Automatic Image Annotation using Multiple Grid Segmentation

Gerardo Arellano, L. Enrique Sucar, Eduardo F. Morales

Computer Science Department
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro 1, Tonantzintla, Puebla, México
{garellano, esucar, emorales}@inaoep.mx

Abstract. Automatic image annotation refers to the process of automatically labeling an image with a predefined set of keywords. Image annotation is an important step of content-based image retrieval (CBIR), which is relevant for many real-world applications. In this paper, a new algorithm based on multiple grid segmentation, entropy-based information and a Bayesian classifier, is proposed for an efficient, yet very effective, image annotation process. The proposed approach follows a two step process. In the first step, the algorithm generates grids of different sizes and different overlaps, and each grid is classified with a Naive Bayes classifier. In a second step, we used information based on the predicted class probability, its entropy, and the entropy of the neighbors of each grid element at the same and different resolutions, as input to a second binary classifier that qualifies the initial classification to select the *correct* segments. This significantly reduces false positives and improves the overall performance. We performed several experiments with images from the MSRC-9 database collection, which has manual ground truth segmentation and annotation information. The results show that the proposed approach has a very good performance compared to the initial labeling, and it also improves other scheme based on multiple segmentations.

1 Introduction

Most recent work on image labeling and object recognition is based on a sliding window approach [1], avoiding in this way the difficult problem of image segmentation. However, the rectangular regions used in these approaches do not provide, in general, a good spatial support of the object of interest; resulting in object features that are not considered (outside the rectangle) or incorrectly included (inside the rectangle). Recently, it has been shown [2] that a good spatial support can significantly improve object recognition. In particular, they demonstrate that by combining different image segmentation techniques, we can obtain better results, with respect to a sliding window approach or any of the segmentation techniques by themselves. They found that for all the algorithms considered, multiple segmentations drastically outperform the best single segmentation and also that the different segmentation algorithms are complementary, with each algorithm providing better spatial support for different object categories. So it seems that by combining several segmentation algorithms, labeling and object classification can be improved. In [2] the authors do not solve the problem of how to combine automatically the several segmentation methods.

More recently, Pantofaru et al. [3] proposed an alternative approach to combine multiple image segmentation for object recognition. Their hypotheses is that the quality of the segmentation is highly variable depending on the image, the algorithm and the parameters used. Their approach relies on two principles: (i) groups of pixels which are contained in the same segmentation region in multiple segmentations should be consistently classified, and (ii) the set of regions generated by multiple image segmentations provides robust features for classifying these pixel groups. They combine three segmentation algorithms, Normalized Cuts [4], Mean-Shift [5] and Felzenszwalb and Huttenlocher [6], with different parameters each. The selection of *correct* regions is based on *Intersection of Regions*, pixels which belong to the same region in every segmentation. Since different segmentations differ in quality, they assume that the reliability of a region's prediction corresponds to the number of objects it overlaps with respect to the class labels. They tested their method with the MSRC 21 [7] and Pascal VOC2007 data sets [8], showing an improved performance with respect to any single segmentation method.

A disadvantage of the previous work is that the segmentation algorithms utilized are computationally very demanding, and these are executed several times per image, resulting in a very time–consuming process, not suitable for real time applications such as image retrieval. Additionally, simple segmentation techniques, such as grids, have obtained similar results for region–based image labeling than those based on more complex methods [9].

We propose an alternative method based on multiple grid segmentation for image labeling. The idea is to take advantage of multiple segmentations to improve object recognition, but at the same time to develop a very efficient image annotation technique applicable in real–time. The method consists of two main stages. In the first stage, an image is segmented in multiple grids at different resolutions, and each rectangle is labeled based on color, position and texture features using a Bayesian classifier. Based on the results of this first stage, in the second stage another classifier qualifies each segment, to determine if the initial classification is correct or not. This second classifier uses as attributes a set of statistical measures from the first classifiers for the region of interest and its neighbors, such as the predicted class probability, its entropy, and the entropy of the neighbors of each grid element at the same and different resolutions. The incorrect segments according to the second classifier are discarded, and the final segmentation and labeling consists of the union of only the *correct* segments.

