Fast image segmentation and classification

Ivana Lj. Sopovska, and Zoran A. Ivanovski, *Member, IEEE*

***Abstract* — The paper presents a method for image segmentation and classification of the segments. Every image segment is associated to one of the 10 predefined classes. Feature vector in the HSV color space is extracted for every 6x6 pixels block. Image segmentation is performed by K-means clustering of the feature vectors. Classification of the segments is achieved by 9 SVM classifiers using one-vs-all method. Finally, „smoothing" of the classification results is performed by median filtering.**

***Keywords* — classification, clustering, color segmentation, HSV, K-means, SVM.**

I. Introduction

IMAGE content classification can be described as labeling of objects in the image in one or several predefined categories. This is usually not a big problem for humans, but a very difficult task for computers. Most of the difficulties come from the variable and often uncontrolled conditions in the image acqusition process, complex objects, objects occluding other objects and also from the difference between the image representation by humans and the computer.

The algorithms for image classification are usually based on image segmentation. For the image segmentation step, mean shift and k-means clustering methods are commonly used. Sometimes segmentation is carried out on multiple hierarchy levels, as in [1]. Two processes are involved in this approach: hierarchical iteration in which the spatial constrained region growing is carried out, and a merging process.

Another approach, [2], is based on the visual perception of the variation in Hue, Saturation and Intensity values of an image pixel. Pixel features are extracted by either choosing the Hue or the Intensity as the dominant property based on the Saturation value of a pixel. For each pixel, therefore, either its Hue or the Intensity as the dominant feature is chosen based on its Saturation. Then the image is segmented using the K-means clustering algorithm.

Some researchers have applied Multiple Instance Learning (MIL) for image classification, [3, 4, 5]. Images are viewed as bags containing a number of instances corresponding to regions obtained from image segmentation. A bag is essentially summarized by one of its instances. Bags and instances share the same set of labels (or categories or classes) and a bag receives a particular label if at least one of the instances in the bag possesses the label.

In [6], an extension of MIL, called DD-SVM is proposed. A bag label is determined by some number of instances satisfying various properties. The feature vector consists of six features. Three of them are the average color components in the LUV color space, the other three represent square root of energy in the high-frequency bands of the wavelet transforms. They used K-means clustering as a method for segmentation. From the perspective of learning, in this approach, labels are associated with images instead of individual regions.

For the classification part, SVMs are widely used. The article in [7] explains the convenience one-vs-all approach for multiclass classification.

The proposed algorithm can be divided in four parts: representation of the image as a set of feature vectors, (II.A), image segmentation by clustering the feature vectors (II.B), classification of the cluster centers, (II.C), and postprocessing (II.D) for „smoothing" the classification result.

Potential application of this algorithm are digital libraries, content based image retrieval, content based image and video analysis, geographic information systems, biomedicine, thematic organizing of images and many others. It can also be used as a preprocessing step for more complex processing.

II. Overview of the approach

A. Feature extraction

The proposed algorithm is designed to work with images of size 640x480 pixels or similar. Therefore, in the firs step, if the image size is significantly different from previous, the image is resized to 640x480 pixels.

The features are extracted in the HSV color space. We chose the HSV color model because it is close to the way people describe color. The image is first divided in small blocks and then a feature vector for every block is extracted. A rigid partition of the image preserves certain spatial information, but it often puts different objects into a single block. By this the visual information about objects could be destroyed. The block size is a compromise between including enough information for succesful block classification and avoiding multiple objects in the block. Based on intensive experimental testing, a block size of 6x6 pixels was chosen.

The feature vector is composed of 15 features: the first three are the average values of the three color components

Ivana Lj. Sopovska, Faculty of Electrical Engineering and Information Technologies, Skopje, Karpos bb, 1000 Skopje, Macedonia (phone: 389-2-3099103, e-mail: ivana.sopovska@gmail.com)

Zoran A. Ivanovski, Faculty of Electrical Engineering and Information Technologies, Skopje, Karpos bb, 1000 Skopje, Macedonia (phone: 389-2-3099103, e-mail: zoran.ivanovski@feit.ukim.edu.mk).

in the block, the next three are the values of the standard deviation of the colors in the block, and the last nine are the nine bins of the histogram of oriented gradients of the V component of the block.

