Computational Imaging 2015

Webcam Classification Using Simple Features

Thitiporn Pramoun,[a] Jeehyun Choe,[b] He Li,[b] Qingshuang Chen,[b] Thumrongrat Amornraksa,[a]
Yung-Hsiang Lu,[b] and Edward J. Delp[b]

[a]*Computer Engineering Department,*
King Mongkut's University of Technology Thonburi,Bangkok, Thailand
[b]*Video and Image Processing Laboratory (VIPER),*
School of Electrical and Computer Engineering,
Purdue University, West Lafayette, Indiana, USA

ABSTRACT

Thousands of sensors are connected to the Internet and many of these sensors are cameras. The "Internet of Things" will contain may "things" that are image sensors. This vast network of distributed cameras (i.e. web cams) will continue to exponentially grow. In this paper we examine simple methods to classify an image from a web cam as "indoor/outdoor" and having "people/no people" based on simple features. We use four types of image features to classify an image as indoor/outdoor: color, edge, line, and text. To classify an image as having people/no people we use HOG and texture features. The features are weighted based on their significance and combined. A support vector machine is used for classification. Our system with feature weighting and feature combination yields 94% accuracy.

Keywords: Image classification, dominant color descriptor , number of edge, gradient magnitude, nearest neighbor, support vector machine

1. INTRODUCTION

Thousands of sensors are connected to the Internet. The "Internet of Things" will contain many "things" that are image sensors [1], [2], [3]. This vast network of distributed cameras (i.e. web cams and surveillance systems) will continue to exponentially grow. We are interested in how these image sensors can be used to sense their environment. How can this ever increasing amount of imagery be interpreted to extract valuable information related to weather forecast, traffic control, environment management, and location identification? Interpreting and learning from such images pose many challenges. In earlier work we described a method for estimating the location of an IP-connected camera (a web cam) by analyzing a sequence of images obtained from the camera [4].

We are also investigating how one would process imagery from thousands of ip-connected cameras. We have at Purdue University been developing the CAM^2 system(Continuous Analysis of Many CAMeras) [**?**], [**?**], [**?**]. CAM^2 is a cloud-based general-purpose computing platform for domain experts to extract insightful information by analyzing large amounts of visual data from distributed sources. CAM^2 uses cloud computing to manage the large amounts of data for better scalability. CAM^2 currently has detected and has access to more than 70,000 cameras deployed worldwide. These include cameras from departments of transportation, national parks, research institutions, universities, and individuals.

In this paper we investigate simple methods of web cam image classification. A method of classifying indoor-outdoor scene using low-level image features was presented in [5]. This method uses four features, histograms in the Ohta color space, multiresolution, simultaneous autoregressive model parameters, and coefficients of a shift-invariant DCT. Each feature is divided into sub-blocks. The sub-blocks are used for classification. In [6], the SIFT descriptor is used to extract the image features. Weighted distance feature vectors are calculated and assigned to each feature. Then the features are classified based on nearest-neighbor using the bag-of-visual words scheme or codebooks. A scene classification method is described in [7] that uses local color features. In this method, the features of block regions and a set of lines in image are calculated using Hough Transform. These

This work was partially supported by the endowment of the Charles William Harrison Distinguished Professorship at Purdue University. Address all correspondence to Edward J. Delp, ace@ecn.purdue.edu

two features are used for classification using “Boosting.” In [8] a method to classify a scene as indoor vs. outdoor is proposed that uses relevant low level features: color and texture to improve the classification performance. The HSV color model is considered as the color feature and DCT coefficients as the texture feature. Entropy is estimated using UV. The K-nearest neighbor was used for classification.

Our image classification method is based on the support vector machine (SVM). It classifies an image as indoor or outdoor and crowd or no-crowd using a set of simple visual features. We consider four different types of features: color, edge and line, and texture for indoor/outdoor, and two different types of feature: Histograms of Oriented Gradients (HOG), and texture features for people/no people. We investigate in total six features to classify an image. These six features consists of one color feature: the dominant color descriptor (DCD), two edge features: the number of edge pixels and gradient magnitude, two line features: the number of vertical and horizontal lines from the Hough transform, and standard deviation of sub-block of luminance.The four features for classifying image as people/no people are HOG, homogeneity, entropy, and energy.

