Refining Object Recognition Using Scene Specific Object Appearance Frequencies

Arthur Daniel COSTEA, Robert VARGA, Tiberiu MARITA, Sergiu NEDEVSCHI
Computer Science Department, Technical University of Cluj Napoca
{ arthur.costea, robert.varga, tiberiu.marita, sergiu.nedevschi }@cs.utcluj.ro

***Abstract*—This paper presents an automatic annotation method for multimedia data. Different object and scene recognition methods are analyzed from the literature. The best components of current methods are used to design and implement an original solution. A novel approach for refining results based on scene specific object appearance frequencies is exposed which improves annotation performance. Experimental results indicate the performance of the proposed solution which successfully competes with currently best methods.**

Image processing; automatic annotation; scene recognition; object recognition

I. INTRODUCTION

Automatic annotation assigns semantic information to resources like images and video data. The simplest case of annotation assigns a semantic label or tag to an image. This label describes the content of the image. The image content may be characterized mainly by the *image scene* - the general setting in which it was taken (context) - and by the *objects* present. Scene and object recognition have the task of obtaining the scene and the objects present for a given input image.

The automatic annotation of images and other data has received much attention in the last years and has proven to be a difficult task. Even though there are numerous methods that perform well for images that are similar to those belonging to the training set used, for general images the results are unsatisfactory.

This paper proposes a method for combining information obtained from scene recognition and object recognition. The information gained from these two procedures can be employed to obtain a much more reliable and robust annotation. In our work both objects and scenes chosen for recognition are simple, the goal being here is to obtain reliable annotations instead of more specific annotations with a low accuracy rate.

Our objectives are:

- develop a scene recognition method
- develop an object recognition method
- combine results in an efficient manner
- all methods must give annotations in a few seconds

II. RELATED WORK

The recognition methods can be mainly grouped into two types. The first type finds objects by comparing them to a previously generated codebook or dictionary. These methods use an underlying object model referred to as *orderless bag-of-keypoints* or *bag-of-words* due to the fact that it was first used in textual information retrieval[1]. This approach has high accuracy results and much research has been conducted on using it for different purposes and improving its performance [2][3][4][5][6][7].

The second type of methods are based on spatial or *part-based models*[8][9]. These make use of spatial information which is deliberately ignored in the *bag-of-words* model. Results for this model type on difficult databases have shown to be worse compared to those using bag-of-words models. We continue describing the first type of model.

Constructing a bag-of-words model entails the following main processing steps:

- feature vector extraction;
- clustering applied on feature vectors;
- comparison of instances to centers (histogram);
- classification.

Every step has a well defined purpose and the different implementation choices lead to different recognition methods. In what follows different methods are enumerated for each of the steps from the literature.

Many **feature types** have been employed for visual recognition problems. Some of them are: Color moments, DCT coefficients, HoG, SURF, GLOH, texton. Arguably the most frequently used feature type for recognition is the SIFT[10] (Shift Invariant Feature Transform) which is invariant to scaling, rotation, illumination and viewpoint.

In [11] the authors evaluate different descriptor types for recognition problems. Based on their results an efficient feature extraction technique makes use of both dense sampling and Harris-Laplace point sampling scheme. The best performances were achieved by SIFT descriptors from different color channels.

For **clustering** methods we mention three types. The first is k-means[12], which produces hard clustering assigning points to one specific center. The second is Expectation-

This work is sponsored by the "INSEMTIVES" EU founded FP7-ICT project, 2009-2012.

978-1-4577-1481-8/11/$26.00 ©2011 IEEE

Maximization[13], which performs soft clustering, assigning points to centers with certain probabilities. The third is Meanshift[14] which has been applied in image segmentation.

The usage of spatial pyramids in conjunction with the bag-of-words model has increased recognition accuracy[15]. This technique constructs **histograms** for image partitions resulting from dividing the image into more and more parts (thirds, quarters, sixteenths etc.).

Support Vector Machines are highly used **classifiers**. Improved accuracy can be obtained using different kernel types for transforming input data to higher dimensional or nonlinear spaces. Along with the popular Radial Basis Function kernel Chi-Square type kernels have been shown to achieve good results[16].

Next we present two recognition methods which more or less adhere to the steps presented above.