We performed several experiments with images from the MSRC-9 database collection, which has manual ground truth segmentation and annotation information. The results show that the proposed approach is simple and efficient, and at the same time it has a very good performance, in particular reducing the false positives compared to the initial annotation. We also compared our approach for selecting segments based on a second classifier, against the method of [3] which uses region intersection, with favorable results.

The rest of the paper is organized as follows. Section 2 presents an overall view of the proposed approach. Section 3 describes the segmentation process and the feature extraction mechanism. Section 3.3 describes the first classifier used to label the different segments in the images, while Section 4 explains the second classifier used to remove false positives and improve the overall performance of the system. In Section 5, the experiments and main results are given. Section 6 summarizes the main conclusions and provides future research directions.

2 Image Annotation Algorithm

The method for image annotation based on multiple grid segmentation consists of two main phases, each with several steps (see Figures 1 and 2):

Phase 1 – Annotation:

1. Segment the image using multiple grids at different resolutions.
2. Extract global features for each segment: color, texture and position.
3. Classify each segment based on the above attributes, using a Bayesian classifier previously trained based on a set of labeled images.

Phase 2 – Qualification:

1. For each segment obtain a set of statistical measures based on the results of the first classifier: class probability, entropy of the region, and entropy of its neighboring regions at the same and different resolutions.
2. Based on the statistical measures, use another binary Bayesian classifier to estimate if the original classification of each segment is correct/incorrect.
3. Discard *incorrect* segments (based on certain probability threshold).
4. Integrate the correct segments for the final image segmentation and labeling.

In the following sections each phase is described in more detail.

3 Phase 1: initial annotation

Before applying our classifiers for image annotation, we perform two operations on the images: (i) segmentation and (ii) features extraction.

3.1 Image Segmentation

Carboneto [9] found that a simple and fast grid segmentation algorithm can have better performance than a computationaly expensive algorithm like Normalized Cuts [4]. The effectiveness of grid segmentations depends on the size of the grid and on the nature of the images. On the other hand, combining several segmentation algorithms tend to produce better performance, as suggested in [2]

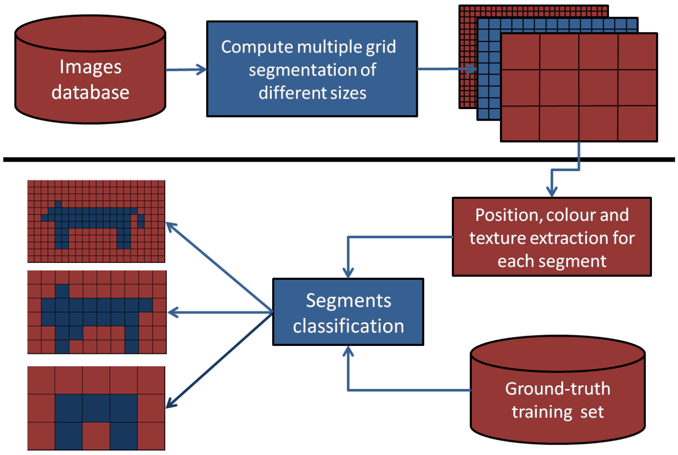
Images database
Compute multiple grid segmentation of different sizes
Position, colour and texture extraction for each segment
Segments classification
Ground-truth training set

Fig. 1. Phase 1: annotation. For each image in the database we compute multiple grid segmentations. For each segment we extract position, color and texture features; and use these features to classify each segment.

and shown in [3]. In this paper we propose to use several grid segmentations, label each segment and combine the results, integrating two relevant ideas: (i) grid segmentation has a very low computational cost with reasonable performance so it can be efectively used in image retrieval tasks, and (ii) using grids of different sizes can produce relevant information for different regions and images of different nature. So rather than guessing which is the right grid size for each image, we used a novel approach to combine the information of the different labels assigned to each grid to improve the final segmentation and labelling results. In this paper we used three grid segmentations of different sizes, but the approach can be easily extended to include additional grids.

