A Convolution Kernel Method for Color Recognition

Jeong-Woo Son, Seong-Bae Park and Ku-Jin Kim
Department of Computer Engineering
Kyungpook National University
702-701 Daegu, Korea
{jwson, sbpark}@sejong.knu.ac.kr, kujinkim@knu.ac.kr

Abstract

Color recognition for out-door images is important for low-level computer vision, but it is a difficult task due to the effect of circumstances such as illumination, weather and so on. In this paper, we propose a novel convolution kernel method to extract color information from out-door images. When two images are compared, the proposed kernel maps images onto a high-dimentional feature space of which features are image fragments of two images and then the similarity between them is obtained through the inner-production of two image vectors. To evaluate the proposed kernel, it is applied to the vehicle color recognition problem. In the experiments on 500 vehicle images, the vehicle color recognition model with the proposed kernel shows about 92% of precision and 92% of recall. On the other hands, the model with a linear kernel shows about 45% of precision and 45% of recall. These experimental results imply that the proposed kernel is a plausible approach for the color recognition task.

1 Introduction

The image color recognition problem over images or parts of images is important for low-level computer vision [15] since colors have much information of an object or an image. By using color information, the performance of other tasks in the field of content-based image retrieval or color indexing [2, 5] can be improved. However, the recognition of a given image's color is not an easy task because colors can be deeply varied by some factors such as illumination, points of view, and so on. For out-door images, the color recognition task is more difficult since additional factors from circumstance when images are generated can affect the apparent color [9]. Consequently, many color recognition methods over out-door images have been studied [2].

In the view point of machine learning, the recognition of image color can be tranformed into a classification problem. The Support Vector Machines (SVMs) [10] has shown to be one of the most effective paradigms for classification. To classfy images into an appropriate color class using SVMs, a similarity measure between images is required. The inner-production between image vectors is generally used to obtain a similarity. Accordingly, images should be mapped onto a vector space. In previous works [2, 4, 9, 14], images are represented as a color histogram, and then, the similarity is computed by inner-production of two color histogram vectors. These approaches use just the statistics of color values thus it gives rise to the loss of information. To handle color values without the information loss, a feature space should reflect not only the color value of each pixel but also the relation among pixels. However, it is a difficult problem to extract these features.

In SVMs, a kernel function measures the similarity among data points by mapping data into a high dimensional space even though we have difficulty to decide features implicitly. Images are complex structured data which consist of a number of pixels. For handling structured data like tree [3], graph [12], string [7] and so on [8], convolution kernels are widely applied In machine learning community [6]. Collins and Duffy [3] proposed the parse tree kernel for processing parse trees. Lodhi et al. [7] applied the string kernel to the text classification task. By adopting convolution kernels, previous works achieved higher performance on their tasks.

In this paper, we propose a novel convolution kernel function, Grid Kernel, and we applied the proposed kernel to the vehicle color recognition problem. In the proposed kernel, an image vector is composed of image fragment features and each feature has information about both a center pixel and its neighbor pixels. The value of each feature is the appearence frequency on an image. By utilizing a dynamic algorithm,

0-7695-2930-5/07 $25.00 © 2007 IEEE
DOI 10.1109/ALPIT.2007.28

these features and values are computed without enumerating them explicitly. For the vehicle color recognition model, first, we convert images into HSV (Hue-Saturation- Value) color space. And then, we generate color recognition models for predefined color class using SVMs with Grid kernel. At this time, Grid kernel compares pixels using H and S value to reduce the effect from circumstance factors.

To evaluate the proposed kernel and the color recognition model, we experimented with 500 out-door vehicle images. In our experiments, we classfy images into 5 classes, blue, red, yellow, black and white. The experimental result shows about a precision of 92% and a recall of 92%. In a comparison experiment, our model outperforms both Linear kernel model and Grid kernel model with RGB (Red-Green-Blue) color space. These results imply that our kernel can obtain a better similarity between color of images by reducing the effect of circumstance factors.

The rest of the paper is organized as follows. Section 2 details Grid kernel and Section 3 presents the vehicle color recognition model. The experimental results are presented in Section 4. Finally, Section 5 draws conclusions.