In Supervised Multiclass Labeling[17] Discrete Cosine Transform coefficients are used as feature vectors. These are extracted from densely spaced 8x8 patches. A hierarchical extension of the Expectation-Maximization clustering method is applied to obtain centroids which characterize each concept. A high-level feature vector is then obtained by finding the probabilities of an image to belong to each separate class. The vector is afterwards classified using an SVM classifier in case of scene recognition[18].

The bag-of-words approach can be used for semantic segmentation, i.e. segmentation of the image into objects and recognition of objects. Such a method was proposed by Yang et al. [19]. The image is first segmented into patches using mean-shift segmentation [14]. A bag-of-words model is used to classify individual patches. The similar patches are grouped and this way objects are obtained. Afterwards the bag-of-words model is used to classify these larger regions. The information obtained from individual patch classification and object classification is fused. Finally each patch of the image will have an object class label.

We mention a work where the simultaneous determination of both image scene and objects present has been treated [22].

Based on the study presented above we chose to use the bag-of-words model for our recognition problem.

III. Underlying Database

It is essential to have an extensive and representative image database when training classifiers for recognition problems. We have gathered images from several complete image databases and numerous additional sources. All images have 3 channels, use a 24 bit representation and are compressed using jpg standard. For all image classes - scene or object - at least 1000 instances were used. As a general preprocessing step we have resized larger images to have a maximum width or height equal to 320 pixels while retaining the image aspect ratio.

For scene recognition the majority of training instances were taken from the Scene UNderstanding database (SUN[23]). This database contains a collection of carefully selected images for important scenes. To achieve the minimum number of images per class we have complemented the available images from the above with image queries for the necessary scenes on the LabelMe database[24]; images from the Corel30k[25] dataset; and also from Google Images searches.

In the case of the object recognition training set almost all images have been obtained from Image-net[26]. For some concepts it was necessary to combine images from multiple nodes of the tree-structure. This entails gathering images from the parent node corresponding to a concept (e.g. train) and its subnodes (e.g. passenger, freight) and eliminating duplicates if needed. The whole data set was parsed several times to check for incorrect positive or negative samples. After initial tests we identified several potential confusion problems. For example sand was several times classified as snow and vice versa or grass as wheat. To resolve these issues, the negative examples where extended with samples containing objects which were often classified as positive.

IV. General Method Description

In the following we present the implementation for the four different phases mentioned in the introduction. This is a general description of the methods used for both object and scene recognition, the specifics for each are presented in the next section. In this section concept may refer to scene or object.

The general steps of the recognition algorithm are described by the following pseudo-code:

Training procedure

Input: Image database along with ground truth information
Output: Classifier models

```
1. for all images
2.        transform_colorspace(image);
3.        extract_descriptors(image);
4.        save_descriptors(image);
5. endfor
------------------------------------------------
6. samples = sample_descriptors();
7. find_cluster_centers(samples);
8. save_centers();
------------------------------------------------
9. for all images
10.       descriptors = load_descriptors(image);
11.       for all descriptors
12.               i = closest_center_index(descriptor);
13.               histogram[i]++;
14.       endfor
15.       normalize(histogram);
16.       save_histogram();
17. endfor
------------------------------------------------
18. for all concepts
19.       pos = collect_positive_histograms(concept);
20.       neg = collect_negative_histograms(concept);
21.       model[concept] = svm_train(pos, neg);
22. endfor
```

Lines 1-5 extract descriptors (feature vectors) from all images. Some descriptors require the image to be transformed to another colorspace as a preprocessing step. Cluster centers are found using k-means in lines 6-8 which will form the codebook (dictionary). Histogram calculation steps are shown in lines 9-17. The last part gathers histograms corresponding to positive and negative examples which will be used for SVM model training.

The steps required for annotating an image are:

```
Automatic annotation
Input: Classifier models, image with unknown concepts
Output: Semantic labels
1. resize(image)
2. transform_colorspace(image);
3. descriptors = extract_descriptors(image);
4. for all descriptors
5.         i = closest_center_index(descriptor);
6.         histogram[i]++;
7. endfor
8. normalize(histogram);
9. labels = svm_predict(models, histogram);
```

Here the same type of features are extracted from the test image as those used in the training. The histogram is formed using the dictionary from training phase and an SVM prediction is performed by supplying the histogram to the model.