The Hue component is usually represented by an angle in a circle of possible Hues, and at angle of 0^o lays the red color. The difference between two Hue values increases as the angle between them increases. The colors near 0^o, but also the colors near 360^o appear very similar to the red color. However, the values of these colors are very different because they are near the both ends of the range. To represent the distance similar to human understanding of colors, a method that „corrects“ the distance measure between two values was used. If an angle α between two Hue values is bigger than 180^o, then this angle is represented by the explementary angle of α (360^o- α).

The Histogram of Oriented Gradients (used for the last 9 features in the feature vector) is computed similarly to the method in [8]. It consists of 9 bins representing intervals of 20^o of the possible orientations of the gradient from -90^o to 90^o. The horisontal and vertical gradients are computed for every pixel in the V color component using masks [-1 0 1] and $[-1\ 0\ 1]^T$, respectively. To achieve lower computational complexity, the L1 norm is used for computing of the magnitude of the vector. The angle of the gradient vector is computed as arctan function of the ratio between the vertical and horizontal component, using a look-up table for increasing the speed of the algorithm. After computing the angle, the value of the bin it belongs to is increased by the magnitude of the vector, instead by 1, as usual. This way the pixels with higher gradient will have bigger influence in the final histogram.

We can use the cross validation accuracy to give an estimation of how adequate the described feature extraction method is. The achieved accuracies of the classes vary between 84% and 97%, with the average accuracy of 91.9%. This can not be used as a measure of the accuracy of classification of test images because the classifiers are trained using the extracted feature vectors of the training set, but what is classified for the test image are the centroids of a segmented image. The final results depend not only on the feature extraction method, but also on the segmentation of the test images.

B. Image segmentation

The main reason for the difficulties of precise image segmentation is the texture and color variation of the objects in the image. If the image contains only homogenous regions, solutions for this task could use simple clustering methods.

Segmentation of the image is done using the clustering method K-means. This method iteratively partitions data in a predefined number of clusters, K, and the algorithm stops when clusters converge or after a predefined number of iterations. We chose 250 iterations although most of the clusters converge after much smaller number of iterations. As a measure of similarity, the L1 norm was used, instead of usual L2 norm, for lower computational complexity.

At this level of development, the algorithm does not incorporate information about spatial layout of the image, so the obtained segments are not necessarily contiguous.

Because K-means is a local searching procedure, the final cluster centers are dependent on the choice of initial centers. Having well chosen initial centers is an important part of the clustering. To find good initial centers we used the technique described in [9] that partitions the data set along the data axis with the highest variance. The main idea is to find a partitioning point on the data axis with the highest variance that would minimize the total clustering error. This algotithm iteratively partitions the data set, and as a stopping criterion we used predefined maximum clustering error. The number of clusters for every image is determined by the stopping criterion. The tests show that an average of 15 clusters per image are obtained. It is almost impossible to find a stopping criterion that is best for a large collection of images, so images sometimes may be undersegmented or oversegmented. The centroids obtained by this method are then the initial cluster centers for the K-means algorithm.

The clusters obtained using K-means correspond to the segments in the image.

C. Classification of the segments

The step following image segmentiation is classifying these segments into 10 predefined categories. Every segment belongs to one of the categories: sky, water, trees, rocks, skin, soil, street, grass, urban and other (if the segment does not belong to any of the previous categories).

For the classification part, 9 binary classifiers for the first 9 classes were trained. We used SVMs with radial basis function kernel (1) for that purpose.

$$K(x_i, x_j) = e^{-\gamma \left\| x_i - x_j \right\|^2} . \quad (1)$$

The parameters for the training of the models, the cost C, that controls the tradeoff between allowing training errors and forcing rigid margins, and parameter γ in the kernel function, were obtained by parameter search that maximizes the cross validation accuracy. The five-fold cross validation was used.

The classifiers were trained using the one-vs-all method by which we get every classifier to be able to distinguish between the samples from one class and the samples of all other classes. The training set consists of feature vectors extracted from cropped parts of training images for the corresponding classes, with the same feature extraction method as above. For every class 3000 training samples were used. Because for the training of every classifier we have 3000 positive and 27000 negative samples, weights were included in the training. Both training and test data were normalized using zero mean and unit variance normalization. The normalization parameters were computed for every model separately using the training data. These parameters are later used for normalization of the test data.

The cluster centroid is a representative of the cluster in the classification process. If, for one test sample (a cluster centroid), more than one classifier gives a positive

decision, the class with the highest decision value is chosen for that sample.