2. THE PROPOSED APPROACH

Our goal is to classify an image with simple features. For indoor/outdoor classification, we compute features based on sub-blocks of the image instead of using features for the entire image. Some specific objects usually appear in the specific part of the image–the sky usually appears on the top part of the image and grass or carpet appear on the bottom part of the image. If we extract simple features over the entire image, we would not be able to take account where the objects appear in the image. By dividing the image into several different sections and extracting features separately over the divided regions, we can make better use of the localized information as to where in the image specific objects appear. This is especially useful when it comes to indoor/outdoor classification since many objects in indoor/outdoor scenes have “typical” positions. For people/no people classification, the feature extraction is done over the entire image. We do not use sub-blocks since we do not consider where in the image people appear–we only consider whether people appear in the image or not. The details of feature extraction and how we combine them are described in Sections 2.1 and 2.2.

2.1 Image Features

For indoor/outdoor classification, we use four types of features: color, edge, lines and standard deviation of luminance. The DCD color feature is found by using color descriptors from the MPEG-7 standard [9]. The number of edge pixels and histogram of gradient magnitude are extracted as edge features. The number of vertical lines and the number of horizontal lines are extracted as line features. For people/no people classification, we use two types of features: HOG and texture. The texture feature is computed using GLCM [10].

2.1.1 Color Feature

Dominant Color Descriptor (DCD)
An input RGB color image is divided into sub-block of size 64 × 64 pixels and converted to the CIE LUV color space. The CIE LUV space is intended to model the human vision system [11], [12]. The Euclidian distance between two color components U and V from the CIE LUV space are strongly correlated with the human visual perception and the L component closely matches human perception of lightness [13]. The L, U and V color components can be obtained from the XYZ color space components as follows [12], [14]:

$$L = \begin{cases} 116\sqrt[3]{y_r} - 16; y_r > (\frac{6}{29})^3 \\ \\ (\frac{29}{3})^3 y_r; y_r \leq (\frac{6}{29})^3 \end{cases} \tag{1}$$

$$v = 13L(v' - v'_r) \tag{2}$$

$$u = 13L(u' - u'_r) \tag{3}$$

where,

$$y_r = \frac{Y}{Y_r} \tag{4}$$

$$u' = \frac{4X}{X + 15Y + 3Z} \tag{5}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \tag{6}$$

$$u_r{}' = \frac{4X_r}{X_r + 15Y_r + 3Z_r} \tag{7}$$

$$v_r{}' = \frac{9Y_r}{X_r + 15Y_r + 3Z_r} \tag{8}$$

where X_r,Y_r, Z_r are the CIE XYZ tristimulus values of the reference white point [12], [15]. X, Y, Z values can be obtained from RGB by the following operation [12], [16].

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{9}$$

Then the dominant colors are determined from the CIE LUV color space by cluster center of L, U, and V which are replaced by v_{iL}, v_{iU}, v_{iV} [17].

$$v_{i_L} = \frac{\sum_{k=1}^{N} \mu_{ki_L}^{m} x_{k_L}}{\sum_{k=1}^{N} \mu_{ki_L}^{m}} \tag{10}$$

$$v_{i_U} = \frac{\sum_{k=1}^{N} \mu_{ki_U}^{m} x_{k_U}}{\sum_{k=1}^{N} \mu_{ki_U}^{m}} \tag{11}$$

$$v_{i_V} = \frac{\sum_{k=1}^{N} \mu_{ki_V}^{m} x_{k_V}}{\sum_{k=1}^{N} \mu_{ki_V}^{m}} \tag{12}$$

where x_{kL}, x_{kU}, x_{kV} represent the L, U, and V channel values at pixel k respectively, k = 1, 2, 3, , N. m is the constant value equal to 2. N is number of pixels. μ_{kiL}, μ_{kiU}, μ_{kiV} are the membership function which composes of grade of membership of each data points for L, U, and V channels as shown in the Eqs. (13), (14), (15).