A. Feature vector extraction

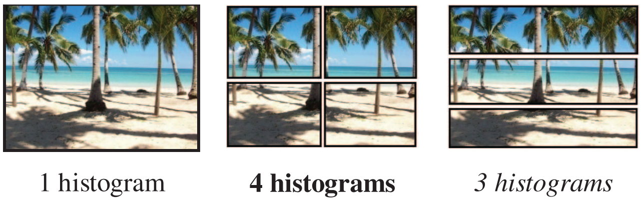
We employ a dense sampling strategy for feature vector extraction as this has shown better results than other point sampling strategies (Harris-Laplace corner detection). Based on [11] one of the best performing descriptors are C-SIFT. These are SIFT features extracted from the 3 channels of the normalized Opponent colorspace. For processing we employ the Vlfeat toolbox[27] to extract descriptors on a densely spaced grid of 6 pixels. Other feature types used are textons, which result from applying the Maximum Response filter bank [21] on the image. The final feature vector is a concatenation of the different types of descriptors.

B. Clustering method

For clustering we use k-means on as many descriptors as possible. This is done using the implemented function from the OpenCV computer vision library[28]. We have used descriptors from 500-700 images for dictionary construction. From a single image approximately 2000 descriptors are obtained. Collecting all responses from each image results in more than one million samples. This is sufficient because in the literature usually sets of around 200000 samples were used. Another possibility to obtain more representative centers for the entire database would be to use less descriptors from individual images (every 10th descriptor) and to use more images to draw descriptors from. We have worked with 200 cluster centers.

C. Histogram construction

We employ a spatial pyramid with 1x1, 2x2 and 3x1 type decomposition, see Fig. 1. For every descriptor the closest match from the dictionary is found. For speed optimality squared Euclidian distance is used – meaning no square root is calculated - and distance calculation halts once the current distance is greater than the current minimum. For every partition a histogram is constructed where each bin of the histogram corresponds to a cluster center. The bin value is the number of descriptors that are the closest to the corresponding cluster center. A final normalization step is performed.



1	**1/4**	**1/4**	**1/4**	**1/4**	*1/3*	*1/3*	*1/3*

Figure 1. Spatial pyramid decomposition and histogram weights

Histograms are weighed with 1/4 for quarter partitions, with 1/3 for 3x1 subdivisions and with 1 for the global image. The final histogram is the concatenation of all histograms leading to a final histogram length of 8 times the dictionary size.

D. SVM training and prediction

In the training phase we have used a workstation with a CUDA enabled GPU. The SVM model parameters (cost and gamma) are found by running a GPU-based implementation of the libsvm library[29] and performing a grid search. Instead of using the standard RBF kernel we have employed a modification using the Chi-Square distance between vectors. We have been experimenting with other kernel types and have found interesting behavior with some achieving similar results.

For two vectors x and y the Chi-square distance between them is defined as:

$$\chi^2(x,y)=\sum_i \frac{(x_i-y_i)^2}{x_i+y_i}.$$

Based on this distance the RBF kernel function can be modified as:

$$\phi(x,y)=exp\left(-\gamma\cdot\chi^2(x,y)\right).$$

In the case were the vectors are a concatenation of histograms obtained from partitions of the image (spatial pyramid type decomposition) each of these histograms needs to be weighed according to the size of the source image. This gives rise to the following definitions. Let the following indicate a partitioning in *p* vectors of *x*:

$$x=\left[x_{(1)},x_{(2)},\ldots,x_{(p)}\right].$$

Then weighing each partition of the vector with normalized weights w_p leads to the following kernel function:

$$\phi_p(x,y)=exp\left(-\gamma\cdot\sum_p w_p\chi^2\left(x_{(p)},y_{(p)}\right)\right)=exp\left(-\gamma\cdot\chi^2(w\circ x,w\circ y)\right)$$

$$w=\left[\underbrace{w_1,w_1,\ldots,w_1}_{partition\ 1},\underbrace{w_2,\ldots,w_2}_{partition\ 2},\ldots,\underbrace{w_p,\ldots,w_p}_{partition\ p}\right].$$

The operation ○ signifies an element by element multiplication (Hadamard product). This means that weighing the partitions is equivalent to weighing the individual elements accordingly. This property is useful at the implementation phase because the Chi-Square distance calculation remains the same.

