AUTOMATIC SELECTION AND ADJUSTMENT OF FEATURES FOR IMAGE CLASSIFICATION

Kenji Saiki and Tomoharu Nagao

Graduate School of Environment and Information Sciences
Yokohama National University
79-7 Tokiwadai, Hodogaya-ku,Yokohama, 240-8501 JAPAN
E-mail: saiki@nlab.sogo1.ynu.ac.jp

ABSTRACT

Recently, image classification has been an important task in various fields. Generally, the performance of image classification is not good without the adjustment of image features. Therefore, it is desired that the way of automatic feature extraction. In this paper, we propose an image classification method which adjusts image features automatically. We assume that texture features are useful in image classification tasks because natural images are composed of several types of texture. Thus, the classification accuracy rate is improved by using distribution of texture features. We obtain texture features by calculating image features from a current considering pixel and its neighborhood pixels. And we calculate image features from distribution of textures feature. Those image features are adjusted to image classification tasks using Genetic Algorithm. We apply proposed method to classifying images into "head" or "non-head" and "male" or "female".

Keywords: image classification, feature extraction, Genetic Algorithm, Support Vector Machine

1. INTRODUCTION

Recently, imaging techniques have been more and more developed. And we easily obtain a large number of images. However, it is hard to obtain target images from a large number of images. Therefore, image classification is an important task in various fields. Generally, the performance of image classification is not good without the adjustment of image features. Though certain image feature is useful in a certain image classification task, the image feature is not necessarily useful in other tasks. Thus, automatic feature extraction methods have been proposed [1, 2]. Those methods extract image features from texture images using Genetic Programming (GP) [3]. However, they use monochrome uni-texture images. Therefore, we propose the method of feature extraction from color natural images.

Texture features are useful in image classification tasks because natural images are composed of several textures. Thereby, the classification accuracy rate is improved by using distribution of texture features. In this paper, we adjust image feature by combination of commonly-used image features using Genetic Algorithm (GA) [4].

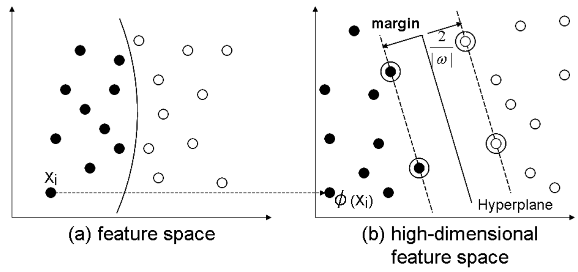

Fig. 1: SVM with kernel trick.

In the following, related works are explained in section 2. In section 3, we describe proposed method. In section 4, we show the experimental results. Finally, in section 5, we describe the conclusions and future works.

2. RELATED WORKS

2.1 Genetic Algorithm

Genetic Algorithm (GA) was proposed by Holland in the 1970s. GA is one of Evolutionary Computation (EC) and optimization algorithm, which is inspired by the evolution of the living thing (e.g. crossover, selection and mutation). That is a simple algorithm. Nevertheless, that effectiveness is shown and that is used widely (e.g. parameter optimization [5], function identification and etc.).

2.2 Support Vector Machine

2.2.1 Concept of SVM

Support Vector Machine (SVM) is one of supervised classification methods. That concept is using maximum-margin hyperplane for separating two classes. Maximum-margin hyperplane means the distances between the hyperplane and the closest points from each class are maximum. In the past, this algorithm is not possible to apply to non-linear space. Therefore, Bernhard Boser, Isabelle Guyon and Vapnik suggested a way to apply SVM to non-linear space by using the kernel trick in 1992 [6]. Figure 1 shows outline of SVM.

2.2.2 *Formalization of SVM*

Discriminant function of SVM is defined as follows

$$f(\mathbf{x_i}) = \omega * \phi(\mathbf{x_i}) + b \quad \begin{cases} \geq 1 & if\ (\mathbf{x_i} \text{ is class A}) \\ \leq -1 & if\ (\mathbf{x_i} \text{ is class B}) \end{cases} \tag{1}$$

where, $\mathbf{x_i}$ is input data, and $\phi(\mathbf{x})$ is function which map from original feature space to high-dimensional feature space. y_i shows the class to which $\mathbf{x_i}$ belongs, and the margin maximization is formulated as follows.

$$y_i = \begin{cases} 1 & if\ (\mathbf{x_i} \text{ is class A}) \\ -1 & if\ (\mathbf{x_i} \text{ is class B}) \end{cases} \tag{2}$$

$$\begin{array}{ll} \text{Minimize} & G(\omega) = \frac{1}{2} \parallel \omega \parallel^2 \\ \text{s.t.} & y_i \cdot (\omega * \phi(\mathbf{x_i}) + b) - 1 \geq 0 \end{array} \tag{3}$$

When this minimization problem is solved by using Lagrange's method of undetermined multipliers,

$$\text{Maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x_i}, \mathbf{x_j}) \tag{4}$$

$$\text{s.t.} \quad \alpha_i \geq 0 \ \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{5}$$

where, α_i is Lagrange multiplier, and $k(\mathbf{x_i}, \mathbf{x_j})$ is a kernel function. Kernel function enables to calculate inner product without calculating $\phi(\mathbf{x})$. That is called kernel trick.

$$k(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i}) * \phi(\mathbf{x_j}) \tag{6}$$

In this paper, we apply Gaussian kernel to kernel function. Gaussian kernel is defined as follows.

