Obscure Bleeding Detection in Endoscopy Images Using Support Vector Machine

Jianguo Liu
Dept. of Mathematics, Univ. of North Texas, Denton, Texas 76203, USA
jgliu@unt.edu

Xiaohui Yuan
Dept of Computer Science and Eng., Univ. of North Texas 76203, Denton, Texas, USA
xyuan@unt.edu

Abstract

Wireless capsule endoscopy (WCE) is a recently established imaging technology that requires no wired device intrusion and can be used to examine the entire small intestine non-invasively. Determining bleeding signs out of over fifty thousand WCE images is a tedious and expensive job by human reviewing. Our goal is to develop an automatic obscure bleeding detection method by employing image color features and support vector machine (SVM) classifier. This detection problem is a binary classification problem. We use SVMs for this problem and a new feature selection procedure is proposed. Our experiments show that SVM can be very efficient and may yield very high accuracy rate, in particular with the new proposed feature selection.

Keywords: Image Classification. Support Vector Machine. Feature Selection.

1 Introduction

Visualization of the small bowel has posed a challenge to gastroenterologists due to the difficulty of physically reaching the small bowel. Traditional gastroscopies can usually visualize the upper part and the lower part of the gastrointestinal tract. A recently established imaging technology, known as wireless capsule endoscopy, has been proven to be the best choice of investigation for visualizing the entire small bowel (see, e.g., [1] and [7]). To carry out this procedure, a capsule with embedded color camera, a wireless transmitter, a battery, and lights is swallowed by a subject. Once activated, this camera will take over 55,000 color images during its 8-hour journey through the digestive tract. The images are continuously transmitted to a storage device worn by this subject. After all the images are collected, a physician will examine the images to see if any of them contains signs of disease, e.g. bleeding, and if there is such a sign, determine where it occurs. This reviewing process usually takes a physician a few hours to complete, the accuracy of which also subject to the experience and concentration.

Detecting the existence of obscure bleeding in a WCE image is mathematically a binary classification problem. Computerized diagnosis could assist physicians to review images and identify possible signs. A well-designed computer aided diagnosis system may finish the classification of the whole set of images in minutes. Among many classification algorithms, such as neural networks, find similar, and decision trees, we focus our attention on SVM. SVM methods were proposed by Vapnik in 1979 ([10], [11]) and have gained popularity in the past two decades. Since then, it has been applied to many problems including text categorization, face detection, and bioinformatics (see, e.g., [6], [9], and [5]). SVM methods have also been applied to medical diagnosis, in particular, for tumor detection in endoscopy color images ([8]). In this work, we employ the SVM method

to bleeding detection. We propose a new color feature extraction method that has been proven effective and efficient. We compare the performance of our classifier using this new feature extraction method with using raw data and a conventional color histogram-based feature extraction method. Also, several kernels, including the linear, polynomial, and radial basis function, are used for comparison. Our numerical experiments show that SVM can be very efficient and yield very high accuracy, in particular with our new proposed feature selection.

This paper is organized as follows. We begin in Section 2 with a brief description of support vector machines. In Section 3 we show how color images can be fed into SVMs, i.e., how to preprocess the data and how to select the features. In particular, we propose a new feature selection which is important to high accuracy and efficiency. Numerical experiment results are then presented in Section 4. We summarize our conclusions and give directions for future research in Section 5.

2 Support Vector Machine

In its simplest (linear) form, an SVM is a hyperplane that separates a set of positive examples by maximizing the class margin. That is, given data points of the form $\{(y_1, x_1), (y_2, x_2), \ldots, (y_l, x_l)\}$, where the y_i is either 1 or -1, a constant denoting the class to which the point x_i belongs. Each x_i is an n-dimensional vector. To train an SVM, a set of x_is are pre-labeled, i.e., the y_i components denote the correct classification which an SVM needs eventually to achieve by searching for a dividing (or separating) hyperplane. This hyperplane takes the form of $w \cdot x - b = 0$, where w is the weight vector and is perpendicular to the separating hyperplane.

