SPEEDING-UP THE SVM-BASED COLOR IMAGE CLASSIFIERS BY USING VECTOR QUANTIZATION

Mihaela Gordan*, Costin Miron*, Ioan Stoian, Ovidiu Dancea**, George Tepes***

**Basis of Electronics Department, Technical University of Cluj-Napoca*
{mihag,miron,george}@bel.utcluj.ro

Constantin Daicoviciu Street, No. 15, Cluj-Napoca, RO-3400, Romania
*** S.C. IPA S.A. CIFATT Cluj, Republicii 109, 3400 Cluj-Napoca, Romania*
{stoian,ipacluj}@automation.ro

Abstract: Support vector machines (SVMs) represent a new powerful technique with remarkable results in image analysis, especially in object detection and localization in images. In many SVM-based applications, the object recognition and localization task in digital images is treated as a pattern classification problem, and the size of the feature vector describing the object to be recognized is in general large. The large size of the feature vector leads to a long computational time required for object localization in typical sized images; therefore attempts are made both in software and hardware implementations of SVMs and of their applications to reduce the amount of computations. In this paper we propose a faster method for color image analysis using SVMs, by combining them with vector quantization. The results shown prove that quantized color images can be used successfully for object recognition and localization in color images, given error rates comparable with the SVMs applied on the original RGB images, while providing the advantage of a double classification speed.

Key words: support vector machine, vector quantization, object recognition, color image analysis, fast implementation

1. INTRODUCTION

SVMs represent a powerful and rather novel technique, which was recently successfully applied in various computer vision fields, mainly in visual object localization and classification tasks. Example of such applications include: face recognition, face localization, speaker identification, object tracking, medical image processing, mouth localization [1,2,3,4]. Some of these applications use gray level images only, whereas others use color images for object localization and classification.

The main operation, which dominates the classification phase is the evaluation of a kernel function on the incoming vector and each of the support vectors. Due to the relatively large number of incoming vectors used to describe the image to be analyzed, the number of kernel evaluations (even for a relatively small number of support vectors) strongly affects the computational time, which increases proportionally with the number of support vectors, the amount of test data and the feature vector size of each data.

There were several attempts presented in literature [5,6] to reduce the computational time required by the image analysis phase with SVMs, especially through hardware implementation, feature space transforms, but these solutions do not always keep the quality of classification results and sometimes they even may add extra-time due to the feature extraction process. Many times, raw data taken from the region of interest (ROI) is enough to give a good classification/ localization rate [4]. When color is available, it can improve the classifier's performance as compared to the use of luminance only, but the cost is the dramatic increase of the size of the feature vector (three times larger than in case of using luminance only).

In this paper we propose a solution to speed up the classification phase in SVM-based color image analysis, by applying an optimal vector quantization (VQ) to the entire dataset (which is, in our case the whole image). This allows us to represent every image pixel by one of the 256 possible codewords instead of 2^{24} possible color values. Then, every dot product between two vectors in the classification phase, can be replaced by a set of memory reads and additions, eliminating completely the multiplications. As we will prove in this paper, this leads to a double speed of the classification phase as compared to the standard SVM applied on RGB color images.

2. OVERVIEW OF SVMs

The Support Vector Machines were designed by Vapnik [7], and exploit the principles of structural risk minimization (SRM). The goal of statistical learning being generalization, SVMs are perfectly adapted to perform this task, due to minimization of an upper bound on the expected risk, as opposed to ERM that minimizes the error of the expected data.

The principle of SVM classifiers is described as follows: SVM receive at the input a set of training vectors as pairs: $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \Re^n$ is the training example, and $y_i \in \{-1,+1\}$ is the label. The SVM will derive a hyperplane which ideally will perfectly separate the positive and negative training examples and maximize the distance of the closest positive and closest negative training example to the hyperplane. When the training examples are not separable, the constraints which impose the perfect separation of the positive and negative training examples must be relaxed by adding a penalty term to the function to be minimized, proportional to the sum of errors by a factor *C*. *C* is a parameter of the SVM.

When the data is separable in their original domain, the hyperplane will perfectly separate the training data with no error. When the data is not linearly separable in the original domain, it is projected by the SVM in a higher dimensional space through a nonlinear mapping $\Phi : \Re^n \rightarrow \Re^m, m > n,$ where it becomes linearly separable. The optimal hyperplane, derived in $\Re^m$, is able to perfectly separate the data and ensure a large distance from the nearest positive and negative example to the hyperplane.

The real valued decision function of the classifier will be of the form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \,, \tag{1}$$

where *N* is the number of support vectors of the SVM, x_i are the support vectors (i.e. the subset of training examples which define the hyperplane), α_i are their associated Lagrange multipliers, *b* is the free parameter of the hyperplane, and $K(x,x_i)$ is a kernel function implementing the dot product of the vectors x and x_i in $\Re^m$, thus providing the advantage that the explicit form of the nonlinear mapping Φ doesn't need to be known.