D. Postprocessing

The resulting clusters after the clustering phase are not necessarily connected. This is why it is possible to have isolated blocks from one cluster surrounded by other cluster. This often happens at the edges in the image because the color and texture of the blocks positioned there are not similar enoguh to the blocks of the environment. Humans usually percive the decision for these blocks as errors; hence a better result would be if these blocks belonged to the same cluster as their environment.

To improve the classification result and make it appear more correct to human users, a postprocessing is applied to the results. The resulting classes can be „smoothed“ by means of median filtering. The size of the filter is 3x3 blocks.

III. Results and Discussion

The results of this algorithm depend on number of factors.

It is very important to have good training sets. It is best if these sets are disjoint and homogenous. Having such sets is almost impossible because of the natural diversity that can appear in all of the classes, but also because of the human uncertainity for the label of some training examples.

Other problem related to the training set is the difficulty of having enough training samples for every class, that would represent all possible situations of a scene. For example, having enough samples from every possible architectural style for the class „urban“ and having samples of buildings taken at different distances.

Very important factor, also, is the image segmentation. It is important to choose the correct number of clusters to avoid undersegmentation and samples from different classes to belong to one cluster, and, also, to avoid oversegmentarion by which we will have small insignificant parts of the image consisting a separate segment.

Fig. 1 shows the colors we used for marking the segments belonging to the corresponding classes.

Fig. 2 is an example of correct classification of the segments. When the content of the image is similar by nature to that of the images in the training set we have good classification results. In this example we can see how segmentation highly impacts the classification. We can notice the red area below the tree in bottom-right picture. If we look at the same location in the segmented image we can notice that this area belongs to the same cluster with some of the blocks from the building. Thus, it was labeled as „urban“.

For comparison purpose, Fig. 3 (left) shows the segmentation results of the algorithm in [10]. It should be pointed out that the method proposed in [10] works on pixel level, and the proposed algorithm is block based. As can be seen on Fig. 3, the results are similar. However, the proposed algorithm has lower computational complexity.

Fig. 1. Legend of colors for marking the classes

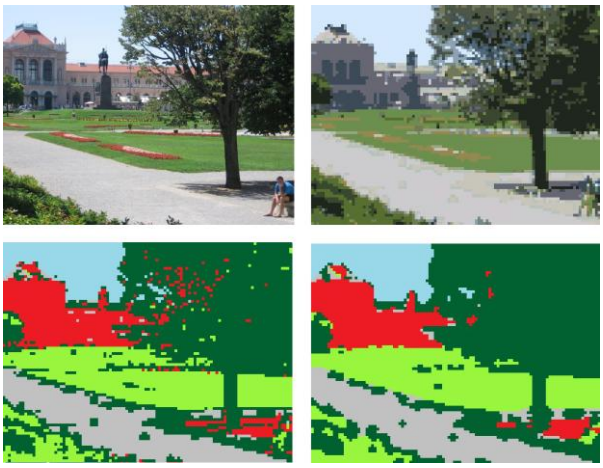

Fig. 2. Top left - original image, top right - segmented image, bottom left - the image after classification of the segments, bottom right - the result of post processing

Fig. 3. Comparison between mean shift (left) and k-means (right) segmentation

Some more examples of relatively correct classification are shown in Fig. 4.

Fig. 5 is an example of incorrect classification. In this example we can notice the problem of good separation of the classes. The regions with smooth gray texture were classified as „street“ (gray) because this description covers the training samples of the class „street“, so the algorithm does not „know“ that in this case those regions are part of a building. Similar explanation holds for the regions labeled as „rocks“ (yellow). Only the segments which have similar features to the training set of „urban“ were classified correctly (with the red color).

More examples of incorrect classification are given on Fig. 6. Here we can see how similarity of the classes (for example sometimes grass and trees or walls made of rocks) leads to classification errors. We can also see how the position of the objects in the image can be of big importance for the classification. For example, if the object in the test image is much closer to us than in the images of the training set, the classifiers would probably not

recognize it correctly. We can also see the problem of predicting all possible situations of scenes while creating the training set (the example in the third row of Fig. 6).