In the linearly separable cases, two parallel hyperplanes, i.e., $w \cdot x - b = -1$ and $w \cdot x - b = 1$, are generated so that there are no training samples lie in between and the distance of these two planes are maximized. This can be formularized as a quadratic programming (QP) problem:

$$\min 1/2 \, \|w\|^2, \ \text{subject to} \ y_i(w \cdot x_i - b) \geq 1, \ 1 \leq i \leq l. \tag{1}$$

This QP problem is clearly convex and its dual form is

$$\min 1/2 \, \alpha^T Q \alpha - e^T \alpha, \ \text{subject to} \ y^T \alpha = 0 \ \text{and} \ \alpha \geq 0, \tag{2}$$

where Q is an $l \times l$ matrix with $Q_{ij} = y_i y_j \, x_i \cdot x_j$ and e is the vector of all ones. If α is a solution of the dual problem (2), then $w = \Sigma_{i=1}^{l} y_i \alpha_i x_i$ is a solution of the primal problem (1). Those vectors x_i corresponding to $\alpha_i > 0$ lie on the margin and are called the support vectors. Once (1) or (2) is solved, new items (vectors) can be classified by computing $w \cdot x$ where w is a solution to (1) or from (2) and x is a new instance vector to be classified.

In the non-linearly separable cases, Cortes and Vapnik ([4]) proposed a modification (called the soft margin) to the QP formulation that allows, but penalizes, examples that fall on the wrong side of the decision boundary. Another extension to the non-linear classifiers was proposed by Boser et al. ([2]). A more general form of the QP problem (1) with soft margin and nonlinear classifier is as follows:

$$\min 1/2 \, \|w\|^2 + C\xi^T e, \ \text{subject to} \ y_i(w \cdot \phi(x_i) - b) \geq 1 - \xi_i \ \text{and} \ \xi_i \geq 0, \ 1 \leq i \leq l, \tag{3}$$

where ξ represents the training error and the parameter C adjusts the training error and the regularization term $1/2 \, \|w\|^2$. The function ϕ is a mapping from $\Re^n$ to a higher

dimensional space. Practically, kernel functions are used to perform the mapping. The kernel functions are represented in the product form: $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. Some common kernel functions include

Linear: $k(x_i, x_j) = x_i \cdot x_j$

Polynomial (homogeneous): $k(x_i, x_j) = (x_i \cdot x_j)^d$

Radial Basis Function: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \ \ \gamma > 0$

3 Image Representation and Feature Selection

WCE images are color images with a dimension of 255-by-255. A widely used means of describing color images is the RGB (Red-Green-Blue) space. Three matrices are used to store the intensity of the red, green, and blue colors. The dynamic range of the intensity values is $[0, 255]$. These three matrices are the color components and are denoted by $M1, M2$, and $M3$. The pixel at row i and column j can be denoted by the triplet $(M1(i, j), M2(i, j), M3(i, j))$. Figure 1 shows the RGB color space mapped to a unit cube (X - red, Y - green, Z - blue).

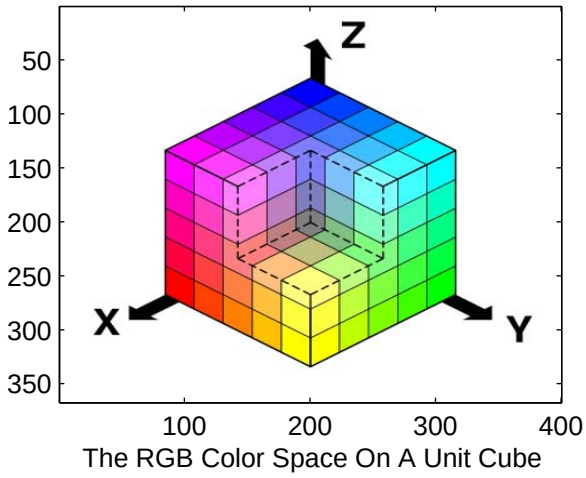

Figure 1: The RBG Color Space Mapped to a Unit Cube

Four sample WCE images are shown in Figure 2. The top row illustrates images without bleeding signs; the bottom row shows two images with bleeding signs. It is clear that although the view is reddish overall, the bleeding region is more saturated. Therefore color is an important feature to detect existence of bleeding.