3.2 Features

There is a large number of image features that have been proposed in the literature. In this paper we extract features based on position, color and texture for each image segment, but other features could be incorporated.

As position features, we extract for each image segment the average and standard deviations of the coordinates, x and y, of the pixels in the rectangular region.

Color features are the most common features used in image retrieval. In this paper we used the average and standard deviation of the three RGB channels for each image segment (6 features). In the CIE-LAB Color space the numeric differences between colors agrees more consistently with human visual perceptions. Thus, we also include the average, standard deviation and skewness of the three channels of the CIE-LAB space for each image segment.

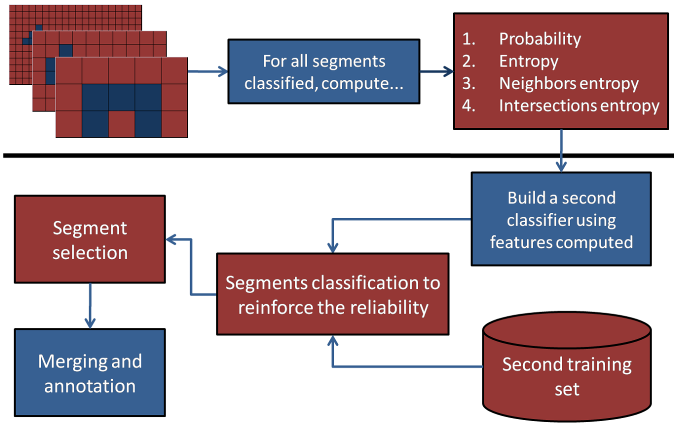
For all segments classified, compute...
1. Probability
2. Entropy
3. Neighbors entropy
4. Intersections entropy
Build a second classifier using features computed
Segments classification to reinforce the reliability
Second training set
Segment selection
Merging and annotation

Fig. 2. Phase 2: qualification. For all the classified segments, we obtain the probability of the label, its entropy, the neighborhoods entropy and the intersection entropy; as input to another classifier to reinforce the confidence of each label and then we finally select the *correct* segments and merge segments with the same class.

The perception of textures also plays an important role in content-based image retrieval. Texture is defined as the statistical distribution of spatial dependences for the gray level properties [10]. One of the most powerful tools for texture analysis are the Gabor filters [11], a linear filter used in image processing for edge detection. Frequency and orientation representations of Gabor filter are similar to those of human visual system, and it has been found to be particularly appropriate for texture representation and discrimination. Gabor filters could be viewed as the product of a low pass (Gaussian) filter at different orientations and scales. In this paper we applied Gabor filters with four orientations $\theta = [0, 45, 90, 135]$, and two different scales, obtaining 8 filters in total.

3.3 Base Classifier

The previously described features are used as input to a Naive Bayes classifier. This classifier assumes that the attributes are independent between each other given the class, so using Bayes theorem the class probability given information of the attributes $(P(C_i|A_1, ..., A_n))$ is given by:

$$P(C_i|A_1, ..., A_n) = \frac{P(C_i)P(A_1|Ci), ..., (A_n|Ci)}{P(A_1, ..., A_n)} \tag{1}$$

where C_i is the *i-th* value of the class variable on several feature variables $A_1, ..., A_n$. For each segment we obtain the probabilities of all the classes, and select the class value with maximum *a posteriori* probability; in this way we label all the segments created by the multi-grid segmentation process.

4 Phase 2: annotation qualification

In the second phase we use a second classifier to qualify the classes given by the first classifier and improve the performance of the system. This is a binary classifier that decides whether the predicted label of the first classifier is correct or not, given additional contextual information. We compute the likelihood of the predicted class using another Naive Bayes Classifier.

As attributes for this second classifier we use:

Class Probability: The probability of the label given by the first classifier.

