

# OBJECT COLOR CATEGORIZATION IN SURVEILLANCE VIDEOS

Yimeng Zhang<sup>1</sup> ChengChuan Chou<sup>2</sup> Shiaw-Shian Yu<sup>2</sup> Tsuhan Chen<sup>1</sup>

<sup>1</sup>Cornell University <sup>2</sup>Industrial Technology Research Institute

## ABSTRACT

We address the problem of automatic color categorization of the objects in surveillance videos. This problem is challenging for realistic situations due to the large intra-class variations of the same color and the large portions of noisy areas including the backgrounds and the parts of the objects that do not contribute to color assignments. We develop an integrated color categorization system with algorithms that address these challenges. With the algorithms proposed in this paper, we can improve the average color categorization accuracy by 18% from our previous work [7].

*Index Terms*— Color Categorization, SVM, Kernel methods, Object detection

## 1. INTRODUCTION

We explore the problem of automatic color assignment to the objects in surveillance videos. Given an image frame including the object of interest, the system will classify the color of the object into one of the predefined color categories.

While human can effortlessly tell the color of an object in an image, the mapping between the RGB values of an image and the color categories is challenging for the computers due to the following reasons: 1) The large intra-class variations: the same color appears quite different in different environments, under different lighting conditions, and with different camera settings. 2) Interference from irrelevant parts: Not every part of an object contributes to its color assignment. For example, a red car may have gray or black windows, silver wheels, and black tires. 3) Imperfect preprocessing: to fully automate the process, there are some preprocessing steps, such as background subtraction and object detection. The results of these preprocessing steps are usually far from perfect. Therefore, the color categorization must be performed with the noisy areas resulted from these steps. In this paper, we target to design a system that can overcome these challenges.

For color categorization, one type of previous works is the chip based method [1][3]. The mapping from the RGB values to the color names is derived from the labeled set of color chips. Another type of methods is the learning based methods [2][5][7]. Statistical models that classify the object image into color categories are learned with real world

images. The learning based methods are more flexible to environmental changes of the colors compared to the chip based methods. Weijer et al. [5] learn the distribution of RGB values for each color category based on a PLSA (probabilistic Latent Semantic Analysis) based model. The RGB distribution of an image is modeled as a combination of the foreground distribution determined by the color label and the background distribution shared among all images. Our previous work [7] proposes a Co-PLSA model, which uses shape information to help filtering out the background and the parts of the objects that do not contribute to the color assignments, and thus improve the color categorization performance. Both the models proposed by Weijer et al. [5] and us [7] are based on the PLSA model, which is a generative classifier. In general, a discriminative classifier, especially the maximum margin classifier (SVM), produces better classification performance than a generative classifier. In this paper, we propose a SVM based color classifier, and generalize the shape model in [7] so that we can still use shape to improve the proposed classifier.

We summarize our contributions of this paper as follows. 1) We develop an integrated color classification system for objects in surveillance videos with components including background subtraction, weather classification and color classification. 2) We propose a two-step SVM for color classification and design kernels that are suitable for measuring the color histograms. 3) We generalize the shape model in [7] so that we can integrate shape information into the proposed color classifier to filter out the backgrounds and the parts of objects irrelevant for color assignment.

## 2. APPROACH

We define our problem as follows: given an image frame, we want to classify the object of interest (for example, a car) in the image into one of the seven categories: black, gray, white, green, red, yellow, and blue. These categories are widely used color names in previous works [2][7].

### 2.1. System Overview

Figure 1 illustrates the overview of our system. There are three main components in the system: 1) Background subtraction. Given an image frame, an initial background subtraction based on the pixel changes among the frames is conducted. We use Gaussian Mixture Model to model the distribution of the pixel-wise RGB values. Next, we use the shape model to further filter out the backgrounds that the

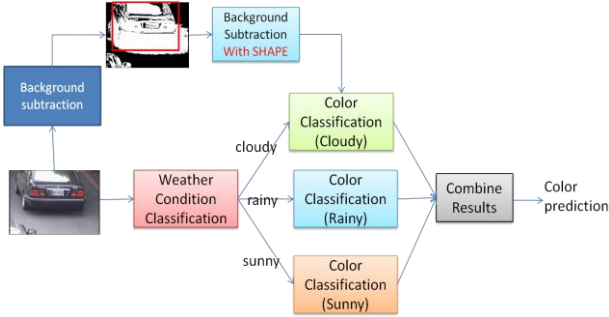


Figure 1: System Overview

pixel based method failed to remove, and the parts of the objects that do not contribute to color assignment. 2) Weather classification. This step gives the system the ability to overcome the main cause of color changes in surveillance videos. 3) Color classification. During training, we train different color classifiers for different weather conditions. Given a testing image, first the weather classifier is applied, and then the color classification results of all weather conditions are combined based on the weather classification result.