A naive and straightforward representation of a color image as an input instance for the SVM would be a vector consisting of all the entries of the three RGB matrices, lined up in a row-by-row order. That would result in a vector with 195,075 components (it requires about 1.56MB memory space when each element is represented with eight bytes.) This would be prohibitive when the number of training vectors is large.

To reduce the size of input vectors, we first downsample the images by k ($k = 3, 9, 17, 21, 25$ and 29, respectively). That is, we divide an image into k-by-k blocks and keep only the intensity value at the center of a block. The region of interest excludes the filling pixels, i.e., the pixels outside the circular region of interest (see Figure 2) are

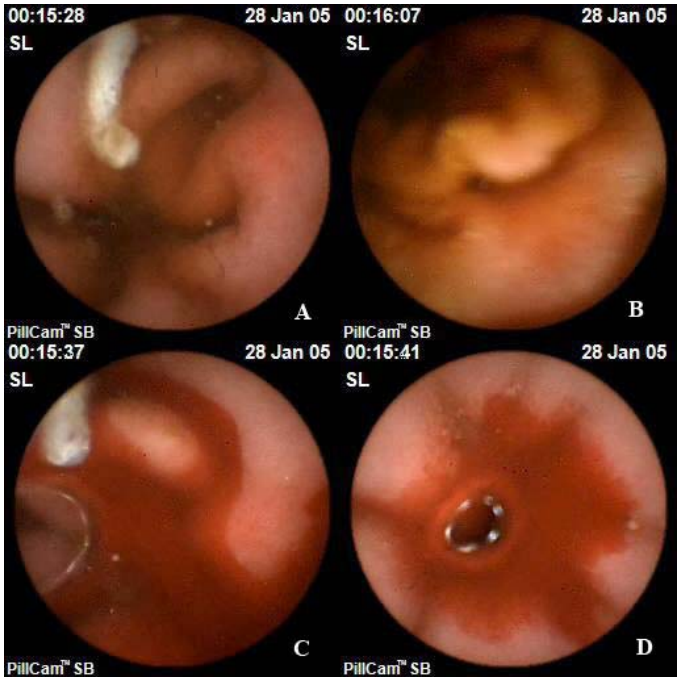

Figure 2: Sample WCE Images. A and B are the non-bleeding samples. C and D are the bleeding samples.

all dark and insignificant. We use only those pixels inside the circle. All the selected pixels are lined up in a row-by-row order, and the entries from all three RGB matrices are used. The downsampling and cropping operations tremendously reduce the data size. For example, the resulting vector from a given image with $k = 3$ will have 13,254 components and require about 106,032 bytes of memory space, which is about 1/15 of the original size. Vectors obtained by this reduction will be called the "Raw Vectors", to be used as input instances for the SVM.

Feature selection (or subset selection) ([12]) is a process commonly used in machine learning, wherein a subset of the features available from the data are selected for application of a learning algorithm. Feature selection is necessary either because it is computationally infeasible to use all available features, or because of problems of estimation when limited data samples (but a large number of features) are present. For some cases, feature selection is critical to classification accuracy and speed.

Color histogram ([13]) is a widely used feature in many applications. For example, In remote sensing, color histograms are typical features used for classifying different ground regions from aerial or satellite photographs. In computer vision, color histograms has been employed to solve the problem of object recognition. Our first feature selection is to use color histogram with k bins ($k = 8, 16, 32, 64, 128$, and 256, respectively). We would not use more bins since our initial tests showed that the classification accuracy would go down when more bins were used. These vectors are "short". For example, when 256 bins are used, a resulting vector will have 768 components. We call these vectors the "Histogram Vectors".

A color histogram is a representation of the distribution of colors in an image, derived by counting the number of pixels of each of given set of color ranges. The histogram provides a compact summarization of the distribution of data in an image, and is invariant

with translation and rotation about the viewing axis. The main drawback of histograms for classification is that the representation is dependent of the color of the object being studied, ignoring its shape and texture.