2 Grid Kernel

Grid kernel computes a similarity between an image pair using implicitly extracted feautres. In Grid kernel, a feature is defined to be an image fragment which appeares on an image. Thus n size image fragment is composed of $n \times n$ pixels. By this definition, an image which has size of $h \times w$ can be represented as

$$Image\ I = \{SI_I^1, SI_I^2, \ldots, SI_I^n\},$$

where $n \geq 0$ and $n \leq min(w,\ h)$. SI_I^k is an image fragment set which has $k \times k$ size image fragments. To generate these image fragment sets, SI_I^k, we need to enumerate all image fragments. Let us define the function $si_I^k(i,j)$ to be image fragments which have a size of $k \times k$ and be located on $(i,\ j)$, so that the $n \times n$ size image fragment on $(i,\ j)$ is represented as

$$\begin{aligned} si_I^n(i,\ j) = \{ & si_l^{n-1}(g,h) | i-1 \leq g \leq i+1, \\ & j-1 \leq h \leq j+1, (g,\ h) \neq (i,\ j)\}. \end{aligned}$$

Note that the function $si_I^k(i,j)$ returns a set which consists of smaller image fragments. After then, V_I, the vector of image I is represented as

$$V_I = \{\#si_I^n(i,j) | 0 \leq n \leq min(w,h), 1 \leq i,\ j \leq w,\ h\}$$

where $\#si_I^n(i,j)$ is the frequency of an image fragment from the function $si_I^n(i,j)$. After defining an image vector, the similarity between an image pair can be obtained by computing the inner product of the vectors, $V_{I_1} \cdot V_{I_2}$. If two images, $I_1,\ I_2$ have size of $h_1 \times w_1$ and $h_2 \times w_2$ respectively, then equation of their inner production is defined as

$$\begin{aligned} < V_{I_1}, V_{I_2} > &= \sum_{s \in S} V_{I_1}^s V_{I_2}^s \\ &= \sum_{j,k,m,o}^{h_1,w_1,h_2,w_2} \sum_{s \in S} Id_{I_1}^s(j,k) Id_{I_2}^s(m,o) \\ &= \sum_{j,k,m,o}^{h_1,w_1,h_2,w_2} K(j,k,m,o,I_1,I_2,n), \end{aligned}$$

where S is a set of image fragments which appeare in both image I_1 and I_2 and has the size of n. $\sum_{j,k,m,o}^{h_1,w_1,h_2,w_2}$ is $\sum_j^{h_1} \sum_k^{w_1} \sum_m^{h_2} \sum_o^{w_2}$. The function $Id_i^s(j,k)$ is an indicator function which returns 1 if the center pixel of image fragment s is same with the j, k pixel of image i, and 0 otherwise. We define the function $K(j,k,m,o,I_1,I_2,n) = \sum_{s \in S} Id_{I_1}^s(j,k) Id_{I_2}^s(m,o)$. Then, we can compute $K(j,k,m,o,I_1,I_2,n)$ in polynomial time, due to the following recursive definition:

1. If $C(j,k,m,o,I_1,I_2)$ is 0,
   - $K(j,k,m,o,I_1,I_2,n) = 0$
2. If $C(j,k,m,o,I_1,I_2)$ is 1 and two pixels are terminals $(n = 0)$,
   - $K(j,k,m,o,I_1,I_2,n) = \lambda$
3. else
   - $K(j,k,m,o,I_1,I_2,n) = \lambda \prod_i^{np(out)} (1 + K(out_i(j,k).height, out_i(j,k).width, out_i(m,o).height, out_i(m,o).width, I_1, I_2, n-1))$

where $out(j,k)$ is the function which returns the outer pixel among the pixels at jth row and kth column, and $np(out)$ returns the size of out. λ is decay factor which makes the farther pixels from the central pixel have less influence. The function $C(j,k,m,o,I_1,I_2)$ is defined to decide two pixels are same or not. Therefore we can define $C(j,k,m,o,I_1,I_2)$ as,

1. If two pixels at jth row and kth column on image I_1 and at mth row, oth column on I_2 are same
   - $C(j,k,m,o,I_1,I_2)$ returns 1

2. else
   - $C(j, k, m, o, I_1, I_2)$ returns 0.