V. Specifics of each Method

A. *Scene recognition*

For scene recognition we propose a two level classification. First an image is classified as indoor or outdoor and then it is classified further as a more specific indoor or outdoor class. In [15] a single multi-class classifier was used that was trained for 15 indoor and outdoor classes. In [30] first an image was classified as indoor or outdoor, but only the outdoor ones were classified further as a more specific class.

The used approach for scene recognition is based on the general method described in Section IV. Instead of training classifiers that predict the presence or absence of an object in an image, three multi-class classifiers are trained: indoor/outdoor, indoor and outdoor. The bag-of-words approach is used to obtain the feature vectors that are used for classification.

Three feature types were chosen as local descriptors for bag-of-words: CSIFT, dense-SIFT, texton. Sande et al. evaluated several color descriptors in [11] for object recognition, in which the CSIFT descriptors had the best performance. Lazebnik et al. used densely sampled gray-SIFT descriptors for scene recognition[15]. Li and Perona[31] showed, that dense features work better for scene classification than features sampled only at corner points. The gray-SIFT descriptor provides a certain degree of color invariance. The third descriptor used is the texton feature. A MR (Maximum Response) filter set consisting of 17 linear filters is applied over the image of interest. This way a feature vector of size 17 is obtained at each pixel. The MR filter set was used in [20][21], which provide local color and texture information.

Codebooks have to be obtained for each feature type. Using spatial pyramids, 8 histograms are obtained for each feature type, which results in a total of 24 histograms. Using codebooks of size 200 the length of the final feature vector, that describes the whole image, is 4800. These feature vectors are used for classification. Classification is done using a multiclass SVM classifier.

B. *Object recognition*

The set of objects is divided into indoor and outdoor objects. Objects may refer to general materials (e.g. snow, sand) or concrete objects (e.g. car). For each object a binary classifier is trained where the positive class means the object is present in the image, the negative class denotes the absence of the object.

SIFT descriptors are computed on the three channels of all images transformed to the normalized Opponent Color Space. A dictionary is constructed that consists of 200 cluster centers obtained using k-means on approximately 700000 descriptors previously found. Afterwards, the images are partitioned in quarters and also in one thirds in vertical direction. For every descriptor obtained the closest element from the dictionary is found and histograms are created for all image partitions as well as for the global image [15]. The final 1600 dimensional histograms are then used to construct training sets with positive and negative examples.

In the case of object recognition we rely on indoor/outdoor classification in order to choose to look for indoor or outdoor objects. This is why it is essential to have a very accurate indoor/outdoor classifier.

VI. Combining Results

By using both information resulting from scene recognition and object recognition a robust general annotation can be obtained for images. The main idea is to enhance one method using the other. Based on our tests, scene recognition provides a more reliable basis which can afterwards influence the probability of an object being present in an image.

Let s denote a column vector of n components containing the probabilities of each of the n scenes. Similarly, define c as a row vector of m components. The entries of c are the probabilities of each object being present. These are obtained by performing a classification on a test image. We next define the *object appearance frequency* matrix W. The entries of this matrix were obtained from the scene specific object appearance frequencies of the manually labeled training set. In order to penalize incorrect results entries on each column are normalized to have zero sums:

$$W \in M_{nxm}, \sum_{i=1}^{n}[W]_{ij} = 0, \forall j$$

This property is also necessary in order to prevent favoring a specific object.

In order to refine object probabilities we combine every scene probability with every object probability by multiplying them together along with the corresponding element from the W matrix. Afterwards we sum the products along the columns to obtain scores for each object (See Fig. 2).

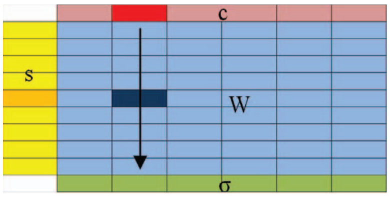


Figure 2. Obtaining score values

The formula for the score of each concept is given by:

$$\sigma_j = \sum_{i=1}^{n} s_i [W]_{ij} c_j \Rightarrow \sigma = \left(s^T \cdot W\right) \circ c$$

The last formula expresses the whole score row vector as an entry-wise (Hadamard) product denoted by ○. Object probabilities are modified using the following rule (Note: scores can be negative):

$$p \leftarrow \begin{cases} p + \Delta p & ,\sigma > \ t \\ p - \Delta p & ,\sigma < -t \\ \quad p & ,otherwise \end{cases},$$

where p signifies the probability of an object, Δp is the probability difference, σ is the score for the object and t is a threshold value. The best values for Δp and σ can be determined experimentally using a grid search.

