with each descriptor votes for a given keyword. The keyword with a maximum of votes will be deemed as the proper keyword for the annotation of an object contained in a query image.

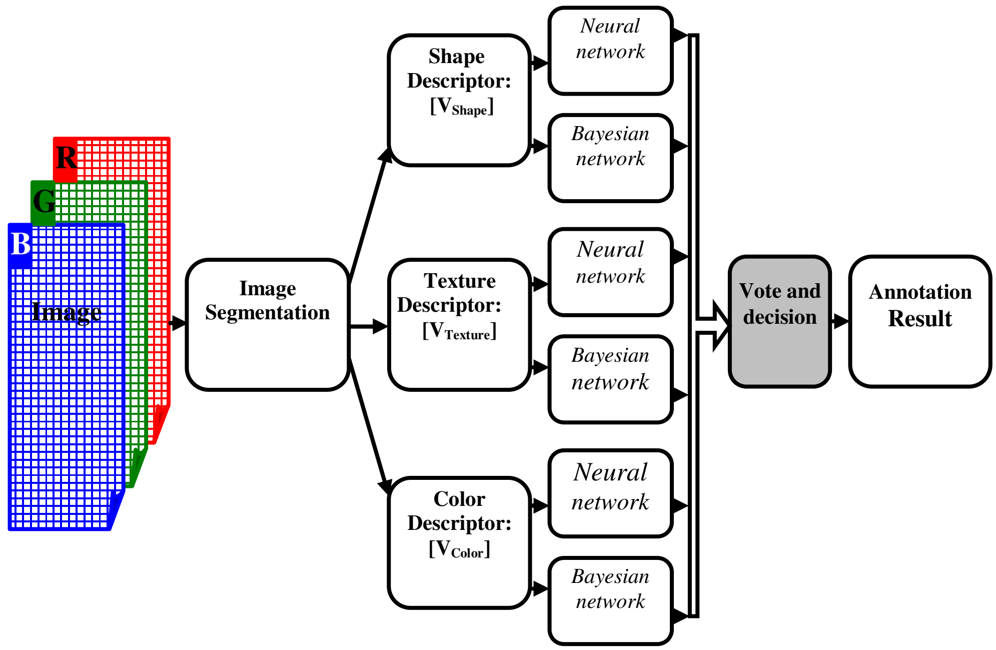


FIGURE 11: Block diagram that illustrates principle of combining discriminative and generative classifiers for automatic image annotation.

4.4 Results

Table 2 shows the average image annotation rate obtained by combining neural networks and Bayesian network classifiers and Figures 12, 13 and 14 shows the confusion matrix.

Database	Average Annotation Rate	Error Rate
ETH-80	92.50%	7.50%
COIL-100	87.50%	12.50%
NATURE	96.67%	3.33%

TABLE 2: Average annotation rate and error rate.

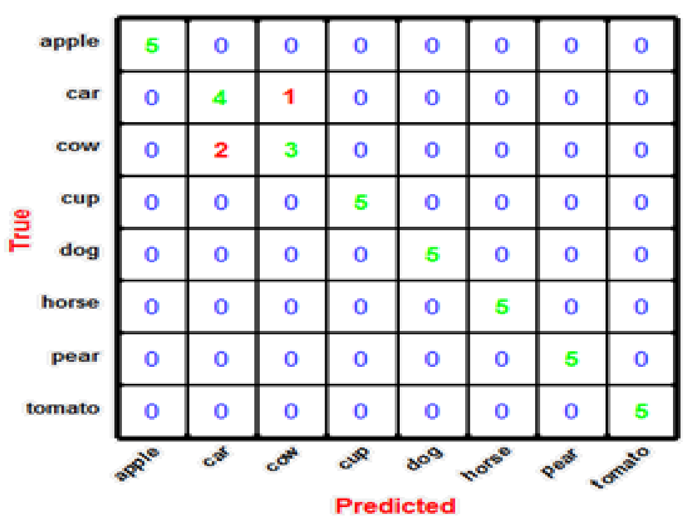


FIGURE 12: Confusion matrix for images of database ETH-80.