Entropy: The entropy of each segment is evaluated considering the probabilities of the predicted labels by the first classifier, defined as:

$$H(s) = -\sum_{i=1}^{n} P(C_i) \log_2 P(C_i) \tag{2}$$

where $P(C_i)$ is the likelihood of prediction for class i and n is the total number of the classes.

Neighborhood Entropy: We also extract the entropy of the segment's neighbors and add this information if they have the same class:

$$H(v) = \frac{1}{|v_c|} \sum_{x \in V_c} H(x)\delta(Class(x), Class(v)) \tag{3}$$

where V_c are all neighbors segments, $H(x)$ is the entropy from neighbors segments with the same $Class(v)$, normalized by the number of neighbors with the same class $|v_c|$ and,

$$\delta(x, v) = \begin{cases} 1 & \text{if } x = v \\ 0 & \text{otherwise} \end{cases}$$

This is illustrated in Figure 3, where the class label is illustrated with different colors. In this case, three neighbors have the same class as the central grid, while one neighbor has a different class. The value of the attribute will have the normalized sum of the entropies of three neighbors with the same class.

Intersection Entropy: We also consider information of the cells from other grids that have an intersection with the current segment. In this case we used Equation 3 and applied it to the segments of the different grids. In the case of the three grids considered in the experiments, the cells in the largest grid size have 20 neighbors each, the middle size segments have 5 neighbors each and the smallest segments have only two neighbors. This is illustrated in Figure 4. Different grid segmentations and intersection schemes could be used as well.

Combined Attributes: We also incorporated new attributes defined by a combination of some of the previous attributes, namely the entropy of the segment plus the neighbor's entropy and the entropy of the segment plus the

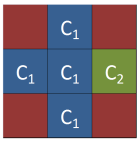

Fig. 3. Neighbors in the same grid. We consider the top, down, right and left neighbors and obtain their entropy, considering only the neighbors that have the same class as the current segment (best seen in color).

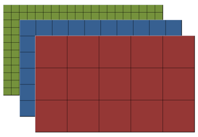

Fig. 4. Neighbors in the other grids. The image is segmented in grids of different sizes, each segment will have a different number of intersected neighbors. For the three grids we used in the experiments, the largest segments have 20 neighbors, the middle size segments have 5 neighbors, and the smallest segments have only 2 neighbors (best seen in color).

intersection's entropy. The incorporation of additional features that represent a combination of features can sometimes produce better results. Other features and other combinations could be used as well.

The second classifier is shown in Figure 5. This classifier is used to qualify each segment and filter the incorrect ones (those with a low probability of being correct are discarded).

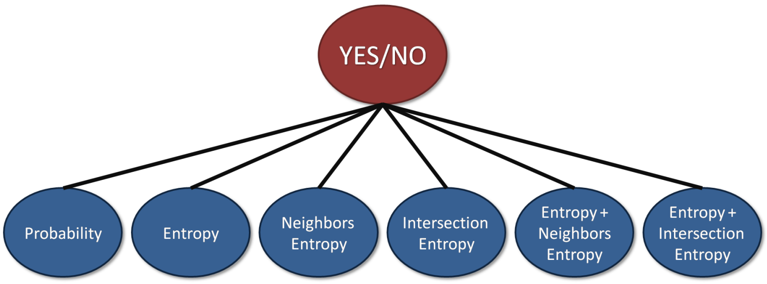

Fig. 5. Second classifier. Graphical model for the qualification classifier, showing the features used to decide if segments were correctly or incorrectly classified.

5 Experiments and Results

We performed several experiments using the MSRC-9 database with 9 classes: grass, cow, sky, tree, face, airplane, car, bicycle and building. All images were resized to 320x198 or to 198x320 depending on the original image size, and segmented using three different grids with cell sizes of 64, 32 and 16 pixels.

The database has 232 images, 80% were randomly selected and used for training and 20% for testing for both classifiers. Position, color and texture features were extracted for the first classifier and the features mentioned in Section 4 for the second classifier.

We first analyzed some of the features used in the second classifier. We found that entropy and the intersection entropy measures work well as a discriminant between segments that are correctly or incorrectly classified. This is shown in Figure 6, where the left graph shows the division between correctly and incorrectly classified segments using entropy, while the right graph shows the separation considering the intersection entropy.