So far, several form of kernel functions are employed in SVM classifiers, and successfully been used for image recognition applications. The most frequently used kernel functions are the polynomial kernel and the RBF kernel.

3. OVERVIEW OF VECTOR QUANTIZATION AND ITS APPLICATION TO COLOR REDUCTION

Vector quantization (VQ) is a lossy data compression method based on the principle of block coding. In 1980 Linde, Buzo, and Gray (LBG) proposed a VQ design algorithm based on a training sequence. A VQ that is implemented using this algorithm is referred to as an LBG-VQ. The VQ design problem can be stated as follows. Given a vector source with its statistical properties known (in our case the source vector consists of the image pixels, in fact their R, G and B values, read from the color histogram), given a distortion measure, and given the number of training vectors (total number of pixels in the image), find a codebook and a partition which results in the smallest average distortion among the training vectors set.

Assume the training sequence consists of ***M*** *k*-dimensional source vectors, $T = \{x_1, x_2, ..., x_M\}$, $x_m = (x_{m,1}, x_{m,2}, ..., x_{m,k})$, $\mathrm{m} = 1,..,\mathrm{M}$. Let N be the number of codevectors. $C = \{c_1, c_2, ..., c_N\}$ represents the codebook of N k-dimensional codevectors, $c_n = (c_{n,1}, c_{n,2}, ..., c_{n,k})$, $\mathrm{n} = 1,..,\mathrm{N}$. Let S_n be the encoding region associated with c_n and $P = \{S_1, S_2, ..., S_n\}$ denote the partition of space. If the source vector x_m is in the encoding region S_n, then its approximation ($Q(x_m)$) is c_n.

The average distortion is given by:

$$D_{ave} = \frac{1}{Mk}\sum_{m=1}^{M}\left\|x_m - Q(x_m)\right\|^2 , \tag{2}$$

where $\|e\| = e_1^2 + e_2^2 + ... + e_k^2$. So, the design problem can be stated as follows: given T, and N, find C and P such that D_{ave} is minimized. If C and P are the solutions for the above minimization problem, they must satisfy the following optimality criteria:

a) nearest neighbor condition: $S_n = \left\{x : \|x - c_n\|^2 \le \|x - c_{n'}\|^2, \forall n' = 1,2,...,N\right\}$; meaning that the encoding region S_n consists of all vectors that are closer to c_n that any of the other codevectors.

b) centroid condition: $c_n = \dfrac{\sum_{x_m \in S_n} x_m}{\sum_{x_m \in S_n} 1}$, $n = 1,2,...,N$; this imposes the codevector to be the average of all those training vectors that are in encoding region.

The LBG-VQ design algorithm is an iterative algorithm which alternatively solves the above two optimality criteria. The algorithm requires an initial codebook, obtained by the *splitting* method. More information about VQ can be found in [8].

4. APPLYING SVM TO FACE FEATURE LOCALIZATION IN COLOR IMAGES – A SIMPLE APPROACH

The strategy used for applying the SVMs in facial feature localization attempts to formulate this problem as a pattern recognition problem, as follows: considering known a-priori the approximate size of the wanted facial feature in a set of facial images (e.g. the eye), described by its width W_m and height H_m, we can define a bounding rectangle comprising it of size W_b x H_b as close as possible to W_m x H_m, with $W_b \geq W_m$ and $H_b \geq H_m$. Then, the entire facial image can be decomposed in partially overlapping rectangular regions of size W_b x H_b with an arbitrarily chosen step. Let us denote the total number of these partially overlapping bounding rectangles for a facial image by P. In the case of a color facial image, each of the P subimages of size W_b x H_b in which the facial image is decomposed can be represented by a three-dimensional vector of size W_b x H_b, comprising the three primary color components (R,G,B) of the pixels in the subimage scanned in row order, from top to bottom. Let us denote each such a three dimensional vector by $\mathbf{x}_i$, $\mathbf{x}_i = \left[x_{i1R}, x_{i1G}, x_{i1B}, ..., x_{iW_bH_bR}, x_{iW_bH_bG}, x_{iW_bH_bB}\right]^T$, i=1,...,P, where x_{ijR} denotes the red component of the pixel j from the subimage i, j=1,...,W_bH_b , x_{ijG} –the green component, and x_{ijB} – the value of blue component.