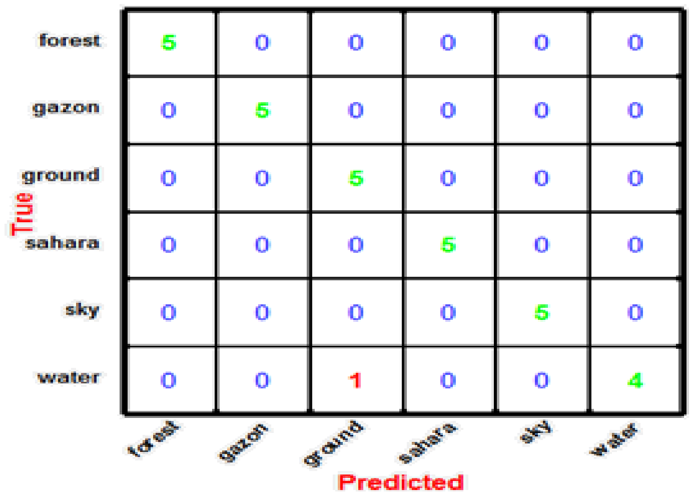


FIGURE 13: Confusion matrix for images of database NATURE.

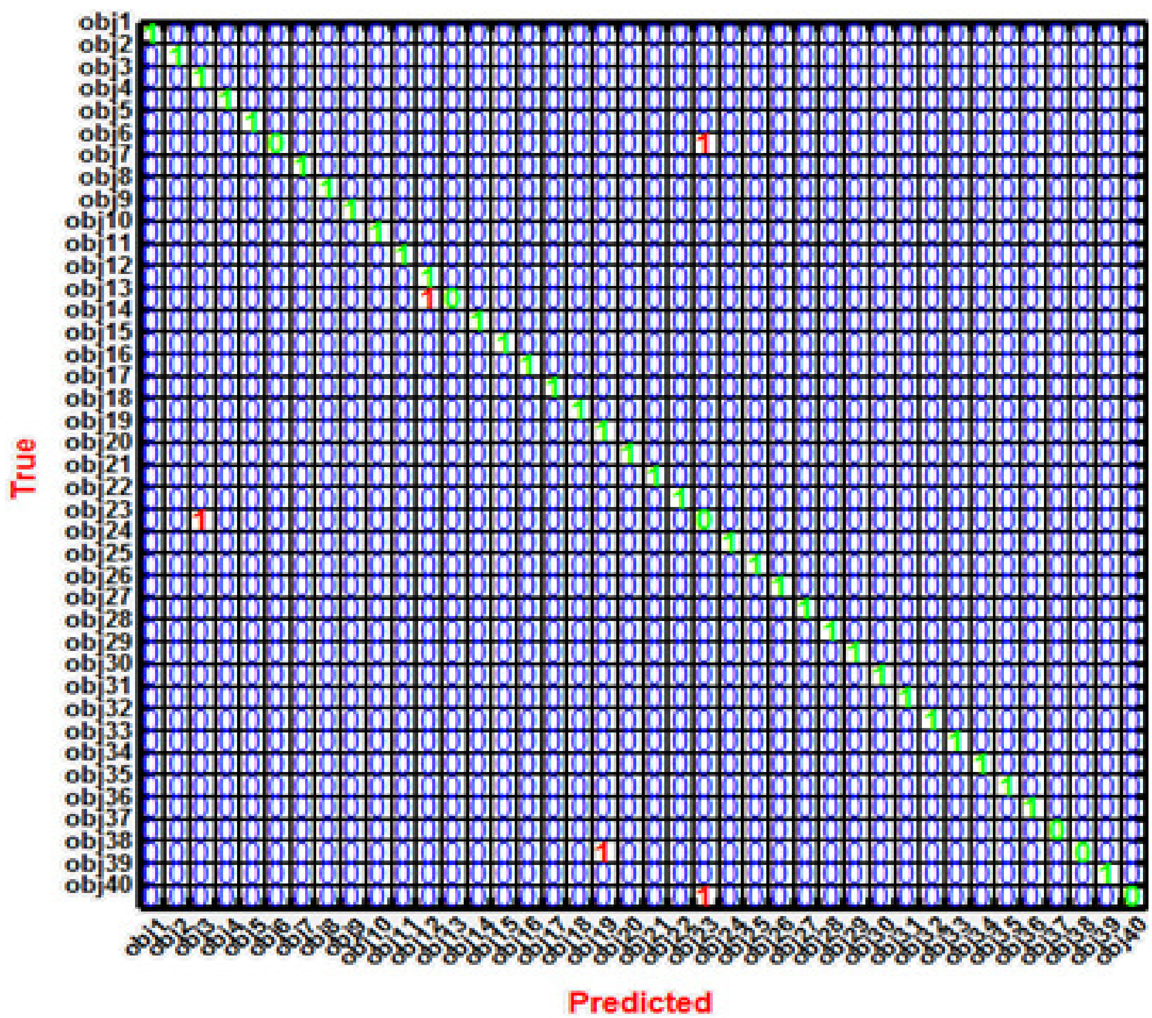
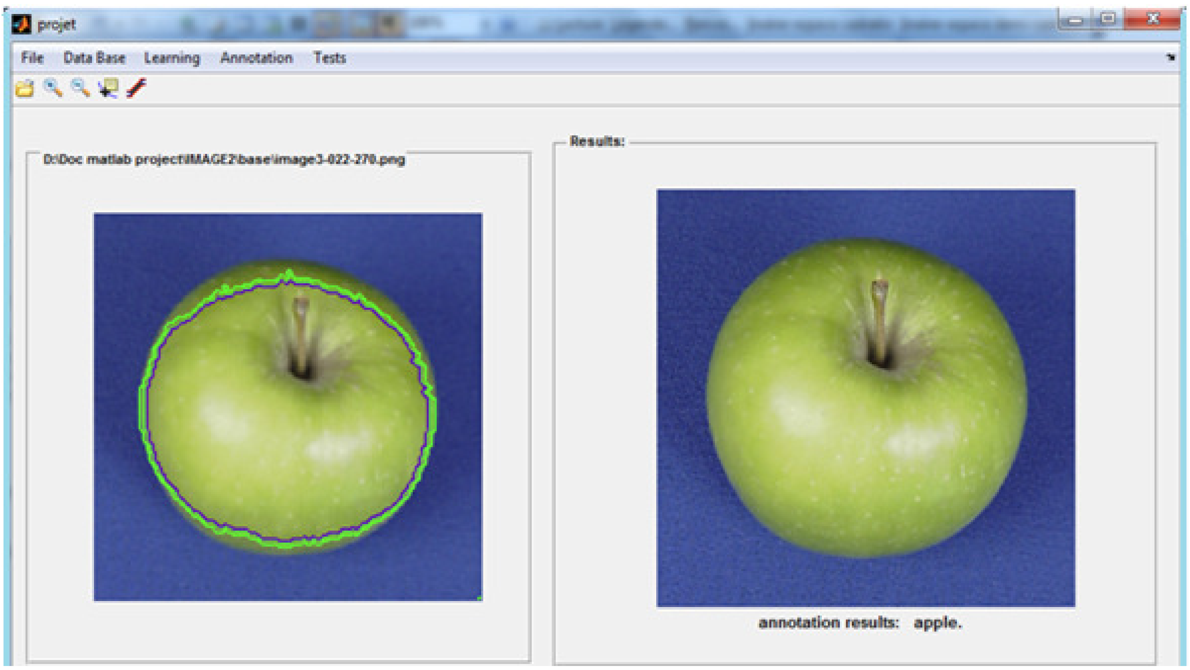
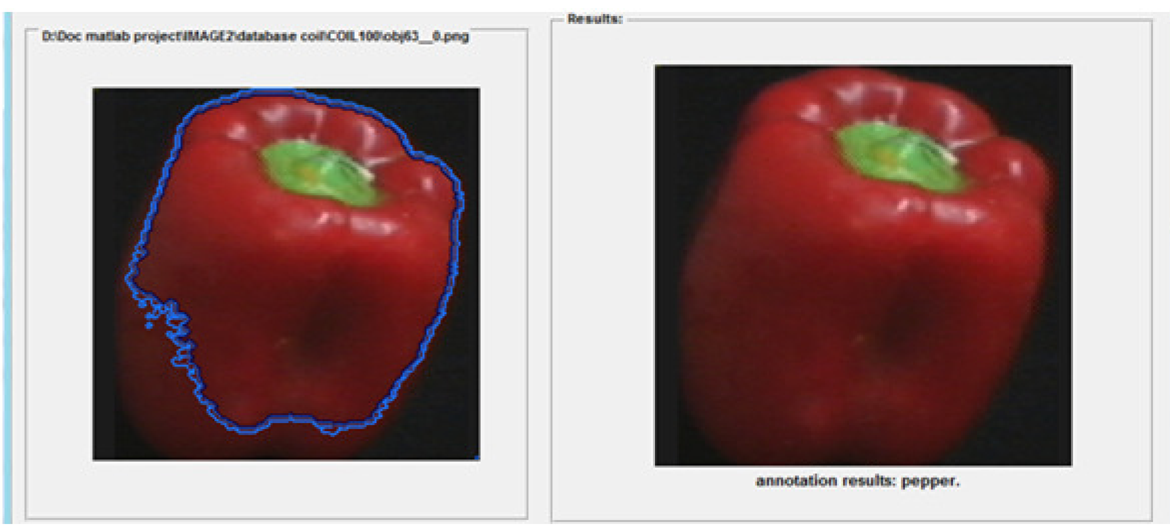