$$\mu_{ki_L} = \frac{1}{\sum_{j_L=1}^{c} \left(\frac{\left\| x_{k_L} - v_{i_L} \right\|}{\left\| x_{k_L} - v_{j_L} \right\|}\right)^{2/m-1}} \tag{13}$$

$$\mu_{ki_U} = \frac{1}{\sum_{j_U=1}^{c} \left(\frac{\left\| x_{k_U} - v_{i_U} \right\|}{\left\| x_{k_U} - v_{j_U} \right\|}\right)^{2/m-1}} \tag{14}$$

$$\mu_{ki_V} = \frac{1}{\sum_{j_V=1}^{c} \left(\frac{\left\| x_{k_V} - v_{i_V} \right\|}{\left\| x_{k_V} - v_{j_V} \right\|}\right)^{2/m-1}} \tag{15}$$

where c is the number of clusters, i = 1, 2, 3,...,c. Then the feature vectors of DCD of each cluster (F_{DCD_i}) are shown in the Equations (16) and the feature vectors of DCD of all clusters (F_{DCD}) are shown in the Equations (17).

$$F_{DCD_i} = \begin{bmatrix} v_{i_L} & v_{i_U} & v_{i_V} \end{bmatrix} \tag{16}$$

$$F_{DCD} = \begin{bmatrix} F_{DCD_1} & F_{DCD_2} & F_{DCD_3} & ... & F_{DCD_c} \end{bmatrix} \tag{17}$$

2.1.2 Edge Intensity

A. Number of Edge pixels (n_e)

Let n_e be the number of edge pixels from the entire image. The input color image is converted from RGB to grayscale. We use the Canny edge detector [18] on the grayscale image to obtain the edge mask.The edge mask is divided into sub-blocks of size 64×64 pixels then n_e is obtained by counting the number of edge pixels in the each sub-block.

B. Gradient Magnitude (D_G)

The RGB input color image is converted to YCbCr then the Y channel is extracted. This Y channel is used to calculated the gradient magnitude (G) using Eqs. (18)

$$G = \sqrt{G_x^2 + G_y^2} \tag{18}$$

where G_x, and G_y are the images, where each pixel represents horizontal and vertical derivative approximation respectively.Then G is divided into sub-blocks of size 64×64 pixels to obtain D_G.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * Y \tag{19}$$

$$G_y = \begin{bmatrix} +1 & 0 & +1 \\ 0 & 0 & 0 \\ -1 & 0 & -1 \end{bmatrix} * Y \tag{20}$$

2.1.3 Lines

We extract two line features: a number of horizontal lines and a number of vertical lines known as n_h, and n_v. The input color image is first converted to grayscale and processed using the Canny edge detector to obtain the edge mask [18]. Note that the minimum and maximum threshold for Canny edge detection are 50 and 180. This edge mask is divided into sub-blocks which size are 64×64 pixels then the sub-blocks of edge mask are applied with the Hough line detector [19]. The family of lines go through point (x_0,y_0) can be written as shown in Equation (21) [20]

$$r_\Theta = x_0 \cdot cos\Theta + y_0 \cdot sin\Theta \tag{21}$$

where represents the algebraic distance between the line and the origin, while represents the angle between the orthogonal vector to the line and the positive x-axis. The threshold of Hough line referred as the minimum number of intersections to detect as the line, is set to 46. And the maximum gaps between two points to be considered in the same line is set to 5. The angle of the vertical line is between 85 and 95 degree, the angle of the horizontal line is considered from 175 to 185 degree. So the n_h, and n_v are calculated by counting the number of vertical and horizontal lines.

2.1.4 Standard Deviation of Sub-Block Y

In this part, the Y channel is divided into sub-blocks of 64×64 pixels and then the standard deviation of each sub-block of Y (σ_Y) is obtained as shown in Equation (22)

$$\sigma_Y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{22}$$

where n is number of pixels per subblock, x_i is value of Y at pixel i^{th}, and $\bar{x}$ is average value of Y of each sub-block as Eqs. (23)

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{23}$$

2.1.5 HOG

The HOG [21] is implemented based on evaluating normalized local histograms of image gradient orientation in a dense grid. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge position. In the implementation, the RGB input color image with gamma equalization is normalized. Each normalized channel of input image are calculated to obtain the gradients. Each channel is divided into the windows which size is 16×16 then each window is used to calculate the weighted vote for edge orientation histogram channel based on the orientation of gradient element centered on it. The votes are accumulated into the orientation bins known as cells which size is 8×8. Each cell is used to obtain the histogram of gradients using Gaussian smoothing filter over the pixels of the cell. The block is normalized then the output is the HOG feature (F_{HOG}).Note that the parameters used in the implementation are the standard deviation of Gaussian smoothing, σ=0, 1-D [-1,0,1] gradient filter, $L2 - Hys$ (Lowe-style clipped L2norm) block normalization, block spacing stride of 8 pixels (4-fold coverage of each cell), 64×128 detection window, and the orientation bins of gradient voting 0-180 degree [21].