VII. Experimental Results

For evaluation we used three measures from the information retrieval domain: precision, recall and accuracy for binary classification.

Table I contains the confusion matrix for indoor/outdoor classification. Columns indicate true class while rows show predicted class. The results were obtained using a 5-fold crossvalidation on the training set. Average accuracy is 97.7%.

Table II contains precision and recall values for indoor and outdoor scenes which resulted from training set and test set evaluation. Table III shows the performance of each binary classifier for individual objects.

TABLE I. Confusion matrix for indoor/outdoor classification

	outdoor	*indoor*
outdoor	5865	124
indoor	135	5376

TABLE II. Scene recognition 5-fold crossvalidation and test results

	crossv.		**test set**			**crossv.**	
Scene	***prec.***	***rec.***	***prec.***	***rec.***	***Scene***	***prec.***	***rec.***
coast, beach	76.8	75.4	79.5	74	*bathroom*	74.2	78.6
desert	81.3	82.9	74.3	81	*bedroom*	68.5	64.2
forest	87.2	89.2	89.5	86	*dining*	65.5	62.9
grassland	89.1	88.4	91.6	88	*hall*	95.4	95.4
highway	90.7	88.6	87.5	91	*kitchen*	75.4	72.3
lake, river	76.4	69.6	74.7	74	*living*	67.7	69.8
mountain	76.7	77.3	78.2	79	*mall*	80.5	83.3
open water	83.7	84.9	82.3	84	*office*	64.5	66.2
sky	88.2	91.2	87	87	*performance*	73.8	70.1
snow	77.5	78.5	75.7	72	*restaurant*	64.4	68.1
underwater	91.7	93.2	87.0	94	*store*	85.8	85.1
urban	88.1	89.6	86.5	84	average	74.1	74.1
average	83.95	84.07	82.5	82.8			

TABLE III. Object recognition 5-fold crossvalidation results

Object	**prec.**	**rec.**	**acc.**	**Object**	**prec.**	**rec.**	**acc.**
people	77.9	68.6	81.4	*window*	75.7	66.3	81.6
sky	84.3	82.3	86.2	*carpet*	88.9	82.3	91.0
water	90.2	87.2	91.5	*curtain*	86.5	79.0	88.8
tree	85.1	80.3	87.5	*floor*	77.2	68.9	82.5
grass	83.4	78.4	86.5	*door*	83.3	78.6	87.6
sand	82.4	75.3	86.7	*bed*	77.7	69.0	83.1
snow	84.2	80.3	87.1	*lamp*	80.7	66.4	85.4
rock	80.1	72.4	86.3	*tv*	80.6	70.5	84.2
ship	91.8	88.3	93.0	*computer*	76.4	70.3	82.8
bird	80.6	77.0	85.4	*wardrobe*	89.3	84.6	91.7
flower	84.7	83.0	88.8	*toilet*	81.9	74.6	86.0
wheat	90.9	88.9	93.7	*bathtub*	79.3	72.3	84.4
car	88.4	86.0	91.5	*shower*	83.7	78.5	87.6
train	88.1	86.5	89.8	*table*	77.3	67.4	84.5
airplane	90.2	87.4	92.8	*chair*	73.8	60.0	80.7
cloud	77.6	74.3	86.3	*fridge*	83.9	75.7	87.8
mountain	85.3	83.0	87.7	*armchair*	70.4	56.5	77.5
road	78.1	67.9	85.8	*shelf*	81.3	67.7	84.6
house	88.6	86.4	90.3	*desk*	73.4	65.4	80.4
building	87.4	85.0	89.5	average	80.07	71.26	84.85
street	87.4	84.5	89.1				
average	85.08	81.1	88.43				