Our proposed feature selection is based on the special nature of the bleeding detection problem and an observation of the RGB color space showing in Figure 1. To have a detailed description, let again the three RGB matrices be $M1, M2$, and $M3$ for a given image. It is clear that the color WCE images contain colors mainly at the bottom of the color space cube (refer to Figure 1). Hence, the blue color, or the matrix $M3$ representing the intensity of blue, plays a less significant role in classification and can be suppressed. In addition, what we are interested in is whether there is a bleeding. Therefore, what matters is the ratio of the red intensity over the green intensity. In other words, for the ij-th pixel, the ratio $M1(i,j)/M2(i,j)$ determines whether it is likely a bleeding spot or not. Our feature selection is to use these ratios. Given an image, we first downsample it again, calculate the componentwise ratios of $M1$ over $M2$, then sort the ratios into a vector. The resulting vectors have a dimension of 5811 and we call them the "Original Ratio Vectors".

To make a sensible comparison with the histogram vectors, we use ratio vectors with about the same number of components, selected using an evenly spaced manner. For example, let an original ratio vector be v. To compare with the histogram vectors with 768 components (i.e., 256 bins), we use $v = v(4 : 8 : 5811)$; where the number 8 comes from $round(5811/768) = 8$. We will call these vector the "Ratio Vectors".

4 Numerical Test Results

We have 800 color images from the WCE, all have been manually classified by specialists. Four hundreds of them show a sign of bleeding and the other four hundreds contains health regions. From each image, three types of vectors, Raw, Histogram, and Ratio, were generated using the procedure described in Section 3.

The SVM package LIBSVM ([3]) was used for our experiments. (There are several good SVM packages available on the Internet free for academic use. See $http://www.support-vector-machines.org/SVM\_soft.html$ for a list.) We used the default settings of LIBSVM. All the computations were performed on a Dell workstation with dual Xeon CPUs, 8GB memory and running a Linux. Matlab 7.4 was used.

We used the three types of vectors (Raw, Histogram, and Ratio) as the input for the SVM. Three kernels, linear, polynomial, and redial basis function, were used for comparison. For a given combination of vector type and kernel, we had 100 runs and for each run, we randomly selected 80% of the vectors as the training data and the rest 20% as the testing data for accuracy and time analysis. Therefore, about 640 vectors are for training and 160 vectors for testing.

To evaluate accuracy, the two commonly used statistical measures for binary classification, sensitivity and specificity, are used. Sensitivity, or recall rate, is a measure of how well a binary classification test correctly identifies a condition, e.g., picking up on a disease in a medical screening test. A sensitivity of 100% means that the test recognizes all sick people as such. Sensitivity is calculated by

$$\text{Sensitivity} = \frac{\text{number of True Positives}}{\text{number of True Positives} + \text{number of False Negatives}}.$$

where the true positives are those cases that contain bleeding sign and are correctly identified by the classifier. Whereas false negatives are those that also contain bleeding sign

but failed to be correctly identified by the classifier.

Specificity is a measure of how well a binary classification test correctly identifies the negative cases. For example, given a medical test that determines if a person has a certain disease, the specificity of the test to the disease is the probability that the test indicates 'negative' if the person does not have the disease. A specificity of 100% means that the test recognizes all healthy people as healthy. Specificity is defined by the formula

$$\text{Specificity} = \frac{\text{number of True Negatives}}{\text{number of True Negatives} + \text{number of False Positives}}.$$

Our experiment results are reported in the following tables. The number n is the number of components of each vector.

n\Kernels	Linear	Polynomial	Radial
Sensitivity (%)			
141	97.57 (93.51)	97.56 (91.85)	92.72 (84.68)
270	98.72 (94.43)	97.29 (92.23)	87.28 (72.87)
705	99.03 (95.40)	98.12 (92.85)	93.51 (55.95)
1473	99.30 (95.96)	98.73 (93.99)	91.42 (50.81)
4770	99.25 (95.96)	98.68 (90.83)	82.20 (50.0)
13254	99.22 (95.96)	98.72 (90.93)	64.11 (0.0)
Specificity (%)			
141	97.60 (90.91)	95.73 (89.90)	98.67 (94.30)
270	97.76 (92.11)	96.35 (89.06)	99.23 (87.94)
705	98.14 (92.69)	96.89 (89.90)	66.12 (54.89)
1473	98.51 (93.42)	97.25 (93.66)	62.96 (52.14)
4770	98.38 (93.42)	97.12 (93.10)	65.00 (0.0)
13254	98.26 (93.42)	97.06 (93.10)	55.24 (0.0)