2.1.6 Texture

The gray level co-occurence matrix (GLCM) [10] is used to obtain the texture feature [22]. This method is based on estimation of the second-order joint conditional probability density function $(f(i,j)|d,\theta)$ [10]. Each $(f(i,j)|d,\theta)$ is the probability of the pair of gray levels (i,j) occurring in a pair of pixels of the image which are separated by a distance d along the direction θ. So the estimated values form a two-dimensional histogram which can be written in the matrix form [10], [23]. In the implementation, the entire RGB color image is quantized from 256 levels to 8 gray levels and then it is used to obtain the GLCM by estimating the joint histogram of neighboring pixel values $p(i,j|\theta)$ for angles θ equal to 0 degree. This GLCM is used to obtain three the properties of texture: homogeneity, energy, and entropy are computed as shown in Equations (24), (25), (26) [10]

$$F_H = \sum_i \sum_j \frac{p(i,j)}{1+|i-j|} \tag{24}$$

$$F_T = \sum_i \sum_j p(i,j)^2 \tag{25}$$

$$F_N = -\sum_i \sum_j (p(i,j)log(p(i,j))) \tag{26}$$

where i,j is the position of pixel in the GLCM.

2.2 Features Combination and Weighting

The dimension of each feature is shown in Table 1 and the features are combined using a method known as "feature combination or feature weighting." [24]. This is based using the significance of each feature and how it contributes to the classification. If all the other conditions are the same, features that are closer together within each class are more discriminable for classification compared to the features that are more spread apart within each class. In this paper, we consider features with smaller standard deviation within each class to be the better features that we put more weight to those features.This approach is related to the method described in [24] where feature weights are based on the feature distribution denseness.

To obtain the weight of each feature for indoor/outdoor and people/no people, the standard deviation of each feature is estimated from the training data [24] as shown in Equations (27) and (28). The average weight of 2 classes ($\bar{w}_f$) for each feature is calculated using Equation (29).

$$w_{C1_f} = \frac{1}{\sigma_{C1_f}} \tag{27}$$

Table 1. Dimension of The Features

Feature (F)	Dimension
F_{DCD}	1200
n_e	80
D_G	80
n_h	80
n_v	80
σ_Y	80
F_{HOG}	167796
F_H	1
F_T	1
F_N	1

$$w_{C2_f} = \frac{1}{\sigma_{C2_f}} \tag{28}$$

$$\bar{w_f} = \frac{w_{C1_f} + w_{C2_f}}{2} \tag{29}$$

where w_{C1_f}, w_{C2_f} are the weight values of each feature of class 1 (e.g. indoor, no people), and class 2 (e.g. outdoor, people) respectively. f is the feature, σ_{C1_f} and σ_{C2_f} are the standard deviations of each feature for class 1 and class 2 respectively which can be obtained from Equations (30), and (31).

$$\sigma_{C1_f} = \sqrt{\frac{\sum_{i=1}^{N_{C1}}(x_{if} - \bar{x_f})^2}{N_{C1} - 1}} \tag{30}$$

$$\sigma_{C2_f} = \sqrt{\frac{\sum_{i=1}^{N_{C2}}(x_{if} - \bar{x_f})^2}{N_{C2} - 1}} \tag{31}$$