With these notations, each facial image may be viewed as being represented by a set of patterns $\{\mathbf{x}_i\}$, i=1,...,P. In order to localize the eye in a facial image, having available the decomposition of the facial image into eye and non-eye patterns whose location can be determined if the set of patterns is an ordered set, it is enough to identify which patterns $\mathbf{x}_k \in \{\mathbf{x}_i\}$, i=1,...,P, are in the category of eye patterns and to locate them on the facial image. The second task, of localizing these patterns on the facial image, is an easy task once the order of the patterns in the set $\{\mathbf{x}_i\}$, the overlapping step and the size W_b x H_b of the bounding rectangle are known.

5. THE PROPOSED STRATEGY TO JOINTLY USE VQ AND SVMs FOR SPEEDING UP THE LOCALIZATION PHASE

For a digital facial color image represented in the (R,G,B) space, in which some facial feature localization, e.g. the eye, is desired, using the strategy presented in the previous section, a set of test data vectors denoted by $\{\mathbf{x}_{tst}\}$ will describe the image, $\mathbf{x}_{tst}$ of length $3W_bH_b$ according to the notations from Section 4. Assuming a trained SVM available for the classification task in eye vs. non-eye classes, described by the set of support vectors $\{\mathbf{x}_i\}$, i=1,...,N_S, the result of the classification will be the real value given by: $f(\mathbf{x}_{tst}) = \sum_{i=1}^{N_S} \alpha_i K(\mathbf{x}_{tst}, \mathbf{x}_i)$. In the following we will consider that the polynomial kernel was used in the SVM, namely $K(\mathbf{x}_{tst}, \mathbf{x}_i) = (m\,\mathbf{x}_{tst} \cdot \mathbf{x}_i + n)^d$. $\mathbf{x}_{tst} \cdot \mathbf{x}_i$ represents the scalar product of these two vectors, so for this operation we will need 3 W_bH_b multiplications and 3 W_bH_b – 1 additions.

If prior to the feature extraction, a VQ is applied on the color image in which the SVM localization is performed, then the size of each test data vector (and also of the training vectors and therefore of support vectors) can be reduced three times, since a single codeword is enough to uniquely describe each (R,G,B) color combination. First, we will quantize the image using the LBG-VQ algorithm [8], and set the codebook length to 256, which is a good compromise between the number of bits required to represent each color (8) and the color quantization error. Thus, in the LBG-VQ quantized image, the support vectors are described as $\mathbf{x}_i = [\mathbf{x}_{i1}\ \mathbf{x}_{i2}\ \ldots\ \mathbf{x}_{iWbHb}]$, where each $\mathbf{x}_{ik}$ encodes an $(r_{ik}, g_{ik}, b_{ik}) \in \{R_{VQ}, G_{VQ}, B_{VQ}\}$ and similarly the test vectors $\mathbf{x}_{tst} = [\mathbf{x}_{tst1}\ \mathbf{x}_{tst2}\ \ldots\ \mathbf{x}_{tstWbHb}]$. R_{VQ} , G_{VQ}, B_{VQ} are obtained after vector quantization and represent the sets of red, green and blue values which gives us the 256 colors in the quantized image, therefore each $\mathbf{x}_{ik}$ is in fact a codeword (an index) from the corresponding image codebook.

In the case of the original (non-quantized) image, the scalar product between $\mathbf{x}_{tst}$ and $\mathbf{x}_i$ is given by:

$$\sum_{k=1}^{W_b H_b}(r_{tstk} * r_{ik} + g_{tstk} * g_{ik} + b_{tstk} * b_{ik}) \tag{3}$$

Instead, in the VQ color image, $(r_{tstk}, g_{tstk}, b_{tstk}) \in \{(R_{VQ}, G_{VQ}, B_{VQ})\}$ and $(r_{ik}, g_{ik}, b_{ik}) \in \{(R_{VQ}, G_{VQ}, B_{VQ})\}$. Thus the scalar product between any of these two vectors, namely $(r_{tstk} * r_{ik} + g_{tstk} * g_{ik} + b_{tstk} * b_{ik})$, taking only at most $(256^2)/2$ values, can be stored in a matrix of $256x256$. The row indexes can represent the indexes of the components of the support vectors, $\mathbf{x}_{ik}$, and the column indexes can represent the indexes of components of the test vectors, $\mathbf{x}_{tstk}$, k=1,....,$W_b H_b$. The elements of the matrix store the product of these vectors at the corresponding line and column. The matrix can be calculated after the vector quantization phase, and saved in memory. Thus, instead of computing for each vector dot product, according to equation (3), three multiplications and two additions, these operations can be replaced by a single read from the memory.