Table 1: Accuracy (Mean (and Min) of 100 Runs) Using the Raw Vectors

Table 1, 2, and 3 list the average and the minimum accuracy from our experiments. They are the results using 20% of the sample as testing data and 100 runs. The best average sensitivity result was achieved using ratio vectors and polynomial kernel. The best average specificity result was also achieved using ratio vectors and polynomial kernel. Overall, the employment of radial kernel gave less accuracy then the other two kernels, in particular when the raw and histogram vectors were used. The minimum accuracy results demonstrate a similar trend.

n\Kernels	Linear	Polynomial	Radial
Sensitivity (%)			
24	96.54 (90.41)	97.72 (92.51)	98.04 (93.76)
48	97.26 (90.81)	97.36 (92.44)	98.21 (93.98)
96	98.37 (94.52)	97.40 (92.60)	98.97 (95.98)
192	98.21 (92.78)	96.99 (91.24)	99.64 (97.24)
384	95.46 (90.42)	95.87 (90.59)	81.52 (0.0)
768	92.33 (85.22)	93.63 (86.60)	31.06 (0.0)
Specificity (%)			
24	92.60 (87.09)	97.33 (93.59)	96.14 (90.25)
48	92.75 (87.35)	97.86 (92.62)	96.96 (92.60)
96	93.68 (88.25)	97.92 (93.43)	93.54 (86.97)
192	95.08 (89.41)	97.60 (92.30)	78.13(70.0)
384	95.47 (88.91)	96.79 (89.85)	63.69 (0.0)
768	94.62 (85.92)	95.32 (88.79)	46.03 (0.0)

Table 2: Accuracy (Mean (and Min) of 100 Runs) Using the Histogram Vectors

n\Kernels	Linear	Polynomial	Radial
Sensitivity (%)			
24	100.0 (100.0)	99.73 (98.39)	98.51 (90.09)
48	99.44 (97.0)	99.64 (97.47)	95.87 (85.57)
95	94.18 (88.04)	99.62 (96.83)	98.31 (92.22)
192	95.83 (87.37)	99.42 (96.59)	99.24 (92.77)
379	99.56 (96.12)	98.65 (94.61)	99.23 (95.24)
695	99.64 (96.51)	99.15 (95.56)	98.83 (93.90)
Specificity (%)			
24	86.41 (79.79)	98.89 (95.29)	92.11 (84.45)
48	89.09 (81.48)	99.58 (97.33)	98.35 (92.91)
95	98.54 (94.27)	99.49 (97.11)	97.74 (90.67)
192	98.78 (94.31)	99.17 (95.77)	99.03 (94.60)
379	96.91 (91.77)	98.99 (95.70)	98.93 (96.05)
695	96.17 (96.20)	98.68 (96.20)	99.04 (96.05)

Table 3: Accuracy (Mean (and Min) of 100 Runs) Using the Ratio Vectors

Efficiency is another concern in the application, especially in the real-time scenario. Table 4 and 5 list the average time spent on training an SVM and applying the trained SVM to the testing data. Obviously, the size of the input data affects the overall time. The number of iteration to parameter convergence also plays an important role. The results are the average over 100 runs. Eighty percent of samples (or 640 feature vectors) were used in training. Twenty percent of samples (or 160 feature vectors) were used in testing. The time cost in testing phase (Table 5) was the average time used to classify one testing image. The fastest training was using Ratio vector and polynomial kernel. We are more interested in the time spent in testing. Using Ratio vector, we achieved about two hundredth of a millisecond to finish a classification with an SVM based on linear kernel. Recall that the WCE device takes two images per second. At this rate, our classifier can easily handle real-time precessing requirement.