TABLE IV. Scene detection comparison using 5-fold crossvalidation

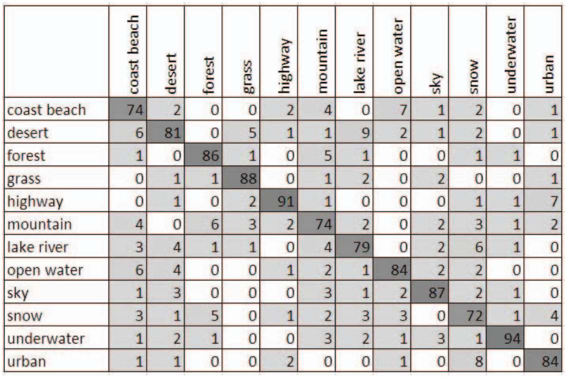
Scene	**Lazebnik [15]**	**Dunlop [30]**	**Our**
coast, beach	44	60	77
desert	48	76	81
forest	85	71	87
grassland	56	79	89
highway	79	67	91
lake, river	42	44	76
mountain	81	73	77
open water	67	70	84
sky	83	82	88
snow	69	75	77
urban	87	90	88
average	71	73	83

We have strived to use the same scene classes and same images for training in order to compare the results with those from [15] and [30]. The comparison of these methods is presented in Table IV based on scene classification precisions.

The Fig. 3-5 depict the confusion matrices for scene recognition in different evaluation scenarios. In the case of Fig. 4-5 1000 instances were used per class. For outdoor classes a test set was assembled containing 100 instances for every scene class, all of them different from those in the training set.

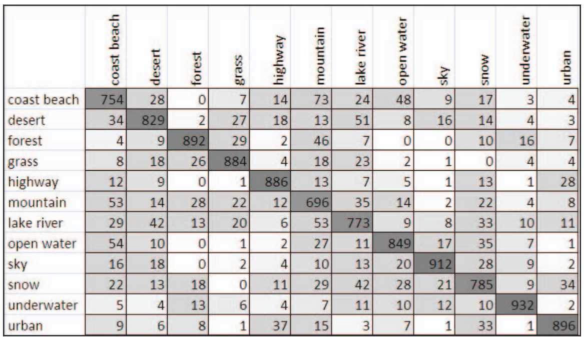
Table V shows the performance evaluation of the object recognition method on the test set for scenes. The objects that did not appear in the set are excluded from the table. We performed a grid search from 0.045 to 0.505 with a step of 0.045 for *Δp* and from 0 to 0.04 with a step of 0.002 for *t*. The table shows results for *Δp*=0.405, *t*=0.02. The entries in bold indicate an improvement in performance. It is obvious that for most cases performance improves, for objects where this is not the case the original probabilities can be used.

One of the major advantages of the presented methods is the speed. Average execution time for scene recognition is around 1.5 seconds. The most time consuming part is the computation of dense SIFT features. SIFT features are computed on 3 color channels and the grayscale image. These computations are obtained on 4 parallel threads. The system used for testing had an Intel Xeon 2.66 GHz CPU and 2 GB RAM. Another advantage is that the object recognition uses the same CSIFT features as the scene recognition. Test images are resized to have a maximum width or height equal to 320 pixels while retaining the image aspect ratio.



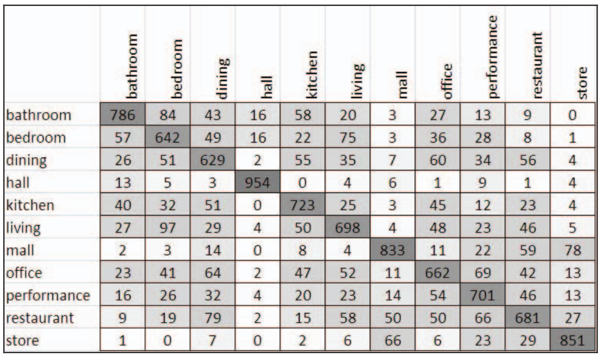
	coast beach	desert	forest	grass	highway	mountain	lake river	open water	sky	snow	underwater	urban
coast beach	74	2	0	0	2	4	0	7	1	2	0	1
desert	6	81	0	5	1	1	9	2	1	2	0	1
forest	1	0	86	1	0	5	1	0	0	1	1	0
grass	0	1	1	88	0	1	2	0	2	0	0	1
highway	0	1	0	2	91	1	0	0	0	1	1	7
mountain	4	0	6	3	2	74	2	0	2	3	1	2
lake river	3	4	1	1	0	4	79	0	2	6	1	0
open water	6	4	0	0	1	2	1	84	2	2	0	0
sky	1	3	0	0	0	3	1	2	87	2	1	0
snow	3	1	5	0	1	2	3	3	0	72	1	4
underwater	1	2	1	0	0	3	2	1	3	1	94	0
urban	1	1	0	0	2	0	0	1	0	8	0	84

