Content Based Color Image Classification using SVM

Saurabh Agrawal
Department of Electrical Engineering, Indian Institute of Technology Kanpur, India
saurabh.agrawal.11486@gmail.com

Nishchal K Verma
Department of Electrical Engineering, Indian Institute of Technology Kanpur, India
nishchal@iitk.ac.in

Prateek Tamrakar
Department of Electrical Engineering, Indian Institute of Technology Kanpur, India
prateek.tamrakar@gmail.com

Pradip Sircar
Department of Electrical Engineering, Indian Institute of Technology Kanpur, India
sircar@iitk.ac.in

Abstract

We propose a novel approach for content based color image classification using Support Vector Machine (SVM). Traditional classification approaches deal poorly on content based image classification tasks being one of the reasons of high dimensionality of the feature space. In this paper, color image classification is done on features extracted from histograms of color components. The benefit of using color image histograms are better efficiency, and insensitivity to small changes in camera view-point i.e. translation and rotation. As a case study for validation purpose, experimental trials were done on a database of about 500 images divided into four different classes has been reported and compared on histogram features for RGB, CMYK, Lab, YUV, YCBCR, HSV, HVC and YIQ color spaces. Results based on the proposed approach are found encouraging in terms of color image classification accuracy.

Keywords: Support Vector Machine, color image histogram, image classification.

1. Introduction

Image information systems are becoming increasingly important with the advancements in broadband networks, high-powered workstations etc. Large collections of images are becoming available to the public, from photo collection to web pages, or even video databases. Since visual media requires large amounts of memory and computing power for processing and storage, there is a need to efficiently index and retrieve visual information from image database. In recent years, image classification has become an interesting research field in application. Efficient indexing and retrieval of large number of color images, classification plays an important and challenging role.. The main focus of this research work is devoted to finding suitable representation for images and classification generally requires comparison of images depending on the certain useful features.

2. Image features

A number of image features based on color and texture attributes have been reported in literature. Although quantifying their discrimination ability to classification problem has not been so easy. Among the many possible features for classification purpose, extracted from an image, we restrict our self to ones which are global and low level features. The simplest way to represent an image is to consider its bitmap representation. More detail is provided ahead.

2.1. Features from Color Histograms

In image processing, a color histogram is a representation of the distribution of colors in an image. For digital images, it is basically the number of pixels that have colors in each of a fixed list of color ranges that span the image color space, the set of all possible colors. Color histogram technique is a very simple and low level method and in practice it has shown good results [2] especially for image indexing and retrieval tasks, where similar (not necessary identical) images are to be retrieved and easy feature extraction. This also ensures full translation and rotation invariance in the color images under classification task.

A color is represented by a three dimensional vector corresponding to a position in a color space. This leaves us to select the color space and the quantization steps in this color space. As a color space, first we chose the RGB (Red-Green-Blue). The reason for this choice of RGB is that it is widely used in the literature. For the sake of comparison and completeness, we have considered other color spaces also, while keeping other conditions identical. An explanation for this fact is that, after quantization into bins, no information about the color space is used by the classifier.

A color histogram is a representation of the distribution of colors in an image. For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges that span the

978-0-7695-4367-3/11 $26.00 © 2011 IEEE
DOI 10.1109/ITNG.2011.202

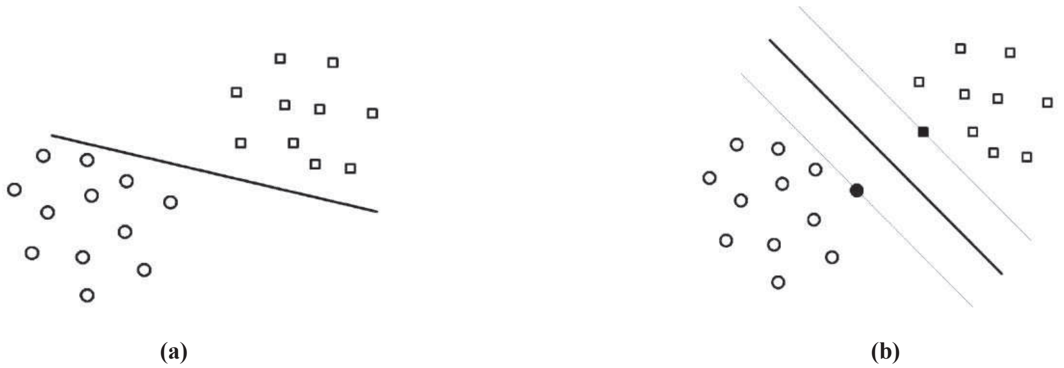
(a) (b)

Figure 1: (a) Separating Hyperplane, **(b)** Optimal Separating Hyperplane

image's color space, the set of all possible colors. The color histogram can be built for any kind of color space, although the term is more often used for three-dimensional spaces like RGB or HSV. For monochromatic images, the term intensity histogram is used.