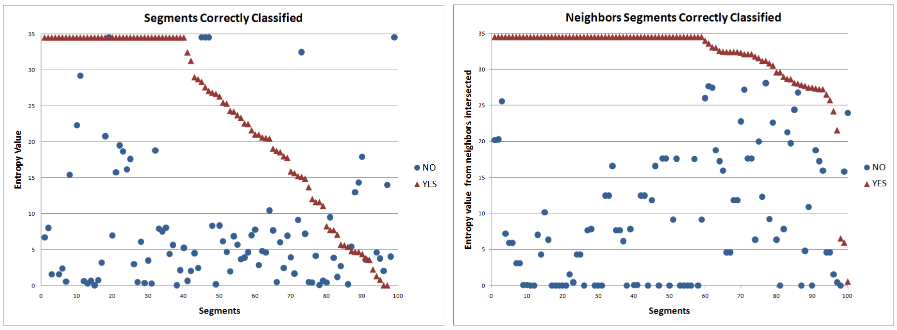

Fig. 6. The figures show how a sample of segments (100) for different classes are classified, as correct (red triangles) or incorrect (blue circles), based on the segment local entropy (left) and the intersection entropy (right). Although there is not a perfect separation, we can observe certain tendency in both graphs, which confirms that these features could be used as indicators to distinguish correct vs. incorrect region classification.

Then we compared the performance of the base classifier against the performance after the segments are selected based on the qualification classifier. The results in terms of true positives (TP) and false positives (FP), are summarized in Figure 7. Although there is a slight reduction in true positives, there is a great reduction in false positives; showing that our method can effectively eliminate incorrect segments. The reduction in TP is not a problem, as there is a redundancy in segments by using multiple grids.

Finally, we compared our method for segment qualification based on a second classifier against the method proposed in [3] based on intersection of regions.

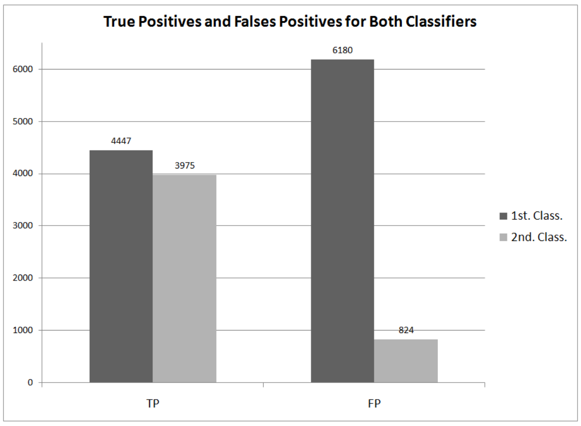

Fig. 7. Comparison of the results with only the first classifier and after the second classifier is applied. The graphs show the TP and FP generated by both classifiers.

For this we used the same data, and incorporated the intersection criteria into our multiple grid segmentations, instead of the second classifier. The results in terms of different performance criteria are summarized in Figures 8 and 9. In Figure 8 we compare the number of true positives (TP) and false positives (FP) for our approach vs. region intersection; and in Figure 9 we compared them in terms of precision, recall and accuracy. To determine the selection of a segment is considered that the likelihood rating was above 90% for TP and FP. We observe that our method for region qualification outperforms the method based on intersections in all the performance measures, with significant differences in TP and precision. Examples of region labeling and qualification for 4 images with two different grid sizes are shown in Figure 10.

Our current implementation in MatLab requires about 30 seconds to process an image for the complete process (both phases) using a PC with a dual core at 2.4 GHz and 4GB of RAM. We expect that an optimized “C/C++” implementation can reduce this time at least an order of magnitude.

These experiments show that our novel approach using a second classifier with contextual information based on statistical measures, produces significant improvements over the initial labeling, and is more effective that the approach based on intersections.