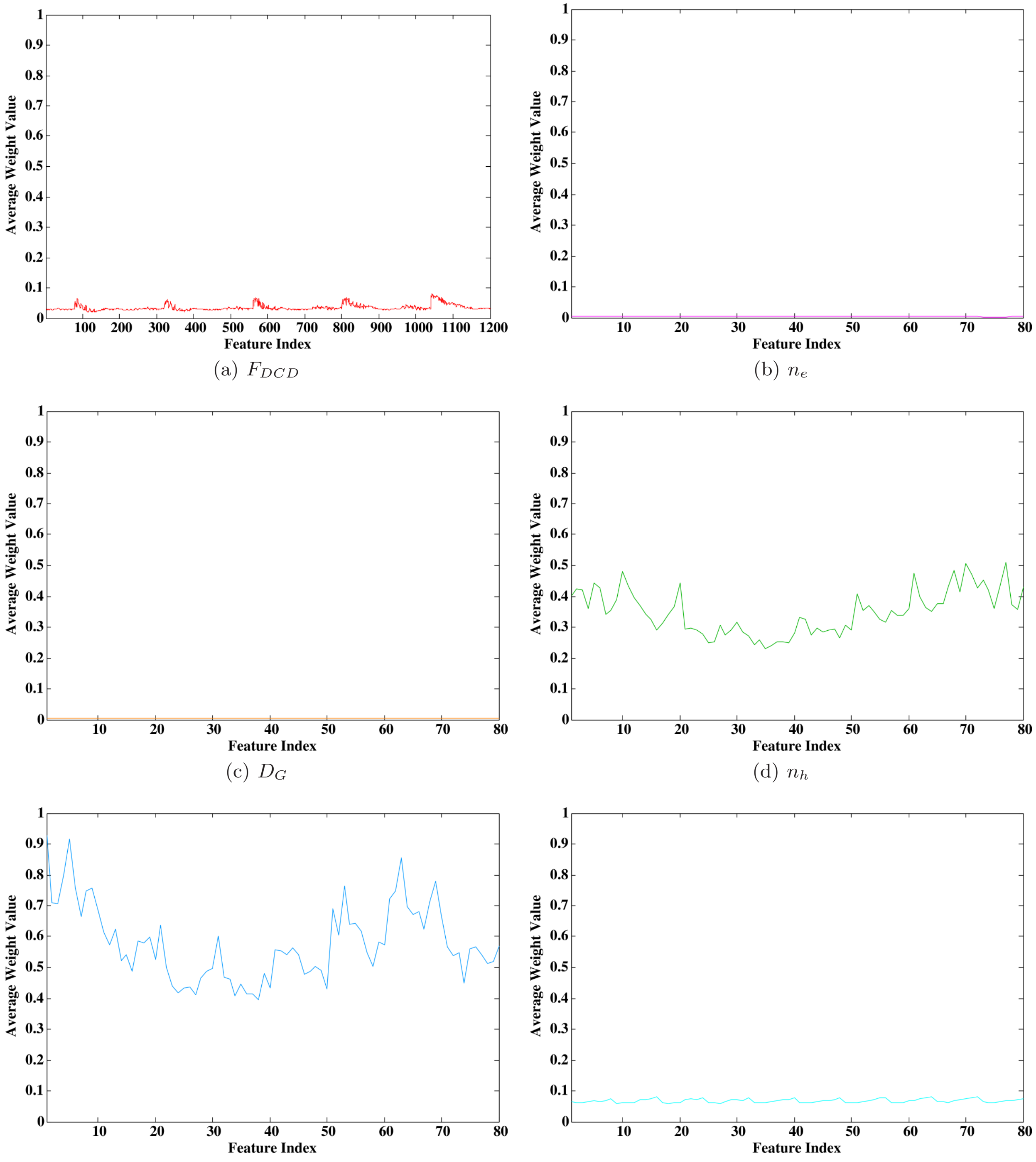
where x_{if} is feature value of f, i is i^{th} image, N_{C1} and N_{C2} is number of class 1 and class 2 images. These features for each class are combined using Eqs. (32)

$$\tilde{X} = \left[\bar{w}_f F\right] \tag{32}$$

where $\tilde{X}$is a matrix of feature vectors after multiplying with the feature weight for each class.