It is clear from $\sum_{j,k} \sum_{m,o} K(j, k, m, o, I_1, I_2, n) = \sum_{s \in S} V_{I_1}^s V_{I_2}^s$, and the recursive rules, that $\sum_{s \in S} V_{I_1}^s V_{I_2}^s$ can be calculated in $O((h_1 \times w_1)(h_2 \times w_2))$. Now we note that the time complexity can be reduce by considering following assumtions:

1. Two images have same size.
2. A feature located on specific position of the image I_1 is only equivalent to same feature on same position of the image I_2.

By those assumtions, the time complexity can be reduced to $O(h \times w)$. Moreover, Grid kernel can utilize not only the information of image's color, but also the information of pixel position. To apply those assumtion, we need to modify equations of the inner production as

$$\begin{aligned} < V_{I_1}, V_{I_2} > &= \sum_{s \in S} V_{I_1}^s V_{I_2}^s \\ &= \sum_j^h \sum_k^w \sum_{s \in S} Id_{I_1}^s(j,k) Id_{I_2}^s(j,k) \\ &= \sum_j^h \sum_k^w K(j, k, I_1, I_2, n), \end{aligned}$$

where $K(j, k, I_1, I_2, n)$ can be defined as $K(j, k, I_1, I_2, n) = \sum_{s \in S} Id_{I_1}^s(j,k) Id_{I_2}^s(j,k)$. Then, the recursion rules should be modified as

1. If $C(j, k, I_1, I_2)$ is 0,
   - $K(j, k, I_1, I_2, n) = 0$
2. If $C(j, k, I_1, I_2)$ is 1 and two pixels are terminals ($n = 0$),
   - $K(j, k, I_1, I_2, n) = \lambda$
3. else
   - $K(j, k, I_1, I_2, n) = \lambda \prod_i^{np(out)} (1 + K(out_i(j,k).height, out_i(j,k).width, I_1, I_2, n - 1))$,

where the function $C(j, k, I_1, I_2)$ can be defined as

1. If the pixels at jth row and kth column on images are same
   - $C(j, k, I_1, I_2)$ returns 1
2. else
   - $C(j, k, I_1, I_2)$ returns 0.

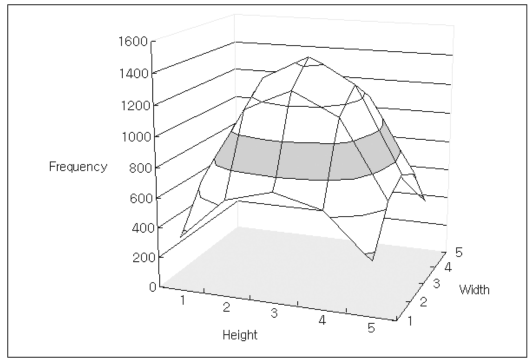

Figure 1. The comparison frequency of each pixels

By adopting Grid kernel, the comparison of each pixel is performed using not only the information of each pixel but also the information of neighboring pixels. Moreover, the kernel downweights the contribution of outer pixels with their distance from the central pixel by fearure definition. It implyes that Grid kernel compares images using the local information and the influence of the local information is different according to their location. Figure 1 shows that the comparison frequency of each pixel in an image fragment which has size of 5. Note that the central pixel has the biggest effect and the outer pixels have less effect during comparison.

3 Support Vector Models for Vehicle Color Recognition

This section presents the support vector models using Grid kernel for vehicle color recognition. The problem of vehicle color recognition can be transformed to the one of classification tasks. To classfy vehicle images according to their color, we generate a support vector model for each class using the Support Vector Machine [10]. Each support vector model considers one color as positive and other colors as negative. SVMs try to find the optimal support vector which can split these two classes and maxmize the distance between positive data and negative data [1]. In generating models and classifying input data, we consider the following assumptions:

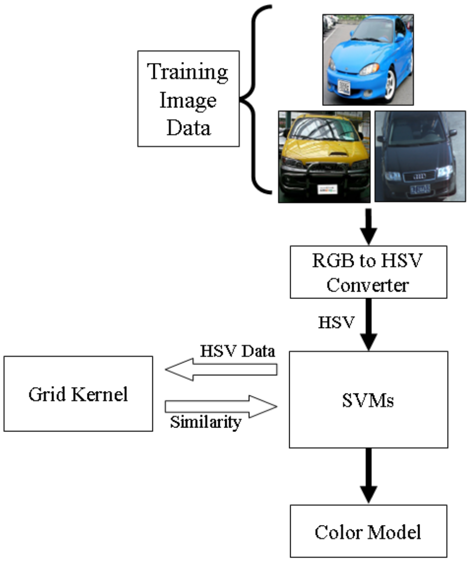

Figure 2. The model generation process

1. Each image has only one vehicle object to detect its color.

2. The vehicle object is located on the nearby center of an image.

Figure 2 shows the process of generating models. This process is composed of 2 steps. The first step of the process is converting an image from RGB (Red-Green-Blue) color to HSV (Hue-Saturation-Value) color. Through this step, we try to reduce the noise from environment. Since our image data are composed of our-door images, there are some information from circumstance such as illumination, wheather, and angle. To split circumstance information from color information, the image is mapped on HSV color space and we use just H, S values because they have color information while V has brightness information. In our models, we devide an HS color space into an 100×100 grid. In the next step, the generated image data are given to SVMs. At this time, grid kernel plays the comparing role at SVMs. In comparing two image data, grid kernel utilizes H and S values to exclude the circumstance information. To do this, we modify the function of C as:

1. If $|pixel(i,j,I_1).h - pixel(i,j,I_2).h| > 10$,

   - $C(j,k,I_1,I_2)$ returns 0

2. If $|pixel(i,j,I_1).s - pixel(i,j,I_2).s| > 10$,

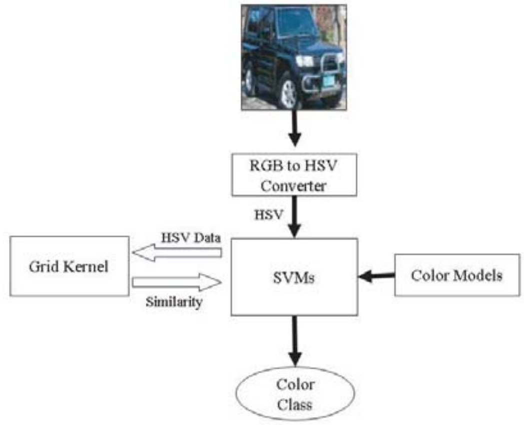

Figure 3. The classification process

   - $C(j,k,I_1,I_2)$ returns 0

3. else,

   - $C(j,k,I_1,I_2)$ returns 1

To classfy images using generated models, test images are converted into test data through same processing of generating model. After that, the test data are applied to SVMs. Figure 3 shows the classification process using SVMs and genenrated models.

4 Experiments

To demonstrate the performance of our kernel and model, we performed experiments with 500 vehicle images which are composed of 5 color classes: black, white, blue, yellow, and red. Each color class has 100 images and every image has the same size of 150×150. See Figure 4 for simple example images of data.

We split these images into a training set of 450 and a test set of 50. SVM light is adopted to construct a color recognition model and classfy the test set. In grid kernel, we set parameters, λ and n as 0.1 and 5 respectively. These values are heuristically decided. The experimental result of each class is reported in Table 1. In this table, relatively better performance is shown on chromatic color classes: blue, red, and yellow. We can obtain a average precision and recall of about 97% on these 3 classes.

To compare the grid kernel with other kernel methods, we experimented with a linear kernel. In this experiment, we give HSV information of each pixel to a linear kernel. The result of this experiment in Figure 5 shows that about 90% of precision and recall are

Figure 4. The frequency of each pixels on image frequency

Table 1. The experimental result using Grid kernel

	Blue	Red	Yellow	White	Black
Precision	90.9%	100%	100%	81.8%	88.8%
Recall	100%	90%	100%	90%	80%

obtained with grid kernel. On the other hands, only about 45% of precision and recall are obtained with linear kernel.

In the previous section, we insisted that the color recognition model with HSV color space will be better than the one with RGB color space. Figure 6 verifies our insistence. In this figure, even though the result of grid kernel using RGB color space shows better performance than one of a linear kernel in Figure 5, it still shows 10% and 25% lower precision and recall comparing to grid kernel using HSV color space.