FIGURE 14: Confusion matrix for images of database COIL-100.

4.5 Analysis of Results

Analysis of the results presented in Table 2, Figures 12, 13 and 14, allows us to notice that the combination of neural networks with Bayesian networks in a parallel scheme, has significantly improved the quality of image annotation. Although, some errors are still persistent, namely in particular, the confusion between car and Cow in some times. This result is also illustrated by the examples of annotated images presented by figures 15 and 16 which shows that the exploitation of complementarities of generative and discriminative classifiers can contributes to the improvement of the image annotation. So, it would be interesting to investigate other ways to combine these two different classification approaches to possibly correct the observed annotation errors.





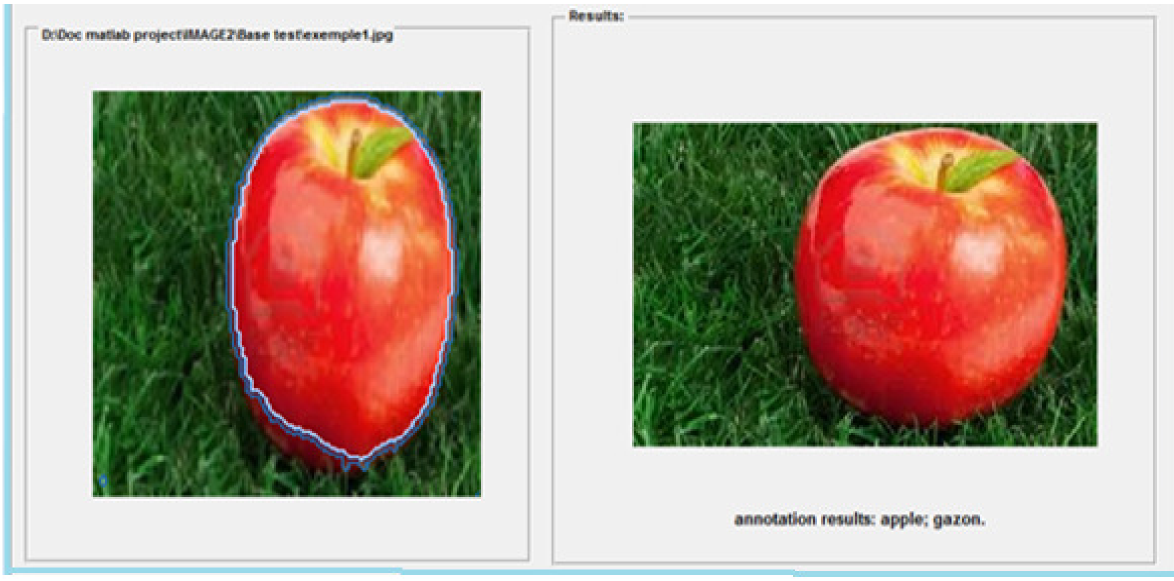


FIGURE 15: Examples of annotated images.