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\delta^2}\right) \tag{7}$$

3. PROPOSED METHOD

3.1 Local Feature and Global Image Feature

3.1.1 *Local Feature and Local Feature Image*

In this research, we calculate image features from a current considering pixel and its neighborhood pixels. "Local feature" is defined as the image feature. And we calculate local feature against all pixels in an image. As a result, we obtain the distribution of local feature in an image (we define this distribution as "local feature image"). Moreover, we make local feature images to visualize. Figure 2 shows the sample of local feature images.

3.1.2 *Global Image Feature*

We calculate the image feature from the local feature image. "Global image feature" means the image feature calculated from the whole of the pixels in a local feature image. Global image feature shows distribution of local feature in the image.

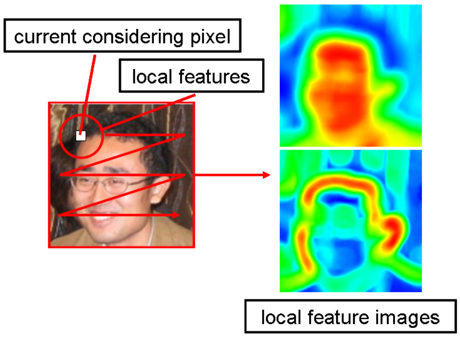

Fig. 2: Local features and local feature images.

3.2 Concept of Proposed Method

Texture features are useful in image classification tasks because natural images are composed of several textures. Therefore, the classification accuracy rate is improved by using distribution of texture features. In this research, we propose the method which obtains texture feature by using local feature and distribution of local feature by using global image feature. Additionally, we select a color component and a window size (which determines "neighborhood pixels") automatically. And we create effective image features in each image classification task. The final target of this research is automatic development of local feature and global image feature. However, there are a lot of time complexities of local feature. Thus, it takes a lot of time to optimize the function of local feature using EC.
In this paper, therefore, we adapt commonly-used image feature as local feature and global image feature. And we adjust image features by those combinations for each task. One adjusted image feature is calculated by a color component, a window size, a type of local feature and a type of global feature. However, the number of adjusted image features is huge. Therefore, we try adjusting features for image classification tasks using GA.

3.3 Setting of GA

3.3.1 *Design of Individual*

Optimized factor is the following four. Those conbinations is 1008. And the number of image feature is adjusted by no operation (NOP) in color component.

1. Color component
   Red, Green, Blue, Saturation, Value, L*, a*, b*, NOP
2. Window size (Radius)
   5 pixels, 9 pixels, 13 pixels
3. Local feature
   Minimum (Min), Maximum (Max), Average, Range, Decentration, Sobel operator, NOP
4. Global image feature
   Median, Average, Range, Decentration, Average of high rank 5% of histogram (AveHigh), Average of subordinate position 5% of histogram (AveSub)

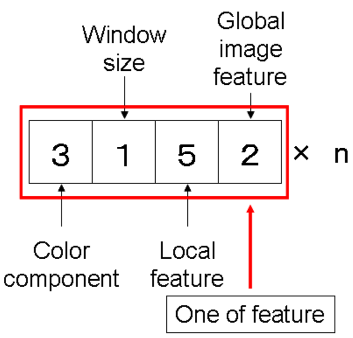

Fig. 3: Design of Individual.

Table 1: Parameter for GA.

	Parameter
Max number of image feature	n = 15
Population size	100
Alternation model	MGG
MGG child number	5
Crossover rate	1.0
Mutation rate	0.0167
Crossover type	uniform

Figure 3 illustrates the genotype of an individual in the proposed method. One image feature is shown by four genes. n is the max number of image feature. Therefor, the individual have 4 n genes.

3.3.2 *Parameters for GA*

The parameters for GA are shown in Table 1. We use Minimal Generation Gap (MGG) [7] as an alternation model.

3.4 Pattern Classifier

In this paper, we adapt SVM as a pattern classifier because SVM is non-random algorithm. Therefore, it is possible to evaluate individuals uniquely. However, because SVM in this paper is a hard margin, the accuracy rate becomes 100% for the data used when the pattern classifier is made. Thus, we use three data sets: to make SVM, to evaluate individuals (For GA), to evaluate the adjusted features (Unknown).

4. RESULTS

4.1 Head or Non-head task

4.1.1 *Data Set of Head or Non-head task*

We apply proposed method to classifying images into "head" or "non-head". Where, "head" images are contained not only the front face but also the profile and back of the head. Figure 4 shows the sample of images. Those size are 64 64 pixels. The total number of images is 1750. Table 2 shows allotment of images.

Fig. 4: Sample images of head and non-head.

Table 2: The number of data: Head and non-head.