6 Conclusions and Future Work

In this paper we proposed an automatic image annotation algorithm using a multiple grid segmentation approach. Images are segmented in a very efficient way using a simple grid segmentation with different grid sizes. Individual features based on position, color and texture are extracted and used for an initial

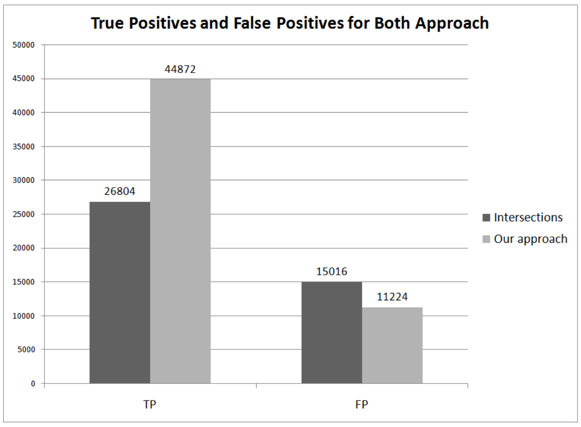

Fig. 8. Comparison of segment selection based on our approach vs. region intersection in terms of true positives (TP) and false positives (FP).

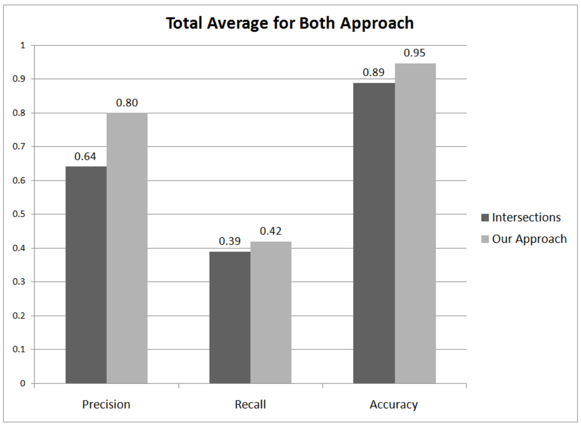

Fig. 9. Comparison of segment selection based on our approach vs. region intersection in terms of precision, recall and accuracy.