n\Kernels	Linear	Polynomial	Radial
Raw Vectors			
141	0.0423	0.0357	0.1768
270	0.0659	0.0592	0.3414
705	0.1899	0.1538	0.9187
1473	0.3396	0.3171	1.9636
4770	1.1020	1.0158	6.0138
13254	3.0351	2.7742	16.6591
Histogram Vectors			
24	0.0129	0.0146	0.0245
48	0.0204	0.0185	0.0487
96	0.0341	0.0302	0.1196
192	0.0631	0.0605	0.2616
384	0.1475	0.1615	0.5446
768	0.4362	0.6203	1.3303
Ratio Vectors			
24	0.0138	0.0092	0.0186
48	0.0207	0.0113	0.0211
95	0.0252	0.0147	0.0347
192	0.0418	0.0308	0.0819
379	0.0798	0.0561	0.1959
695	0.1385	0.1132	0.3951

Table 4: Time (Average of 100 Runs) to Train about 640 Vectors (in Seconds)

n\Kernels	Linear	Polynomial	Radial
Raw Vectors			
141	0.1260	0.0974	0.6961
270	0.1762	0.1423	1.1954
705	0.7545	0.5437	3.0558
1473	0.6557	0.6294	6.5463
4770	2.3883	2.3578	21.1762
13254	6.7275	6.5028	58.9883
Histogram Vectors			
24	0.0468	0.0349	0.0751
48	0.1061	0.0617	0.2741
96	0.0919	0.0958	0.3633
192	0.2014	0.2162	0.9465
384	0.3699	0.5044	1.7978
768	0.8481	1.4634	4.9275
Ratio Vectors			
24	0.0558	0.0301	0.0720
48	0.0685	0.0386	0.0825
95	0.0890	0.0170	0.1164
192	0.1397	0.0552	0.2794
379	0.2740	0.1067	0.6738
695	0.6120	0.3335	1.2079

Table 5: Time (Average of about 16,000 Cases) to Classify an Image (in Milliseconds)

To highlight, the best overall combination is the following.

- Best combination: Ratio vectors with 95 components and polynomial kernel — sensitivity is 99.62 (mean) and 96.83 (min); specificity is 99.49 (mean) and 97.11 (min); average time to train an SVM (640 vectors) is 0.0147 seconds; average time to classify an image is 0.017 milliseconds.

5 Concluding Remarks

In this article, we described a new method to detect obscure bleeding sign in WCE images using color feature selection and SVMs. Our proposed method (ratio vectors) for feature selection tremendously reduces the data size without compromising the classification accuracy. The ratio vectors, generated by the new proposed feature selection procedure, yield the best overall accuracy and efficiency. In addition, the ratio vectors do not seem to be sensitive to the choice of the kernels. The trained SVMs were very efficient for identifying bleeding signs, in particular when the polynomial kernel is used.

To make it practically viable, our next step is to improve the sensitivity measure to 100%. We plan to collect a large amount of data and study other image feature selection methods. Extension to other applications, such as tumor detection, will also be explored.

6 Acknowledgement

We thank Robert Kallman for many helpful comments.

References

[1] D.G. Adler and C.J. Gostout, "Wireless Capsule Endoscopy," *Hospital Physician*, pp. 14-22, 2003.

[2] B.E. Boser, I.M. Guyon, and V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Fifth Annual Workshop on Computational Learning Theory*, ACM, 1992.

[3] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, (software available at *http : //www.csie.ntu.edu.tw/˜cjlin/libsvm*), 2001

[4] C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning*, 20, pp. 273-297, 1995.

[5] N. Cristianini and J. Shawe-Taylor, *Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[6] S. Dumais et al.,, "Inductive Learning Algorithms and Representations for Text Categorization," in *Proceedings of the 7th international conference on Information and knowledge management*, pp. 148-155, ACM Press, New York, 1998.

[7] Z. Fireman et al., "Wireless Capsule Endoscopy," *Israel Medical Association Journal*, Vol. 4, pp. 717-719, 2002.

[8] P. Majewski and W. Jedruch, "Endoscopy Images Classification with Kernel Based Learning Algorithms," in *Innovations in Applied Artificial Intelligence*, pp. 400-405, 2005.

[9] E. Osuna, R. Freund, and F. Girosit, "Training Support Vector Machines: an Application to Face Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* , pp. 130-136, 1997.

[10] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

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

[12] Wikipedia, "Feature Selection," $http://en.wikipedia.org/wiki/Feature\_selection$.

[13] Wikipedia, "Color Histogram," $http://en.wikipedia.org/wiki/Color_histogram$.