## 2.2. Weather Classification

Given an image, we want to obtain the probability of each weather condition. To construct the feature vector of an image, we use the H,S,V histograms and the histograms of Green and Blue channels. Specifically, the HSV histogram for an image is calculated by first quantizing the HSV space into  $N_h \times N_s \times N_v$  number of bins, and then counting the number of pixels in each bin. To construct the Green and Blue histograms, we first normalize the R, G, B values of each pixel with its intensity, and then calculate the histograms of G and B channels respectively. The feature vector is finally created by cascading the histograms of HSV, Green, and Blue channels. We use the probabilistic SVM with RBF kernels to classify the feature vectors of images.

## 2.3. Color Classification

We perform color classification on the foreground regions predicted by the background subtraction component. We train different color classifiers under different weather conditions. During training, the weather conditions are known. Therefore, the classifier is trained with the images labeled with the weather of interest. During testing, an image is tested on the classifiers of all three weather conditions. We combine the results of these classifiers by making a weighted linear summation of the output probabilities of the color categories. The weights are defined with the weather probabilities predicted by the weather classifier.

We use HSV histograms to represent the resulted regions after background subtraction. The histograms are calculated in the same way as Section 2.2.

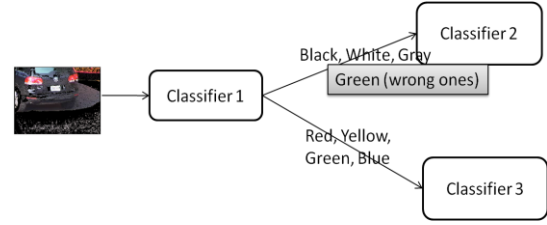


Figure 2: Illustration of the color classifier

### 2.3.1. Two-step color classifier

We propose a two step color classifier which is illustrated in Figure 2. The first step classifies the colorless (black, gray, white) and colorful (blue, red, green, yellow) categories (classifier 1 in Figure 2). The second step further classifies the regions into the seven categories. The intuition is that the classifier for colorful categories and the one for colorless categories should give different weights to the features. For example, when classifying the colorful categories, the H channel is more important. On the contrary, to classify the colorless categories, we should focus more on the S and V channels. In addition, the classifier for color vs. colorless categories will also be different. Therefore, by separating the tasks, we can make each task easier and thus improve the classification accuracy. We use SVM as the classifiers for each step.

We found that some green objects, such as dark green cars, are confused with the black ones. Therefore, when training the classifier 2 (in Figure 2), we use the training images labeled with black, gray or white and the images that are predicted to colorless but labeled as green. Thus, at step 2, we can further correct the green colors incorrectly classified at step 1. Our two step classifier is similar to a decision tree with SVM as the basic classifier, while we pre-define the attributes of each node in the tree with our knowledge about the colors.

### 2.3.2. SVM kernels

We design different kernels to further improve the SVM with the standard kernels (linear, polynomial, Gaussian). The Gaussian (RBF) kernel of two histograms H and K is defined as follows:

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

The kernel is essentially the exponential of the Euclidian distance of two histograms. Many other distance measures have been proposed for measuring the dissimilarity of histograms. In particular, the Earth Mover's distance [6] and Jeffrey divergence [4] have been shown to be more suitable for color histograms. By changing the distance measure in the equation (1) to a more suitable measure, we can expect to improve the color categorization results. Let  $d(H, K)$  denote the dissimilarity defined by the measure  $d$ , the kernel will be calculated as:

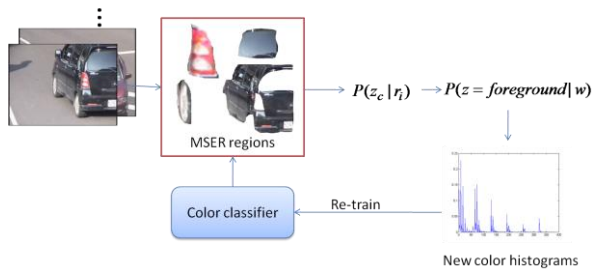


Figure 3: Illustration of the shape model

$$K(x_i, x_j) = \exp\left(-\frac{d(H, K)}{\gamma}\right) \quad (2)$$

Only the kernels that satisfy the Mercer’s condition guarantee a global optimal solution to the SVM. Not all distance metric leads to a Mercer’s kernel. However, in practice, a non-Mercer’s kernel usually reports promising empirical results [8]. This is also seen in our experiments

#### 2.4. Shape Model

In surveillance videos, the pixel based background subtraction results are usually far from perfect. Moreover, the object itself includes many areas that are irrelevant for color assignments. In [7], we proposed a PLSA based model that uses shapes to help filtering out these noisy areas. We generalize the model so that the SVM based classifier we proposed in Section 2.3 can be applied.