Figure 3. Confusion matrix for outdoor scenes evaluated on the test set



	coast beach	desert	forest	grass	highway	mountain	lake river	open water	sky	snow	underwater	urban
coast beach	754	28	0	7	14	73	24	48	9	17	3	4
desert	34	829	2	27	18	13	51	8	16	14	4	3
forest	4	9	892	29	2	46	7	0	0	10	16	7
grass	8	18	26	884	4	18	23	2	1	0	4	4
highway	12	9	0	1	886	13	7	5	1	13	1	28
mountain	53	14	28	22	12	696	35	14	2	22	4	8
lake river	29	42	13	20	6	53	773	9	8	33	10	11
open water	54	10	0	1	2	27	11	849	17	35	7	1
sky	16	18	0	2	4	10	13	20	912	28	9	2
snow	22	13	18	0	11	29	42	28	21	785	9	34
underwater	5	4	13	6	4	7	11	10	12	10	932	2
urban	9	6	8	1	37	15	3	7	1	33	1	896

Figure 4. Confusion matrix for outdoor scenes evaluated on the training set using 5-fold crossvalidation



	bathroom	bedroom	dining	hall	kitchen	living	mall	office	performance	restaurant	store
bathroom	786	84	43	16	58	20	3	27	13	9	0
bedroom	57	642	49	16	22	75	3	36	28	8	1
dining	26	51	629	2	55	35	7	60	34	56	4
hall	13	5	3	954	0	4	6	1	9	1	4
kitchen	40	32	51	0	723	25	3	45	12	23	4
living	27	97	29	4	50	698	4	48	23	46	5
mall	2	3	14	0	8	4	833	11	22	59	78
office	23	41	64	2	47	52	11	662	69	42	13
performance	16	26	32	4	20	23	14	54	701	46	13
restaurant	9	19	79	2	15	58	50	50	66	681	27
store	1	0	7	0	2	6	66	6	23	29	851

Figure 5. Confusion matrix for indoor scenes evaluated on the training set using 5-fold crossvalidation

TABLE V. OBJECT RECOGNITION TEST RESULTS COMPARISON

	original		**refined**	
Object	***prec.***	***rec.***	***prec.***	***rec.***
people	72.5	40.0	52.0	**51.2**
sky	97.8	85.9	**97.9**	83.2
water	78.0	75.5	**87.3**	**83.7**
tree	92.4	60.3	92.3	59.1
grass	73.1	66.5	**77.2**	66.5
sand	71.2	79.6	**71.4**	**88.8**
snow	64.2	46.7	**74.4**	**62.0**
rock	67.4	46.9	56.1	**64.8**
flower	34.5	38.5	**35.6**	**61.5**
wheat	9.6	70.0	**10.1**	**80.0**
car	70.3	51.8	66.7	**84.7**
cloud	75.1	62.8	**76.9**	62.4
mountain	80.8	57.2	79.9	**65.7**
road	59.0	91.8	**78.4**	88.8
house	24.6	35.0	23.5	**47.5**
building	63.4	57.7	**70.0**	**75.7**
street	80.0	86.1	70.1	**94.4**
average	61.9	65.5	**71.8**	**65.9**

VIII. CONCLUSIONS

The work presented in this paper is the result of designing and implementing an automatic annotation system that works for general images. We have provided a detailed description of the methods used and the results obtained.

Contributions include: the assembling of the training image dataset, which contains about 36k images for objects and an additional 23k for scenes; analyzing the currently best performing methods and selecting the best parts from them; implementing both scene and object recognition; introducing a method for combining results in order to achieve more robust annotations. After fusing results from individual scene and object recognition average precision value increases by 10% while average recall value is preserved. Although we have presented a method to improve only object recognition it is possible to improve scene recognition also by making minor modifications (if object classifiers are considered more reliable).