3. Support Vector Machines (SVMs)

SVMs [1] are learning systems that use a hypothesis space of linear functions in a hyper space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. The aim of Classification via SVM is to find a computationally efficient way of learning good separating hyper planes in a hyperspace, where 'good' hyper planes mean ones optimizing the generalizing bounds and by 'computationally efficient' we mean algorithms able to deal with sample sizes of very high order.

3.1. Optimal separating hyperplanes

Let $(\mathbf{x}_i, y_i)_{1 \le i \le N}$ be a set of N training examples, each example $\mathbf{x}_i \in \Re^d$ belongs to a class labeled by $y_i \in \{-1,1\}$, where d being the dimension of the feature space. The aim is to define a hyper plane which divides the set of examples such that all the points with the same label are on the same side of the hyperplane (Fig. 1). This amounts to finding $\mathbf{w}$ and b so that

$$y_i(\mathbf{w}.\mathbf{x}_i + b) > 0, i = 1, \ldots, N \tag{1}$$

If there exists a hyperplane satisfying eq. (1), the set is said to be *linearly separable*. In this case it is always possible to rescale $\mathbf{w}$ and b so that

$$\min_{1<i<N} y_i(\mathbf{w}.\mathbf{x}_i + b) \ge 1, i = 1, \ldots, N$$

i.e. so that the closest point to the hyper plane has a distance of $1/\|\mathbf{w}\|$. Then Eq. 1 becomes

$$y_i(\mathbf{w}.\mathbf{x}_i + b) \ge 1 \tag{2}$$

Among the separating hyperplanes, the one for which the distance to the closest point is maximal is called *optimal separating hyperplane* (OSH). Since the distance to the closest point is $1/\|\mathbf{w}\|$, finding the OSH amounts to minimizing $\frac{1}{2}\|\mathbf{w}\|^2$ under constraints (2).

The quantity $2/\|\mathbf{w}\|$ is called the margin, and thus the OSH is the separating hyperplane which maximizes the margin. The margin can be seen as a measure of the generalization ability: the larger the margin, the better the generalization is expected to be [4], [5].

Since $\|\mathbf{w}\|^2$ is convex, minimizing it under linear constraints (2) can be achieved with Lagrange multipliers. If we denote by $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ the N non-negative Lagrange multipliers associated with constraints (2), our optimization problem amounts to maximizing

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i . \mathbf{x}_j \tag{3}$$

with $\alpha_i \ge 0$ and under constraint $\sum_{i=1}^{N} y_i \alpha_i = 0$. This can be achieved by the use of standard quadratic programming method [6].

Once the vector $\boldsymbol{\alpha}^0 = (\alpha_1^0, \ldots, \alpha_N^0)$ solution of the maximization problem (3) has been found, the OSH $(\mathbf{w}^0, b^0)$ has the following expansion

$$\mathbf{w}^0 = \sum_{i=1}^{N} \alpha_i^0 y_i \mathbf{x}_i \tag{4}$$

The *support vectors* are points for which $\alpha_i > 0$ satisfy Eq. (2) with equality.

Thus from Eq. (4), the hyperplane decision plane can be written as

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i^0 y_i \mathbf{x}_i . \mathbf{x} + b^0 \right) \tag{5}$$

3.2. Linearly non-separable case

When the data is not linearly separable, slack variables $(\xi_1,\ldots\xi_N)$ are introduced, with $\xi_i \geq 0$ [7] such that

$$y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1 - \xi_i, i = 1,\ldots,N \qquad (6)$$

to allow the possibility of examples that violate Eq. (2). The purpose of the variables ξ_i is to allow misclassified points, which have their corresponding $\xi_i > 1$. Therefore $\sum \xi_i$ is an upper bound on the number of training errors. The generalized OSH is then regarded as the solution of the following problem: minimize

$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{i=1}^{N}\xi_i \qquad (7)$$

subject to constrains (6) and $\xi_i \geq 0$. The first term is minimized to control the learning capacity (as in the separable case) and the second term is to control the misclassified points. The parameter C is chosen by the user. Larger value of C corresponds to assigning a higher penalty to errors.