	Head	Non-head
SVM	200	200
For GA	200	200
Unknown	350	600

Table 3: Comparison of classification accuracy rate head or non-head.

	For GA	Unknown
Proposed method	92.2%	81.6%
HLAC	70.8%	72.2%

Table 4: Accuracy rate and number head or non-head of proposed method.

	Head	Non-head
For GA	92.7% (188.5 / 200)	91.7% (183.4 / 200)
Unknown	89.8% (314.3 / 350)	76.9% (461.1 / 600)

4.1.2 *Result of Head or Non-head task*

Table 3 illustrates the experimental result of head or non-head. This result is average of 10 trials. In this experiment, we adapt Higher-order Local Auto-Correlation (HLAC) [8] as a comparative image feature. We calculate HLAC feature using Red, Green and Blue. Proposed method shows high-performance for classification and uses only 14.9 image features on average. However, accuracy rate of unknown data fell more than evaluate individuals data (For GA). The cause is a variety of "non-head". Table 4 shows accuracy rate of non-head lower than head in unknown data. Table 5 shows the image features which appear frequently. Window size often used is 5 pixels. In this experiment, image size is not large. Therefore, proposed method adjusts window size to image size appropriately.

4.2 Male or Female task

4.2.1 *Data Set of Male or Female task*

In this experiment, we apply our method to classifying images into "male" or "female". We assemble images from Yahoo! JAPAN (http://www.yahoo.co.jp/). Those images are full color images and that background is unstandardized.

Table 5: Adjusted features for head or non-head.

Occurrence count	15	9	8
Color component	a*	Green	b*
Window size	5 pixels	5 pixels	5 pixels
Local feature	Min	Mean	Variance
Global image feature	AveHigh	Range	Mean

Table 6: The number of data: Male and female.

	Male	Female
SVM	70	70
For GA	70	70
Unknown	60	60

Table 7: Comparison of classification accuracy rate male or female.

	For GA	Unknown
Proposed method	93.5%	70.4%
HLAC	65.7%	70.8%

Table 8: Adjusted features for male or female.

Occurrence count	17	13	7
Color component	a*	a*	b*
Window size	9 pixels	9 pixels	13 pixels
Local feature	Min	Max	Min
Global image feature	AveHigh	Median	AveSub

The total number of images is 400. Size of those images is 200 200 pixels. Allotment of images is shown in Table 6.

4.2.2 *Result of Male or Female task*

Table 7 shows the experimental result of classifying images into male or female. This result shows average of 10 trials. Proposed method shows high-performance for evaluate individuals data (For GA). Nonetheless, our method uses only 14.6 image features on average. That means proposed method selects and adjusts image feature automatically. However, we apply adjusted image feature to unknown data, the accuracy rate greatly fell. This is thought that the over-training happened because proposed method has approximately 1000 image features. The image features which appear frequently are shown in Table 8.Window size used in this experiment is larger than used in pre-experiment. Proposed method corresponds to the difference of image size.

5. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a classification method which automatically adjusts image features for image classification tasks. One image feature is calculated by a color component, a window size, a type of local feature and a type of global feature. We adjust those image features to the task using GA. The experimental results show proposed method shows high-performance to evaluate individuals data. Additionally, proposed method adjusts effective image feature. However, the over-training happened. We suppose that the cause is performance of global image feature. The ability to extract feature from distribution of local feature of global image feature is insufficient.

Therefore, in the future, we plan to improve global image feature. Additionally, we propose the way of effective use of local image feature (e.g. the understanding of adjusted features).

6. REFERENCES

[1] Brian Lam and Vic Ciesielski. “Discovery of Human-Competitive Image Texture Feature Extraction Programs Using Genetic Programming,” GECCO 2004, LNCS 3103, pp. 1114–1125, 2004.

[2] Melanie Aurnhammer. “Evolving Texture Feature by Genetic Programming,” EvoWorkshops 2007, LNCS 4448, pp. 351–358, 2007.

[3] John R. Koza. “Genetic Programming: On the Programming of Computers by Means of Natural Selection,” MIT Press, Cambridge, MA, USA, 1992.

[4] J.H.Holland. “Adaptation in Natural and Artificial Systems,” The Univ. Michigan Press, 1975.

[5] V.Ayala-Ramirez, C.H.Garcia-Capulin, A.Perez-Garcia and R.E. Sanchez-Yanez. “Circle detection on images using genetic algorithms,” Pattern Recognition Letters 27, 2006.

[6] Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik. “A training algorithm for optimal margin classifiers,” 5th Annual Workshop on Computational Learning Theory, Pittsburgh, ACM, 144–152, 1992.

[7] Hiroshi Sato, Isao Ono, Shigenobu Kobayashi. “A New Generation Alternation Model of Genetic Algorithms and Its Assessment,” Journal of Japanese Society for Artificial Intelligence, Vol.12, No.5(19970901) pp. 734–744 (in Japanese).

[8] N.Otsu and T.Kurita. “A new scheme for practical, flexible and intelligent vision systems,” Proc. IAPR Workshop on Computer Vision, pp.431–435, 1988.