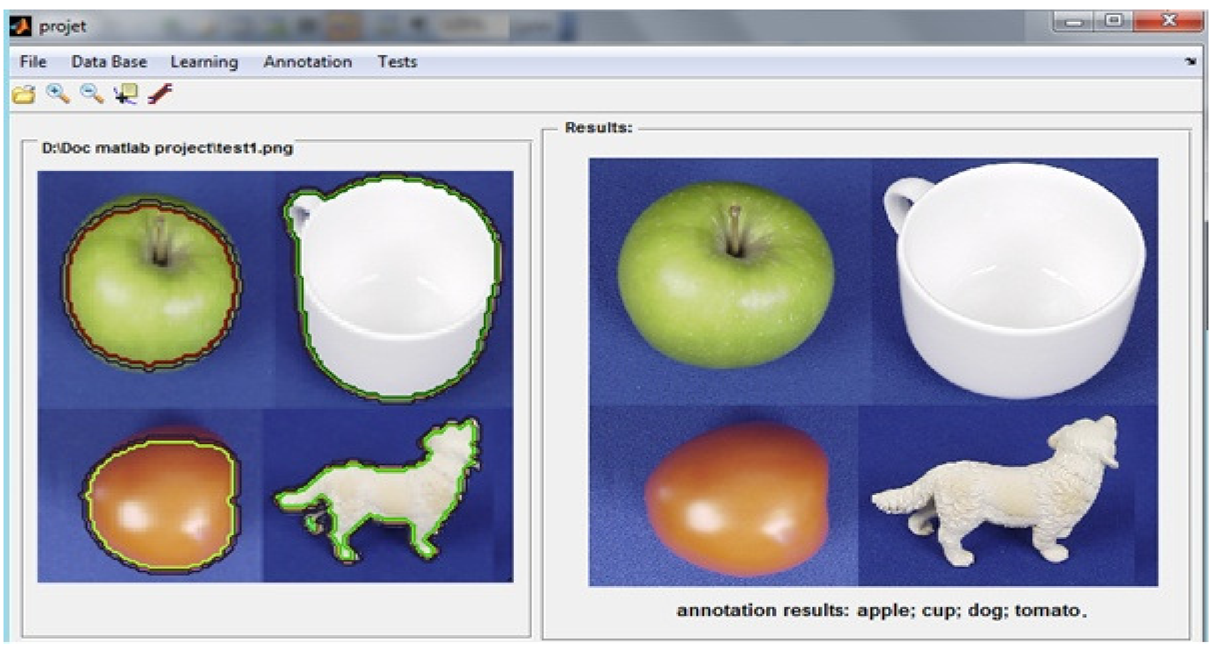


FIGURE 16: Examples of annotated images

5. CONCLUSION AND FUTURE WORK

In this work, we have proposed to build an efficient classifier for automatic image annotation via combining generative and discriminative classifiers which are respectively Bayesian networks and neural networks.

Starting with comparing these classifiers by realizing experiments on three image dataset, we have observed that neither classifier alone will be sufficient for semantic image annotation. So, we have combined the generative and discriminative classifier in parallel scheme in order to join and exploit their strengths. Experimental results show that this approach is promising for automatic image annotation because it gives better classification accuracy than either Bayesian networks or neural networks alone.

Our investigations suggest that the most fruitful approaches will involve some combination of generative and discriminative models. A principled approach to combining generative and discriminative approaches not only gives a more satisfying foundation for the development of new models, but it also brings practical benefits, address the extreme data-ambiguity and overfitting vulnerability issues in tasks such as automatic image annotation (AIA). In future work, we would like to develop others hybrid schemes that sought to integrate the intra-class information from generative models and the complementary inter-class information from discriminative models, and to research alternative optimization techniques utilizing ideas from the multi-criteria optimization of literature.

6. REFERENCES

[1] Li Z, Z. Shi P, Liu X and Shi Z, (2010) Automatic Image Annotation with Continuous PLSA, Proceedings of the 35th IEEE Intern Conf on Acoustics, Speech and Signal Processing, pp.: 806-809.

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

[3] Jianping Fan .Yuli Gao .Hangzai Luo (2007) Hierarchical Classification for Automatic Image Annotation. SIGIR Proceedings.

[4] Zhang R, Zhang Z, Li M, *et al*., (2005) A Probabilistic Semantic Model for Image Annotation and Multi-Model Image Retrieval, *Proc of the* 10*th IEEE Inter Conf on Computer Vision, pp.*:846- 851.

[5] Lavrenko V., Manmatha R., and Jeon J (2003) A model for learning the semantics of, pictures. In Proc of Advances in Neural Information Processing Systems, pp.:251–259.

[6] Oksana Yakhnenko (2009) Learning from Text and Images: Generative and Discriminative Models for Partially Labeled Data. Thesis, Iowa State University Ames.

[7] A. Y. Ng and A, Jordan M (2001) On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. In *Neural Information Processing Systems, pp.*: 841–848.

[8] lex Holub, Max Welling, Pietro Perona (2008) Hybrid Generative-Discriminative Visual Categorization**.** Inter Jour of Computer Vision, 77(3), pp.: 239-258.