SVM training requires fixing C (the penalty term for misclassification) in Eq. (7). While dealing with image data, generally the dimension of input space is large compared to the size of the training set, so that the training data is generally linearly separable. Consequently, the value of C has less impact on performance.

3.3. Non Linear Support Vector Machines

The input data is mapped into a high-dimensional feature space through some nonlinear mapping chosen a priori [1]. In this feature space, the OSH is constructed.

If we replace $\mathbf{x}$ by its mapping in the feature space $\Phi(\mathbf{x})$, Eq. (3) becomes:

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j) \qquad (8)$$

If we have $K(x_i, x_j) = \Phi(x_i).\Phi(x_j)$, then only K is needed in the training algorithm and the mapping Φ is never explicitly used. Conversely, given a symmetric position kernel $K(\mathbf{x},\mathbf{y})$, Mercer's theorem [1] indicates us that there exists a mapping Φ such that

$$K(\mathbf{x},\mathbf{y}) = \Phi(\mathbf{x}).\Phi(\mathbf{y}) \qquad (9)$$

Once the kernel K satisfying Mercer's condition has been chosen, the training algorithm consists of minimizing

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) \qquad (10)$$

and the decision function becomes

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x}_i,\mathbf{x}) + b\right) \qquad (11)$$

3.4. Multi class classification

Support Vector Machines are designed for binary classification. When dealing with several classes, as in object recognition and image classification, one needs an appropriate multi-class method. Different possibilities include:

- Modify the design of the SVM, as in [3], in order to incorporate the multi-class learning directly in the quadratic solving algorithm.
- Combine several binary classifiers: "One-against-One" (OAO) [7] applies pair wise comparisons between classes, while "One-against-All" (OAA) [8] compares a given class with all the others put together.

In the "One-against-All" algorithm, n hyperplanes are constructed, where n is the number of classes. Each hyperplane separates one class from the other. In this way, we get n decision functions $(f_k)_{1\leq k\leq n}$ of the form (5) as given in eq. (12). The classes of a new point x is given by arg max$_k$ $f_k(x)$, i.e. the class with the largest decision function.

$$\min_{\mathbf{w}^i, b^i, \xi^i} \frac{1}{2}(\mathbf{w}^i.\mathbf{w}^i) + C\sum_{j=1}^{l}\xi_i^j \qquad (12)$$

In the "One-against-One" algorithm, $n(n-1)/2$ hyperplanes are constructed, where n is the number of classes. Each hyperplane separates only two classes. In this way, we get decision functions of the form (5) as given in eq. (13).

$$\min_{\mathbf{w}^{ij}, b^{ij}, \xi^{ij}} \frac{1}{2}(\mathbf{w}^{ij}.\mathbf{w}^{ij}) + C\sum_{t}\xi_t^{ij} \qquad (13)$$

In Directed Acyclic Graph SVM (DAG) proposed in [9]. Its training phase is the same as the "One-against-One" method by solving $n(n-1)/2$ binary SVMs. However, in the testing phase, it uses a rooted binary directed acyclic graph which has $n(n-1)/2$ internal nodes and n leaves.

The assumption here is that every image has a single label i.e. every image belongs to only one class. However, in image classification, an image may belong to several classes as its content is not unique. It would be possible to make multiclass learning more robust, and extend it to handle multi-label classification problem.

4. Implementation scheme and case study

To implement the above technique, we first find the histogram for images. The number of pixels in each bin is referred to as one feature. These features are then stored in a matrix in row by row fashion i.e. each row contains the histogram corresponding to one image. Class of each image is appended with each row. For validation of the implemented code we use *k*-fold cross-validation algorithm [10]. In *k*-fold cross-validation, the complete dataset is divided into k subsamples. One subsample is used for validation (testing) and (*k*-1) subsamples are used for training. This is repeated k-times so that each subsample is used for validation. The *k*-accuracies are then averaged to get the actual accuracy. Some of the images used are shown below:

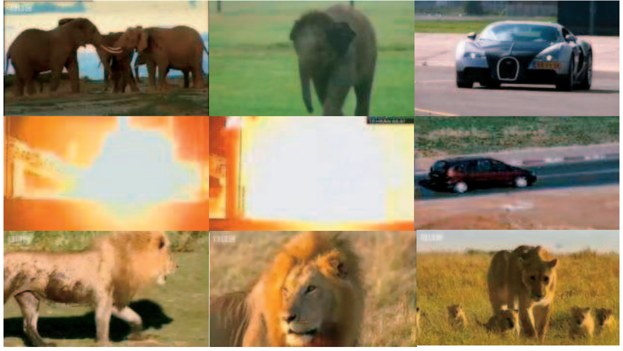

Figure 2: Images used to verify the implementation

The training of SVM is done by $n \times (d+1)$ matrix where n is the number of training data is and d is the number of features. Last column contains the class of each data. The image database contains around 575 images divided into four different classes. The training matrix is as follows:

$$M_{TR} = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,d} & c_1 \\ f_{2,1} & f_{2,2} & \cdots & f_{2,d} & c_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_{(n-1),1} & f_{(n-1),2} & \cdots & f_{(n-1),d} & c_l \\ f_{n,1} & f_{n,2} & \cdots & f_{n,d} & c_l \end{bmatrix}$$

Similar is the test matrix M_{TS} .

5. Results

Table 1 shows the classes assigned to different types of images used.

Table 1: Classes assigned to different types of images

Image Classes	Class Assigned
Car	1
Fire	2
Elephant	3
Lion	4

The accuracies are calculated for “OAA”, “OAO” and DAG-SVM for different histogram levels and for different color spaces, using *5*-fold cross-validation.

5.1. Results with changing histogram level

For *RGB* image, the histogram level is changed and the accuracies are tabulated in Table 2. It is to be noted that histogram level *N* corresponds to the data with a dimension of 3*N* (i.e. *N* for each R, G and B image).

Table 2: Accuracies for OAA, OAO and DAG classifier with changing histogram level

Histogram Level (*N*)	Accuracy		
	OAA	OAO	DAG
2	88.23%	89.08%	65.72%
5	82.10%	82.03%	61.14%
10	79.63%	80.20%	68.39%
15	77.54%	79.52%	72.42%
20	83.26%	76.11%	70.34%
25	87.61%	68.26%	71.26%

5.2. Results with different color spaces

Here we keep the histogram level fixed and change the color space. Histogram of 10 is used for all the color spaces and the accuracies are tabulated in Table 3.

Table 3: Accuracies for OAA, OAO and DAG classifier for different color spaces (histogram level=10)

Color Space	Accuracy		
	OAA	OAO	DAG
RGB	79.63%	80.20%	68.39%
CMYK	76.29%	76.20%	71.29%
Lab	78.26%	79.86%	72.22%
YUV	80.27%	79.63%	69.82%
YC_BC_R	84.39%	81.25%	68.76%
HSV	74.47%	80.39%	62.37%
HVC	81.12%	78.93%	65.27%
YIQ	82.11%	77.39%	71.15%

6. References

[1]. Christopher J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Mining and Knowledge Discovery* 2, pp.121-167, 1998.

[2]. M. Swain and D. Ballard, “Indexing via color histograms,” *International Journal of Computer Vision*, Vol. 7, pp. 11–32, 1991.

[3]. J. Weston and C. Watkins, “Multi-class support vector machines,” *Technical Report CSD-TR-98-04*, Royal Holloway, University of London, 1998.

[4]. V. Vapnik, *Statistical Learning Theory,* John Wiley, New York, 1998.

[5]. P. Bartlett and J. Shawe-Taylor, “Generalization performance of support vector machines and other pattern classifiers” in Advances in Kernel Methods-Support Vector Learning, 1998, MIT Press-Cambridge, USA.

[6]. M. Bazaraa and C. M. Shetty, *Nonlinear programming*, John Wiley, New York, 1979.

[7]. M. Pontil and A. Verri, “Support vector machines for 3-d object recognition,” in Pattern Analysis and Machine Intelligence, June 1998, vol. 20.

[8]. V. Blanz, B. Sch¨olkopf, H. B¨ulthoff, C. Burges, V. Vapnik, and T. Vetter, “Comparison of view-based object recognition algorithms using realistic 3d models,” in Artificial Neural Networks- ICANN’96, Berlin, 1996, pp. 251–256.

[9]. J. C. Platt, N. Cristianini, and J. Shawe-Taylor, “Largemargin DAG’s for multiclass classification”, Advances in Neural Information Processing Systems, Cambridge, MA: MIT Press, vol.12, pp. 547-553, 2000.

[10]. Mosteller and Frederick, “A *k*-sample slippage test for an extreme population”, Annals of Mathematical Statistics Vol.-19, No.-1, pp.58-65, 1948.