3. EXPERIMENTAL RESULTS

In our experiments, we used the MIT Scene Understanding (SUN) database. For classifying images as indoor/outdoor, 400 images are used for training (200 indoor images and 200 outdoor images) and 200 images are used for testing (100 indoor images, and 100 outdoor images). For classifying images as people/no people, 400 images are used for training (347 no people and 53 people), and 200 images are used for testing (157 no people, and 43 people). The SUN database contains various sizes of images and we selected two image resolutions for our experiments: 640×480 and 1280×960. The 1280×960 images are downsampled to 640×480 so that all the input images to our system are set to 640×480. We feel this is more representative of the web cam images we seen in our work. Table 2 shows the categories of the images used in our experiments. For the indoor/outdoor, the weights of F_{DCD}, n_e, D_G, n_h, n_v, σ_Y are shown in the Figure 1. The dimensions of each feature are 80 except F_{DCD}. From the results, the range of weight of each feature can be any values based on the distribution of feature value of training data. So n_h and n_v get more weight than F_{DCD}, n_e and D_G because the feature values of them are closed together therefore n_h and n_v are more important features used in the classification. For people/no people, the dimension of these features are 1 so the weight of each feature is only 1 value. The lists of the weights of F_H, F_T, F_N are 16.77, 0.00007, 19.87. While the dimension of F_{HOG} is larger, the number of weight values are also larger and they are shown in Figure2.



(a) F_{DCD}

(b) n_e

(c) D_G

(d) n_h

(e) n_v

(f) σ_Y

Figure 1. Average Weight Values for Features Used in Indoor/Outdoor Classification

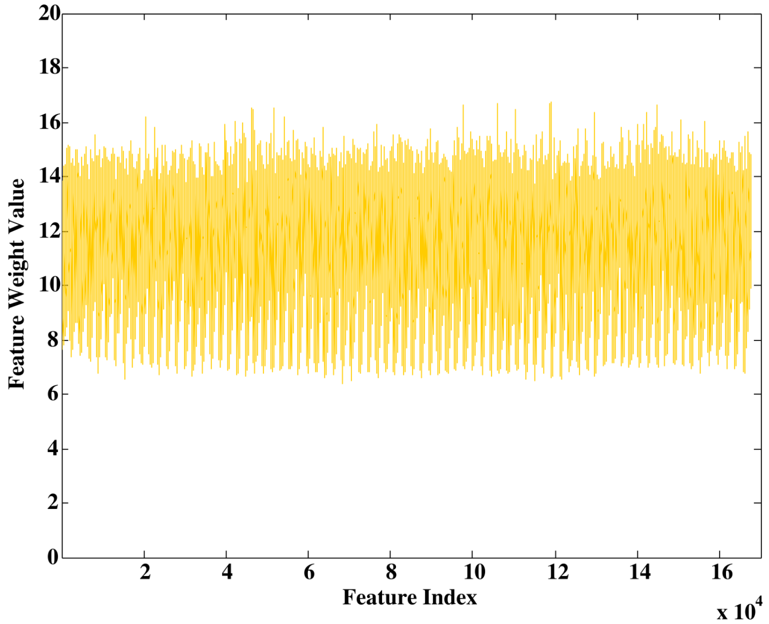


Figure 2. Feature Used in People/No People Classification

Table 2. Categories and Number of Training and Testing Images

Categories	Number of Testing Images	Number of Training Images
Bathroom	10	30
Bedroom	20	30
Bowling	20	30
Conference room	10	30
Hospital room	10	20
Kitchen	10	30
Living room	20	30
Badlands	10	25
Highway	13	30
Pasture	10	29
Mountain	13	18
Vegetable garden	10	17
Desert/vegetation	4	4
Beach	7	7
Botanical Garden	10	17
Baseball Field	10	18
Playground	12	39
Total	200	400

The features of images are classified using a support vector machine (SVM) [25]. The SVM is described by Equation (33).

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{33}$$

subject to

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i; \xi_i \geq 0 \tag{34}$$

where the training set of instance-label pair (x_i, y_i) $i = 1, ..l; x \epsilon R^n; y \epsilon (1, -1)^l$ and the kernel function is the

Radial Basis Function (RBF) [26] and where the kernel parameters are C, and γ

$$K(x_i, x_j) = exp(-\gamma \left\| x_i, x_j \right\|^2), \gamma > 0 \tag{35}$$

We used the LIBSVM library [25] for our experiments. The parameters C and γ are determined empirically by exponentially growing. We found that the parameters which gave the highest accuracy for indoor/outdoor are $C = 2^{-2}$,and 2^{-1} and $\gamma = 2^{-24}, 2^{-23}, ..., 2^{-11}$ while for people/no people $C = 2^{-2}$,and 2^{-1} and $\gamma = 2^{-39}, 2^{-38}, ..., 2^{-1}$. We report the performance of our method in terms of accuracy and precision/recall. Accuracy, recall and precision are obtained as shown in Equation (36), (37) and (38) respectively.