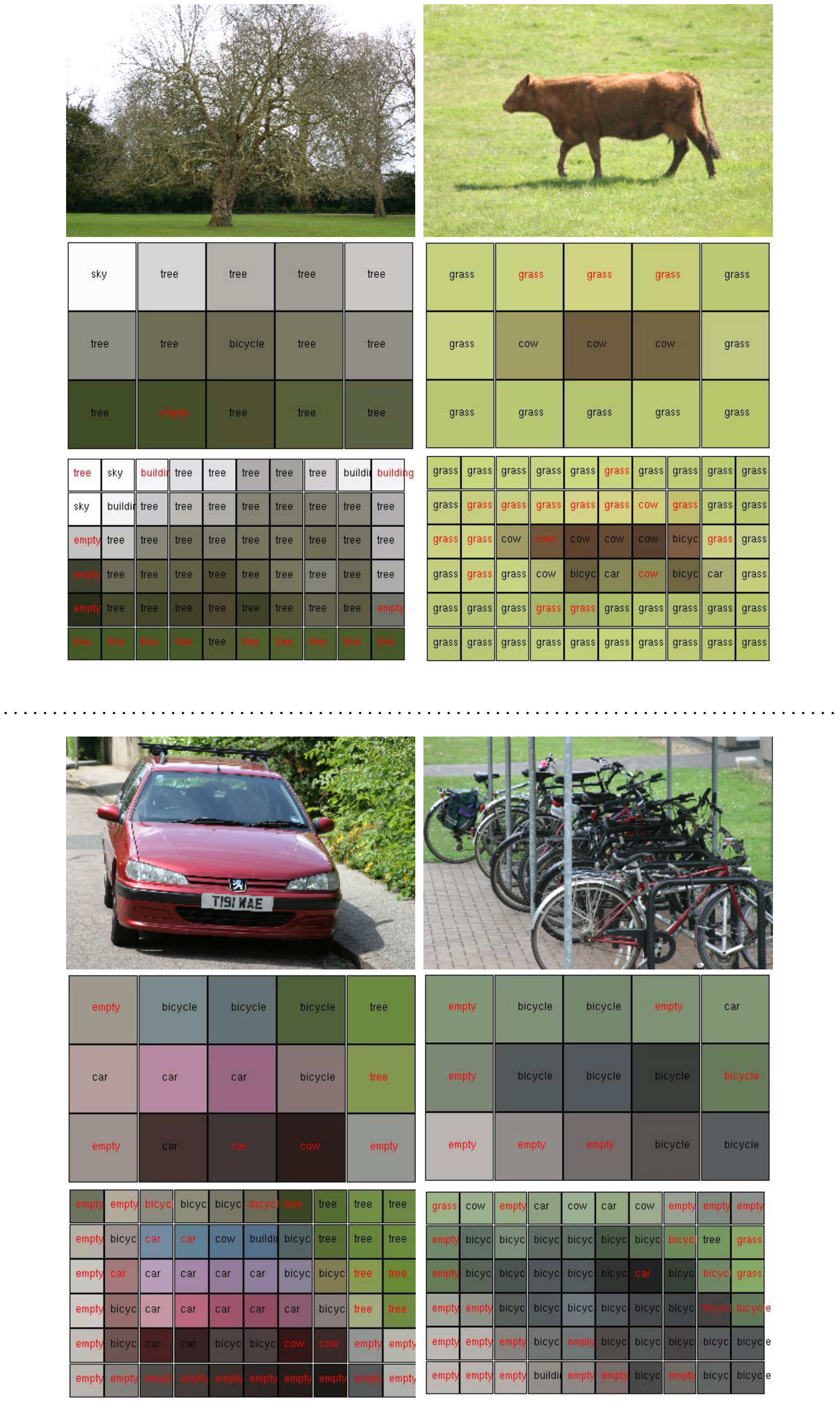

Fig. 10. Sample images. Up: original image. Middle: coarse grid. Down: intermediate grid. Correct labels are in black and incorrect in red (best seen in color).

labeling of the different grid segments. This paper introduces a novel approach to combine information from different segments using a second classifier and features based on entropy measures of the neighbor segments. It is shown how the second classifier significantly improves the initial labeling process decreasing the false negative rate; and has a better performance that a selection based on region intersection.

As future work we plan to combine the correct regions in the different grids to produce a final single segmentation and labeling; and to test our approach in other image databases.

Acknowledgments

We would like to thank to the members of the INAOE-TIA-research group by their comments and suggestions. This work was partially supported by CONACyT under grant 271708.

References

1. Viola, P., Jones, M.: Robust real-time face detection. Int. J. of Comp. Vision (2001)
2. Malisiewicz, T., Efros, A.A.: Improving spatial support for objects via multiple segmentations. In: BMVC (2007)
3. C. Pantofaru, C.: Object recognition by integrating multiple image segmentations. In: Computer Vision – ECCV 2008. Lecture Notes in Computer Science (2008) 481–494
4. J. Shi, J.M.: Normalized cuts and image segmentation. In Proc. CVPR (1997) pages 731–743
5. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis pami (2002). IEEE Trans. Patt. Anal. Mach. Intell. **24** (2002) 603–619
6. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. Int. Journal of Computer Vision **59** (2004) 167–181
7. Shotton, J., Winn, J., Rother, C., Criminisi, A.: The msrc 21-class object recognition database. (2006)
8. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal voc 2007 (2007)
9. Carbonetto, P.: Unsupervised statistical models for general object recognition. Master's thesis, The University of British Columbia (2003)
10. Aksoy, S., Haralick, R.: Textural features for image database retrieval. CBAIVL 1998, IEEE Computer Society (1998) 45
11. Chen, L., Lu, G., Zhang, D.: Content-based image retrieval using gabor texture features. In: PCM 2000. Sydney, Australia (2000) 1139–1142