6. PERFORMANCE EVALUATION

To test the performances of the proposed approach, we implemented in C++ the modules for: (1) optimal vector quantization of (R,G,B) images using LBG-VQ algorithm; (2) scanning the facial image with the rectangular window of user-definable size and scan step; (3) training labels assignment; and (4) final localization of the ROI comprising the eyes on the facial image. For the SVM training and classification, we used the publicly available SVMLight software, integrating the training and classification modules from SVMLight into our application. As test set, we used the color facial image database, from the Informatics and Mathematical Modeling, Computer Vision Laboratory, Technical University of Denmark [10]. The data set comprises 37 still images of 37 different frontal human faces, all without glasses and with a neutral expression.

The performance was evaluated using 17 subjects from the database for training and the remaining 20 subjects for testing. We evaluated the performance in terms of the classification error and of the computational time required for each approach (original RGB images, and VQ images); thus two SVMs with polynomial kernel of degree 2 and C=1000 were trained on the same training data and tested on the same test data, one using as features the (R,G,B) pixel values and the other - the code vectors generated by VQ, as described in Sections 4 and 5. The comparison of the two SVMs from the classification error point of view is summarized in Table 1 below, and shows acceptable performance for the VQ SVM as well.

Table 1. The classification error in the test set

	Misclassification error	FAR	FRR
SVM on color (R,G,B) images	1.5%	0.6%	0.9%
SVM on VQ images	2.7%	2.4%	0.3%

Estimation of the computational time in the classification phase: The time required to locate these facial features is considerably reduced when using VQ images in the classification phase. This computational time was evaluated in software, considering the processor time in equivalent assembly language for the dot product evaluations and memory reads; thus in the (R,G,B) color image-based SVM, computing $(r_{tstk} * r_{ik} + g_{tstk} * g_{ik} + b_{tstk} * b_{ik})$ takes the equivalent in assembly of (1 fld + 1 fmul + 1 fstp) instructions/multiply * 3 multiplies + (1 fld + 1 fadd + 1 fstp) instructions/add * 2 adds, requiring 145 processor CLK, whereas using VQ and SVM we will need for the same result (4 mov + 1 fild + 1fstp) instructions, requiring 78 CLK only [9]. Figures 1 and 2 show some feature localization with SVMs in a VQ quantized color image and in an original, (R,G,B) color image. The localized patterns are marked by red rectangles.

Figure 1 Eyes classification (VQ image)

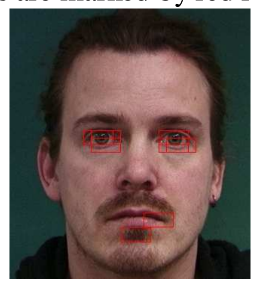

Figure 2. Eyes classification (color image)

7. CONCLUSIONS

We proved that the results obtained applying SVM classifiers over VQ images are comparable with the results obtained using original color images, but the process of classification is improved in terms of speed. In our future work, we will address the software implementation of this proposed method, and speed optimization for even more facial features. Also, we plan to verify the functionality of the proposed method on different fields of object localization in color images.

8. REFERENCES

[1] Chodorowski, A., Gustavsson, T., Mattsson, U., "Support vector machines for oral lesion classification", *Proc. IEEE Int. Symp. on Biomedical Imaging* 2002, pp. 173–176
[2] T.J. Terrillon, M. N. Shirazi, M. Sadek, H. Fukamachi, S. Akamatsu, "Invariant face detection with support vector machines", *Proc.15th International Conference on Pattern Recognition*, Barcelona, Spain, 2000, Vol. 4, pp. 210-217
[3] I. Buciu, C. Kotropoulos, I. Pitas, "Combining support vector machines for accurate face detection", *Proc. IEEE Int. Conf. Image Processing*, Greece, 2001, pp. 1054-1057
[4] M. Gordan, C. Miron, G. Ţepeş, S. Pop, "Application of Support Vector Machine Classifiers for Mouth Localization in Facial Images*", Proc. of the Symposium on Electronics and Telecommunications*, Timişoara, Romania, Sept. 2002
[5] J. Platt, "Fast Training of Support Vector Machines Using Sequential Minimal Optimization," in B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods— Support Vector Learning,* pp. 185-208, Cambridge, MA: MIT Press, 1999
[6] R. Genov, S. Chakrabartty, and G. Cauwenberghs "Silicon Support Vector Machine with On-Line Learning," *International Journal of Pattern Recognition and Artificial Intelligence,* World Scientific, 2003.
[7] V. N. Vapnik, *Statistical Learning Theory,* J. Wiley, N.Y., 1998
[8] Linde, Y., Buzo, A., and Gray, R. M., (1980). "*An algorithm for vector quantizer design*", IEEE Trans Commun COM-28, 84-95
[9] www.intel.com/design/intarch/MANUALS/273021.htm-*Embedded Intel486(TM) Processor Family Developer's Manual*
[10] *** http://www.imm.dtu.dk/~aam/