$$\text{Accuracy} = \frac{N_c}{N_t} \times 100\% \tag{36}$$

where N_c and N_t are number of correct images from the classification and number of images in the testing set.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \times 100\% \tag{37}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \times 100\% \tag{38}$$

where TP, FP, and FN represent true positive, false positive, false negative respectively [27]. For the indoor/outdoor classification, the accuracy of each feature is shown in Table 3. For people/no people classification, the accuracy of each feature is shown in Table 4.

Table 3. Accuracy of Each Feature for Indoor/Outdoor Classification

Feature	Accuracy (%)	Recall (%)	Precision (%)
F_{DCD}	87	Indoor: 89, Outdoor: 85	Indoor: 85.57, Outdoor: 88.54
n_e	88.5	Indoor: 90, Outdoor: 87	Indoor: 87.37, Outdoor: 89.69
D_G	84.5	Indoor: 92, Outdoor: 77	Indoor:80, Outdoor: 90.58
n_h	74	Indoor: 87, Outdoor: 61	Indoor: 69.04, Outdoor: 82.43
n_v	74	Indoor: 78, Outdoor: 70	Indoor: 72.22, Outdoor: 76.08
σ_Y	74	Indoor: 96, Outdoor: 92	Indoor: 92.3, Outdoor: 95.83

Table 4. Accuracy of Each Feature for People/No People Classification

Feature	Accuracy (%)	Recall (%)	Precision (%)
F_{HOG}	79	People: 100, No People: 85	People: 85.57, No People: 88.54
F_H	78	People: 100, No People: 0	People: 78.5, No People: ∞
F_N	78	People: 100, No People: 0	People: 78.5, No People: ∞
F_T	78	People: 100, No People: 0	People: 78.5, No People: ∞

From Table 3, we observe that the color feature and the edge intensity features outperform line and standard deviation features. This is because most of the outdoor images are composed of a large blue-colored sky region and/or green meadow regions, that is not present in indoor images. This makes color as the dominant feature for indoor/outdoor classification. Most indoor images tend to have more horizontal and vertical lines than the outdoor images because they compose of some objects such as chair, bed, and sofa.

We now compare the classification accuracies with and without the feature weighting as shown in Table 5. The accuracy which is using the feature weighting is 95.5% while not using feature weighting is 94%. This means that the feature weighting is important to increase the accuracy of image classification.

Table 5. Accuracy Comparison for Different Features Combination Methods for Indoor/Outdoor Classification

Processes	Accuracy (%)	Recall (%)	Precision (%)
With Feature Weighting	95.5	Indoor: 94, Outdoor: 97	Indoor: 96.9, Outdoor: 94.17
Without Feature Weighting	94	Indoor: 96, Outdoor: 92	Indoor: 92.3, Outdoor:95.83

Table 6. Accuracy Comparison for Different Feature Combination Methods for People/No People

Processes	Accuracy (%)	Recall (%)	Precision (%)
With Feature Weighting	78.5	Indoor: ∞, Outdoor: ∞	Indoor: ∞, Outdoor:∞
Without Feature Weighting	78.5	Indoor: ∞, Outdoor: ∞	Indoor: ∞, Outdoor:∞

4. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method to classify images as indoor or outdoor and people or no people scenes using a set of simple visual features. Four types of image features: color, edge, lines, and standard deviation are considered for indoor/outdoor classification while people/no people the HOG and texture features which are homogeneity, entropy, and energy are used. The accuracy of image classification which the features are combined using the feature combining with the feature weighting is 95.5% while without the feature weighting is 94% for indoor/outdoor scene classification. For people/no people classification, the accuracy which is not using feature weighting is 78.5%. In future, we plan to extend our method to classify an image as crowd or no-crowd and to implement this work on our CAM^2 system and test it using thousands of cameras..

REFERENCES

[1] "ITU internet reports 2005: the internet of things," *International Telecommunication Union (ITU) Technical Report*, November 2005.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Septempber 2013.