The main idea is that the foreground areas (areas that contribute to color assignments) should be consistent in shapes for a certain object category. During training, the color detector helps to learn the shape detector for separating foreground areas from other areas. In turn, the shape detector helps to improve the color detector by focusing the learning of colors on the foreground areas.

To represent the shapes, we extract SIFT features on Maximally Stable Extremal Regions (MSER). The SIFT features are further clustered with k-means to create the code words.

The algorithm is illustrated in Figure 3.

1) For each MSER region  $R$  in a training image, we extract its HSV histogram, apply the color classifier and obtain the probabilities of the color categories for the region.

2) Since we know the color label of the objects, we can calculate the probability of foreground of a region  $R$ ,  $P(z = \text{foreground} | R)$  as the probability of the labeled color, and the probability of background as the summation of the probabilities of all other colors.

3) Each region has a code word assignment. We perform maximum likelihood estimation for the probability of foreground of each word  $P(\text{foreground} | w)$ .

4) With this probability, we can further calculate the probability of foreground for each pixel on the image by averaging the foreground probability of the regions covering that pixel. Note here we use the probability of the word assigned to that region rather than the probability we calculated at step 2. Thus, we obtain a probabilistic back-

ground subtraction result. We combine this result with the pixel based background subtraction result by making a multiplication. New HSV histograms are computed with the new foreground estimation. Since the foreground prediction results are probabilistic, we use the probabilities of foreground pixels when calculating the color histograms.

5) With the new color histograms, we retrain the color classifier.

These steps can be iterated as needed. We iterate 3 times in our experiments. In the beginning, the color classifier is initialized with the classifier learned on the regions without applying shape refinement.

### 3. EXPERIMENTS

To evaluate our color categorization system, we collected image from the frames of surveillance videos taken with 9 cameras under different weather conditions: sunny, cloudy, and rainy. There are 2 or 3 videos in different scenes for each camera and each weather condition. We use the videos from 7 of the 9 cameras for training, and videos from the other 2 cameras for testing to ensure that our system can perform color categorization for unknown cameras. For the objects, we focus on vehicles in the videos. The vehicle color labels of each frame are provided. Table 1 summarizes the number of image frames we used for training and testing.

Table 1. The number of frames used for training and testing.

	Total	Training	Testing
Sunny	5409	3968	1441
Cloudy	6127	5030	1097
Rainy	3920	3013	907
Total	15456	12011	3445

In the experiments, we would like to verify: 1) by changing the color classifier from PLSA to the proposed two-step SVM, we can improve the classification accuracy; 2) the proposed SVM kernels outperform the standard SVM kernels; 3) performing weather classification increases the robustness of the system towards color changes caused by weather.

#### 3.1. Color Classifier

To compute the color histograms, we quantize the HSV space into  $8 \times 8 \times 8$  bins.

For evaluating the proposed two-step color classifier, we perform experiments on the images of sunny weather. Table 2 shows the classification accuracy of each color category. The accuracy of a color category is calculated as the percentage of correctly classified images of that category. For the one step color classifiers, SVM performs similar to PLSA. Using the two-step classifier, the average accuracy increased around 7%. By changing the standard RBF kernel to the kernels based on the earth mover’s distance and Jeffrey divergence, we can further improve the performance. Jeffrey divergence provides the best results, which achieves

82.9% average accuracy. For the following experiments, we assume that Jeffrey kernel is used.

Table 2. Accuracies of different color classifiers

Method	Black	Blue	Gray	Green	Red	White	Yellow	Avg.
PLSA [7]	40.2	73.2	59.2	42.3	97.2	39.4	98.5	64.3
SVM (RBF)	20.3	84.2	71.6	42.3	97.2	38.7	98.5	64.7
two-step RBF	38.7	82.5	62.0	69.2	95.8	52.1	<b>100</b>	71.5
SVM EMD	65.7	86.8	<b>76.6</b>	59.3	97.3	47.1	98.6	75.9
Jeffrey	<b>71.1</b>	<b>88.0</b>	65.9	<b>84.6</b>	<b>97.3</b>	<b>73.2</b>	<b>100</b>	<b>82.9</b>

### 3.2. Integrated performance

We show the overall results on images of all weather conditions. Table 3 shows the accuracy of each category with and without weather classification. Without weather classification, we make no difference of the weather, and train one color classifier on training images of all weathers. According to the table, performing weather classification first can increase the average accuracy by 6%.