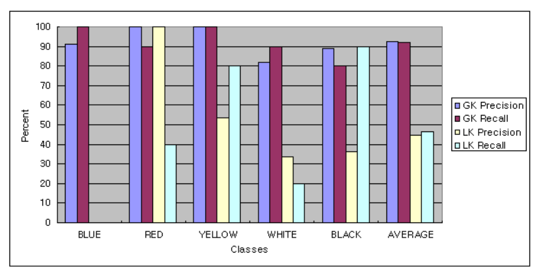

Figure 5. Comparison result with a linear kernel

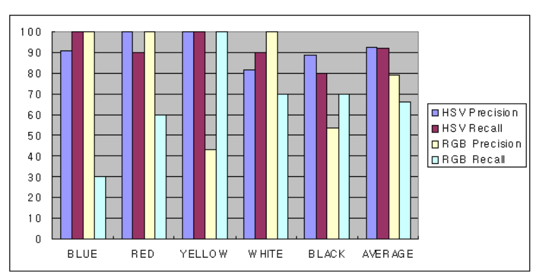

Figure 6. Comparison result with REG color space

5 Conclusion

In this paper, we proposed a novel convolution kernel, grid kernel, and applied it to a vehicle color recognition problem. In the Grid Kernel, image fragments are defined as features and it can make the proposed kernel handle information of a pixel along with its neighboring pixels. The proposed model for a vehicle color recognition is based on support vector machines with grid kernel. It first maps the given images onto HSV color coordinates to reduce the effect of circumstance. After that, SVMs generate models for each color class by comparing image pairs with grid kernel. Finally, test images are classified into an appropriate class using generated models.

The experiments showed that the proposed kernel outperforms a linear kernel. Further experiments on two kinds of color coordinates showed that HSV color coordinates can reduce more efficiently the effect of circumstance factors than RGB color coordinates can do. Consequently, the vehicle color recognition model with the proposed kernel showed promising results which is 92% of precision and 92.4% of recall with depth of 5. It implies that the proposed kernel can extract color information from not a pixel but pixel information obtained from both a pixel and neighboring pixels. Furthermore, since various color coordinates can applied to the proposed kernel, it is expected that we can apply the proposed kernel to diverse applications which use color information.

Even though the proposed kernel showed promising performance for color recognition task, it has high time complexity and any redundant features. Therefore, our future work is to reduce computation time of the proposed kernel and redundant features. We expect the future work can be achieved by some techniques such as kernel PCA [11] and fast algorithm for kernel computation [13].

Ackonowledgments

This work was supported in part by MIC & IITA through IT Leading R&D Support Project and by the Korean Ministry of Education under the BK21-IT Program.

References

[1] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] S. Buluswar and B. Draper. Color recognition in ourdoor images. In *Proceedings of sixth International Conference on Computer Vision*, pages 171–177, 1998.

[3] M. Collins and N. Duffy. Convolution kernels for natural language. In *Proceedings of the 14th Neural Information Processing Systems*, pages 652–632, 2001.

[4] D. Finlayson, B. Schiele, and J. Crowley. Comprehensive color image normalization. In *Proceedings of 5th European Conference on Computer Vision*, pages 475–490, 1998.

[5] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *Advances in Neural Information Processing Systems 20*, 2006.

[6] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-19, University of California, Santa Cruz, 1999.

[7] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[8] A. Moschitti. “a study on convolution kernels for shallow semantic parsing”. In *Proceedings of the 42nd Annual Meeting of the Computational Linguistics*, pages 335–342, 2004.

[9] Y. Rui, T. Huang, and S. Chang. Image retrieval: current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.

[10] B. Schlkopf, J. Burges, and A. Smola. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1999.

[11] B. Schlkopf, A. Smola, and K. Mller. Non-linear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[12] J. Suzuki, T. Hirao, Y. Sasaki, and E. Maeda. Hierarchical directed acyclic graph kernel: Methods for structured natural language data. In *Procedings of the 41th Annual Meeting of Association for Computational Linguistics(ACL 2003)*, pages 32–39, 2003.

[13] S. Vishwanathan and A. Smola. Fast kernels on strings and trees. In *Advances in Neural Information Processing Systems 15*, pages 569–576, 2003.

[14] J. Weijer, T. Gevers, and J. Geusebroek. Color edge detection by photometric quasi-invariants. In *Proceedings of ninth International Conference on Computer Vision*, pages 1520–1525, 2003.

[15] Y.Rubner, J. Puzicha, C. Tomasi, and J. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding*, pages 25–43, 2001.