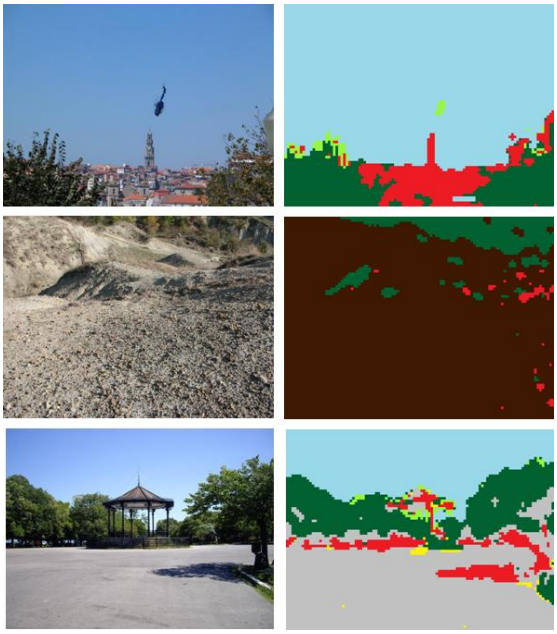

Fig. 4. Examples of correct classification

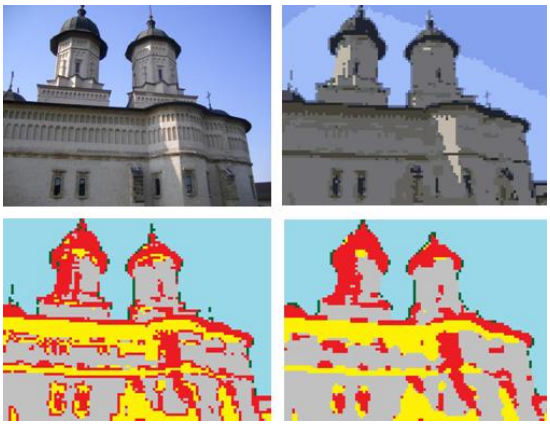

Fig. 5. Top left - original image, top right - segmented image, bottom left - the image after classification of the segments, bottom right - the result of post processing

IV. Concluisions and further imrovements

This algorithm provides good results in cases of scenes that were well defined by the training set. For example, when the objects in the image are in similar distance from the camera, or when the buildings are in the same or similar architectural style as those in the training set. In other situations in which the scenes were not completely predicted in the training set, the results were slightly worse, but still, considering the intention to make a fast algotirhm, they were satisfactory.

The number of operations of the presented algorithm is relatively small, therefore, this algorithm could be called „fast".

Including the information about the spatial appearance of the classes can improve classification results. An example of this is that sky must always be above the ground or water.

Adding spatial information about the blocks in the image can improve the segmentation and thus the classification results. Being closer in the image would mean the blocks are more similar. This would provide results that are more similar to the way humans understand segments.

Fig. 6. Examples of incorrect classification

Making changes in the training set could lead to better classification. By using labeled centroids of segments from the images in the training set, the training set would be more similar to the test set.

References

[1] Luo, Ming, Ma, Yu-Fei and Zhang, Hong-Jiang., "A Spatial Constrained K-Means Approach to Image Segmentation." Fourth International Conference on Information Communications and Signal Processing, 2003.

[2] Sural, Shamik, Gang, Quian and Sakti, Paramanik., "Segmentation and Histogram Generation Using the HSV Color Space for Image Retrieval." 2002. International Conference on Image Processing.

[3] Andrews, S, Tsochantaridis, I and Hoffmann, T., "Support vector machines for multiple-instance learning." 2003. Advances in Neural Information Processing Systems 15.

[4] Maron, O and Ratan, A L., "Multiple-instance learning for natural scene classification." s.l. : Morgan Kaufmann Publishers Inc, 1998. The Fifteenth International Conference on Machine Learning.

[5] Zhang, Q and Goldman, A S., "EM-DD: An improved multiple-instance learning technique." Advances in Neural Information Processing Systems, 2002.

[6] Chen, Yixin and Wang, James Z., "Image Categorization by Learning and Reasoning with Regions." s.l. : Journal of Machine Learning Research, 2004, Vol. 5.

[7] Dalal, N. and Triggs, B., "Histograms of Oriented Gradients for Human Detection." San Diego, CA, USA : IEEE Computer Society, 2005. Computer Vision and Pattern Recognition. 1063-6919.

[8] Deelers, S and Auwatanamongkol, S., "Enhancing K-Means Algorithm with Initial Cluster Centers Derived from Data Partitioning along the Data Axis with the Highest Variance." 2007. Proceedings of world academy of science, engineering and technology volume 26.

[9] Rifkin, Ryan and Klautau, Aldebaro., "In Defence of One-vs-All Classification." Journal of Machine Learning Research, January 2004.

[10] Comaniciu, Dorin and Meer, Peter., "Mean shift: A robust approach toward feature space analysis." s.l. : IEEE Computer Society, 2002. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 24.