Table 4 presents the confusion matrix of the integrated system with weather classification. The colorful categories (blue, green, red and yellow) are relatively easy to be classified, while the black, gray and white categories are more confusing for our system. This match human’s perspective. Even for people, sometimes it would be difficult to tell apart the gray, black, white vehicles by just looking at the images.

Table 3. Accuracies with and without weather classification

Method	bla.	blu.	gra.	gre.	red	whi.	yell.	avg.
without	62.7	87.1	<b>51.6</b>	56.5	93.2	66.8	100	74.0
with	<b>74.6</b>	<b>94.3</b>	46.8	<b>75</b>	<b>94.1</b>	<b>75.4</b>	100	<b>80.0</b>

Table 4. Confusion matrix

	Black	Blue	Gray	Green	Red	White	Yell.
Black	<b>74.6</b>	4.6	12.0	4.4	0	4.4	0
Blue	2.4	<b>94.3</b>	1.0	1.3	0.7	0.3	0
Gray	20.7	7.7	<b>46.8</b>	4.3	0.3	19.9	0.2
Green	8.3	9.5	1.2	<b>75.0</b>	0	4.8	1.2
Red	1.1	0.5	2.6	0.5	<b>94.2</b>	1.1	0
White	13.4	0.7	9.6	0.9	0.0	<b>75.4</b>	0
Yellow	0	0	0	0	0	0	<b>100</b>

### 3.3. Result visualization

Figure 4 shows example background subtraction results with pixel based method and shape based method. The pixel method is based on the pixel-wise RGB value changes of neighboring frames. The output of the shape model proposed (Section 2.4) is a probabilistic map of the foreground. In order to show the results, we set a threshold to determine the foreground. As shown in the figure, the shape model can help to remove more backgrounds and irrelevant parts of the vehicles, such as windows and tires. While some relevant parts are also removed by the shape model (the third example in the figure), the remaining parts are easier for color classification.

Figure 5 presents some typical errors made by our system. The errors are mainly resulted by the poor background subtraction results (the last image), or the color categories that are confusing for our classifier, such as gray vs. black, black vs. green.

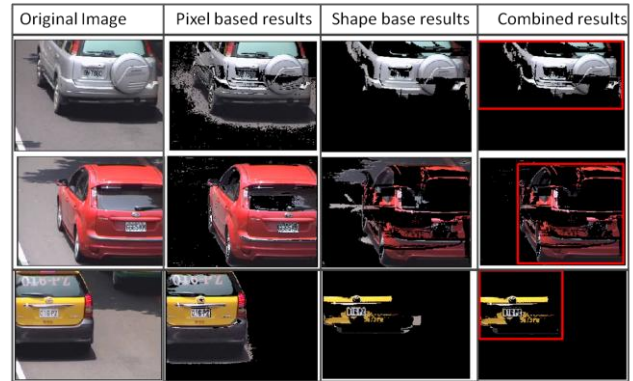


Figure 4: Background subtraction results. The red box is drawn based on the background subtraction results in order to show the locations of the vehicles.



Figure 5: Errors made by our system. The box shows the predicted locations of the vehicles. In the bottom of each image, we show the predicted color (in red letters) and the ground truth (in black)

## 4. CONCLUSION

We developed an integrated color categorization system which is robust enough to be applied to surveillance videos. In addition, we proposed a two-step SVM based classifier with kernels designed to suit the color categorization task. Finally, we generalized our previous PLSA based shape model so that we can integrate shape information to the newly proposed color classifier. The experiment results showed that we can improve the average accuracy from 62.2%, which is the performance of our previous work [7], to 80.0%.

## 5. REFERENCES

- [1] A. Mojsilovic, “A computational model for color naming and describing color composition of images”, IEEE Trans. on Image Processing, 14(5):690-699, 2005.
- [2] K. Kim, S. Park, and Y. Choi, “Deciding the number of color histogram bins for vehicle color recognition”, in APSCC, 2008.
- [3] J.M. Lammens. “A computational model of color perception and color naming”, PHD thesis, University of Buffalo, 1994.
- [4] J. Puzicha, T. Hofmann, and J. M. Buhmann. “Non-parametric similarity measures for unsupervised texture segmentation and image retrieval”, in CVPR, 1997.
- [5] J. van de Weijer, C. Schmid, J. Verbeek “Learning color names from real-world images”, in CVPR, 2007.
- [6] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval”, IJCV, 2000.
- [7] Y. Zhang, S. Yu and T. Chen, “Improving object color categorization with shapes”, in ICIP, 2010.
- [8] C. Ong, S. Canu, and A. J. Smola, “Learning with non-positive Kernels”, in ICML, 2004.