[3] B. Guo, D. Zhang, and Z. Wang, "Living with Internet of Things: the emergence of embedded intelligence," *Proceedings of the IEEE Internet Conferences on Internet of Things and Cyber, Physical and Social Computing*, pp. 297–304, October 2011, Dalian, China.

[4] J. Choe, T. Pramoun, T. Amornraksa, Y. Lu, and E. J. Delp, "Image-based geographical location estimation using web cameras," *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 73–76, April 2014, San Diego, CA.

[5] M. Szummer and R. W. Picard, "Indoor-outdoor image classification," *Proceedings of the IEEE International Workshop on Content-Based Access of Image and Video Databases*, pp. 42–51, January 1998, Bombay, India.

[6] F. Cakir, U. Gudukbay, and O. Ulusoy, "Nearest-neighbor based metric function for indoor scene recognition," *Computer Vision and Image Understanding*, vol. 115, no. 11, pp. 1483–1492, November 2011.

[7] K. Shimazaki and T. Nagao, "Scene classification using color and structure-based features," *Proceedings of the IEEE 6th International Workshop on Computational Intelligence and Applications*, pp. 211–216, July 2013, Hiroshima, Japan.

[8] R. Raja, S. M. M. Roomi, D. Dharmalakshmi, and S. Rohini, “Classification of indoor/outdoor scene,” *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–4, December 2013, Enathi, India.

[9] L. Cieplinski, “MPEG-7 color descriptors and their applications,” *Computer Analysis of Images and Patterns*, vol. 2124, pp. 11–20, Augeest 2001.

[10] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, November 1973.

[11] B. Wandell, *Foundations of Vision.* Sinauer Associates, Inc., 1995, sunderland, MA.

[12] G. Sharma, *Digital Color Imaging Handbook.* CRC Press, 2002, Boca Raton, FL.

[13] Y. He, “Context based image analysis with application in dietary assessment and evaluation,” *Ph.D. Dissertation, School of Electrical and Computer Engineering, Purdue University*, pp. 23–28, May 2014.

[14] J. Schanda, *Colorimetry: understanding the CIE system.* John Wiley & Sons, 2007, Hoboken, NJ.

[15] G. Kennel, *Color and mastering for digital cinema.* Focal Press, 2006, Burlington, MA.

[16] H. Fairman, M. Brill, and H. Hemmendinger, “How the CIE 1931 color-matching functions were derived from WrightGuild data,” *Color Research and Application*, vol. 22, no. 1, pp. 11–22, February 1997.

[17] S. L. Chiu, “Fuzzy model identification based on cluster estimation,” *Journal of Intelligent and Fuzzy Systems*, vol. 2, no. 3, pp. 267–278, 1994.

[18] J. Canny, “Computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, November 1986.

[19] J. Matasa, C. Galambos, and J. Kittler, “Robust detection of lines using the progressive probabilistic hough transform,” *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 679–698, April 2000.

[20] R. O. Duda and P. E. Hart, “Use of the hough transformation to detect lines and curves in pictures,” *Magazine Communications of the ACM*, vol. 15, no. 1, pp. 11–15, January 1972.

[21] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, June 2005.

[22] A. B. Chan, M. Morrow, and N. Vasconcelos, “Analysis of crowded scenes using holistic properties,” *Proceedings of the IEEE international Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 1–8, June 2009.

[23] A. N. Marana, L. F. Costa, R. A. Lotufo, and S. A. Velastin, “On the efficacy of texture analysis for crowd monitoring,” *Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision (SIBGRAPI)*, pp. 354–361, October 1998.

[24] K. Wang, X. Wang, and Y. Zhong, “A weighted feature support vector machines method for semantic image classification,” *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation*, vol. 1, pp. 377–380, March 2010.

[25] C. Hsu, C. Chang, and C. Lin, “A practical guide to support vector classification,” *Technical Report, Department of Computer Science, National Taiwan University*, April 2010.

[26] S. Keerthi and C. Lin, “Asymptotic behaviors of support vector machines with Gaussian kernel,” *Neural Computation*, vol. 15, no. 7, pp. 1667–1689, July 2003.

[27] D. L. Olson and D. Delen, *Advanced data mining techniques.* Springer Berlin Heidelberg, 2008.