[9] Ilkay Ulusoy1 , Bishop M (2006) Comparison of Generative and Discriminative Techniques for Object Detection and Classication: toward Category-Level Object Recognition, springer, pp.: 173-195.

[10] Lasserre J, Bishop C, Minka P. (2006) Principled hybrids of generative and discriminative models. Proc of the IEEE Computer Society Conf on Computer Vision and Pattern Recognition (CVPR), pp.:87–94.

[11] Timothy M, Shaogang G, and Xiang T. (2013) Finding Rare Classes: Active Learning with Generative and Discriminative Models. IEEE transactions on knowledge and data engineering 25 (2), pp.: 374 – 386.

[12] Cristani A, Castellani U, Murino V (2009) A hybrid generative/discriminative classification framework based on free energy terms. In ICCV.

[13] Anna B, Andrew Z, Xavier M (2008) Scene Classification Using a Hybrid Generative/Discriminative Approach. IEEE transactions on pattern analysis and machine intelligence, 30 (4), pp.: 712-727.

[14] Kelm M, Pal C, McCallum A (2006) Combining generative and discriminative methods for pixel classification with multi-conditional learning. ICPR, pp :828–832.

[15] Guillaume Bouchard and Bill Triggs (2004) The trade-off between generative and discriminative classifiers. proc of Computational Statistics Symposium,Physica-Verlag,Springer.

[16] Zhixin Li1, Zhenjun Tang1, Weizhong Zhao2, Zhiqing Li2 (2012) Combining Generative/Discriminative Learning for Automatic Image Annotation and Retrieval. Inter J of Intelligence Science, pp.:55-62.

[17] Shuang Hong Yang Jiang Bian College Hongyuan Zha (2010) Hybrid Generative/Discriminative Learning for Automatic Image Annotation. Proc Uncertainly artificial intelligence (UAI), pp.: 683-690.

[18] Frank Y , Shouxian C (2005) Automatic seeded region growing for color image segmentation . Image and Vision Computing 23, pp.:877–886.

[19] Ryszard S, Chora (2007) Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems, Inter J of Biology And Biomedical Engineering,1(1)1, pp.:6-16.

[20] Chee Way Chonga, Raveendranb P, Mukundan R, (2004) Translation and scale invariants of Legendre moments, Pattern Recognition 37, pp.:119 – 129.

[21] Haralick R, Shanmugan K, Dinstein I (1973) Textural features for image classification. IEEE Transactions on SMC, 3(6), pp. :610–621.

[22] Yue Cao, Xiabi Liu, Jie Bing, Li Song (2011) Using Neural Network to Combine Measures of Word Semantic Similarity for Image Annotation, IEEE International Conference on Information and Automation (ICIA), pp. :833 – 837.

[23] Simard P, Steinkraus D, Platt J (2003) Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, ICDAR, pp. :958-962.

[24] R. Lepage, & B. Solaiman. Les réseaux de neurones artificiels et leurs applications en imagerie et en vision par ordinateur, Ecole de technologie supérieure, 2003.

[25] Ann.Becker, Patrick Naim (1999) les réseaux bayésiens : modèles graphiques de connaissance. Eyrolles.

[26] Pearl J (1995) Bayesian Networks. UCLA Cognitive Systems Laboratory, Technical Report (R-216), MIT Press, pp.:149-153.

[27] Sabine Barrat (2009) Modèles graphiques probabilistes pour la reconnaissance de formes, Thèse, Spécialité informatique, Université Nancy 2.

[28] George H, Pat Langley (1995) Estimating continuous distributions in Bayesian classifiers. The Eleventh Conference on Uncertainty in Artificial Intelligence.

[29] Philippe LERAY (2006) Réseaux bayésiens : apprentissage et modélisation de systèmes complexes. Habilitation à diriger les recherches, Spécialité Informatique, Automatique et Traitement du Signal, Université de Rouen, France.

[30] Patrick Naïm, Pierre Henri Wuillemin, Philippe Leray, Olivier pourret, Anna becker, (2008) Réseaux bayésiens, Eyrolles, 3ème édition, Paris.

[31] Mitchell T (2010) Generative and discriminative classifier: Naïve bayes and logistic regression. Machine learning.

[32] ETH-80 database image. Online. Available: http://www.d2.mpi-inf.mpg.de/Datasets/ETH80.

[33] COIL-100 database image. Online. Available: http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php.