Future research will be conducted on developing further the technique for fusing scene and object information. We have plans to extend the set of objects to be recognized by adding new classes (bus, bicycle, different animals, etc.). Another area of interest is the use of other kernel functions. Some kernel functions provide acceptable results with virtually any SVM parameters. A future paper might discuss these findings.

REFERENCES

[1] R. Baeza-Yates, and B. Ribeiro-Neto, "Modern Information Retrieval," ACM Press. 1999.

[2] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual Categorization with Bags of Keypoints," ECCV International Workshop on Statistical Learning in Computer Vision, 2004.

[3] J. Sivic, B.C. Russel, A.A. Efros, A. Zisserman, and W.T. Freeman, "Discovering objects and their localization in images," Proc. Int. Conf. on Computer Vision, vol. 1, pp 370–377, October 2005.

[4] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning Object Categories from Google's Image Search," Proc. Int. Conf. on Computer Vision, pp 1816–1823 , vol. 2, October 2005.

[5] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," J. Machine Learning Research, pp 993–1022, January 2003.

[6] B.C. Russel, A.A. Efros, J. Sivic, W.T. Freeman, and A. Zisserman, "Using Multiple Segmentations to Discover Objects and their Extent in Image Collections," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp 1605–1614, vol. 2, June 2006.

[7] J. Zhang, M. Marszałek, S. Lazebnik, C. Schmid "Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study," International Journal of Computer Vision, Vol.73, No. 2, pp. 213-238, 2007.

[8] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp 264–271, vol. 2, June 2003.

[9] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. R. Dunlop, "Object Detection with Discriminatively Trained Part Based Models".

[10] D.G. Lowe, "Distinctive Image Features from Scale Invariant Keypoints," Int'l J. Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[11] K.E.A. van de Sande, T. Gevers, and C.G.M. Snoek, "Evaluating Color Descriptors for Object and Scene Recognition", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 32, No. 9, September 2010

[12] J.B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967.

[13] A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," Journal of the Royal Statistical Society. Series B, 1977.

[14] D. Comaniciu, and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 24, No. 5, May 2002

[15] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," In Proc. CVPR, 2006.

[16] O. Chapelle, P. Haffner, and V. Vapnik, "SVMs for Histogram-Based Image Classification," IEEE Transactions on Neural Network, vol. 10, no.5, 1999.

[17] G. Carneiro, A.B. Chan, P.J. Moreno, and N.Vasconcelos, “Supervised Learning of Semantic Classes for Image Annotation and Retrieval,” IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 29, No. 3, March 2007.

[18] N. Rasiwasia, and N. Vasconcelos, “Scene Classification with Low-dimensional Semantic Spaces and Weak Supervision,” IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, June 2008.

[19] L. Yang, P. Meer, and D. J. Foran, “Multiple Class Segmentation Using A Unified Framework over Mean-Shift Patches,” Computer Vision and Pattern Recognition, 2007.

[20] J. Shotton, J. Winn, C. Rother, and A. Criminisi. “Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation”. ECCV, 1:1–13, 2006.

[21] M. Varma, and A. Zisserman, "Classifying images of materials: Achieving viewpoint and illumination independence", ECCV, 3:pp 255–271, 2002.

[22] C. Wang, D. Blei, and L. Fei-Fei, "Simultaneous Image Classification and Annotation", CVPR, 2009.

[23] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. "SUN Database: Large-scale Scene Recognition from Abbey to Zoo," IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[24] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," International Journal of Computer Vision, pp 157-173, Vol. 77, Nr. 1-3, May 2008.

[25] http://www.svcl.ucsd.edu/projects/imgnote/

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," IEEE Computer Vision and Pattern Recognition, 2009, http://image-net.org

[27] A. Vedaldi, B. Fulkerson, "VLFeat: An Open and Portable Library of Computer Vision Algorithms," Proceedings of the international conference on Multimedia, 2010

[28] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[29] C. Chang, and C. Lin, "LIBSVM: a library for support vector machines," 2001, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[30] H. Dunlop, “Scene classification of images and video via semantic segmentation”, POCV, 2010.

[31] F.-F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories," In IEEE CVPR Workshop on Generative-Model Based Vision, 2004.

[32] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, B. Zhang, "A formal study of shot boundary detection," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No.2, pp. 168-186, 2007.