Image Classification Using Naïve Bayes Classifier

Dong-Chul Park

***Abstract*—**An image classification scheme using Naïve Bayes Classifier is proposed in this paper. The proposed Naive Bayes Classifier-based image classifier can be considered as the maximum a posteriori decision rule. The Naïve Bayes Classifier can produce very accurate classification results with a minimum training time when compared to conventional supervised or unsupervised learning algorithms. Comprehensive experiments for pattern classification tasks on an image dataset are performed in order to evaluate the performance of the proposed classifier. The results show that the proposed Naïve Bayes Classifier outperforms conventional classifiers in terms of training speed and classification accuracy.

***Keywords*—** Bayes classifier, image classification, DCT feature, neural networks.

I. Introduction

THE image classification task is one of the ongoing important topics in various computer vision tasks. A rapid increase in the size of data in various areas has been witnessed recently. Automatic processing of these contents requires efficient pattern classification techniques. In general, automatic data classification tasks including image retrieval tasks require two critical processes: an appropriate feature extraction process and an accurate classifier design process. For image classification tasks, a feature extraction process can be considered the basis of content-based image retrieval. Features can be classified into two groups: general features and domain-related features. General features include colors and textures, and the domain-related features include faces, fingerprints, and human irises. Among these available features, we never know which one or which combination of features is suitable for characterizing an image perfectly from the other images. The best strategy to utilize the available features may be the one that uses all of them if there exists any classifier that can use all the available features efficiently. With the use of various methods, different features are extracted for different reasons. Nevertheless, all the features can help to describe objects more precisely. In most cases, more features facilitate a more accurate classification. Note that each feature has different properties including magnitude. When using different features for describing an object, different magnitudes may cause a problem because each feature is independent of the others. In the case of a set of combined features that consist of different individual features for a classification problem, each feature is usually normalized first. A normalization process is required for using the different available features. However, there is no justification for making all the different individual features have the same minimum and maximum magnitudes in the normalization process. Therefore, it is necessary to devise a more effective approach to deal with different individual features.

Thus far, various approaches, including ensemble classifiers, to utilize different feature vectors have been proposed [1, 2]. A classifier model called partitioned feature-based classifier (PFC), which efficiently uses a variety of available features extracted through various tools and enhances the classification performance, has also been proposed [3]. In PFC, all the available features are grouped into several groups, where each group has homogeneous features and forms a feature vector. Each feature vector in PFC is separately used in the independent classifier and can preserve the properties of the individual features in the same group. Each local classifier is independently trained with a specific group of features. Since each feature group is used only for a local classifier in the training stage independently, features from different groups will not interfere with each other. In PFC, each local classifier achieves specific classification accuracy during the training stage. The accuracy for each local classifier using a group of features is then used as a weight for the local classifier when making decisions at the test stage. The PFC model demonstrates that the classification accuracy of conventional clustering algorithms can be significantly improved when the PFC model is used with them [3]. For the given data, however, the PFC model draws the classification results on the basis of each local classifier's performance irrespective of how each classifier performs on each class. If we know from the training dataset that a certain trained local classifier classifies very well on a certain class of data while it gives very inaccurate classification results on another class of data, then it may be reasonable to make a final decision according to the classification results from the local classifier.

In order to improve the classification accuracy, the classifier integration model was proposed as a fusion method for multiple classifiers [4]. As a classifier fusion algorithm, individual local classifiers in CIM are applied in parallel and their outputs are combined in a certain manner to reach an optimal decision. As a multiple sensor data fusion method, the CIM combines feature data from multiple sensors or multiple

Dong-Chul Park is with Department of Electronics Engineering, Myong Ji University, Yong In, Gyeonggi-do, 449-728, South Korea (e-mail: parkd@mju.ac.kr).

feature extraction methods to achieve improved accuracies and more specific inferences as compared to those that could be achieved by the use of a single sensor or feature alone.

In this study, in order to achieve more accurate classification accuracy and higher training speed than conventional learning algorithms, a Naive Bayes algorithm [9] based on a stochastic process is adopted for local classifiers. Since the Naive Bayes algorithm does not require any iterative procedure for its training process, its training process is quite simple and requires a small amount of training data to estimate the parameters while achieving very competitive accuracy of the classification results.

The remainder of this paper is organized as follows: Section 2 summarizes the Naive Bayes algorithm adopted for local classifiers in this study. The proposed SCIM is applied to two pattern classification problems in order to evaluate the performance of the proposed algorithm in terms of training speed and classification accuracy in Section 3. The concluding remarks are presented in Section 4.

II. Naive Bayes classifier

The Naive Bayes classifier is based on Bayes' theorem of probability [1]. In Bayes' theorem, the conditional probability that an event $\boldsymbol{x}$ belongs to a class k can be calculated from the conditional probabilities of finding particular events in each class and the unconditional probability of the event in each class. That is, for given data, $\boldsymbol{x} \in X$, and C classes, where X denotes a random variable, the conditional probability that an event $\boldsymbol{x}$ belongs to a class k can be calculated by using the following equation:

$$P(c_k|\boldsymbol{x}) = P(c_k)\frac{P(x|c_k)}{P(x)} \tag{1}$$

Equation (1) shows that the calculation of $P(c_k|\boldsymbol{x})$ is a pattern classification problem since it finds the probability that the given data $\boldsymbol{x}$ belongs to class k and we can decide the optimum class by choosing the class with the highest probability among all possible classes, C, which can minimize the classification error. For doing so, we need to estimate $P(\boldsymbol{x}|c_k)$ and assume that any particular value of vector x conditional on c_k is statistically independent of each dimension and can be written as follows:

$$P(\boldsymbol{x}|c_k) = \prod_{i=0}^{n} P(\boldsymbol{x}|c_k) \tag{2}$$

where $\boldsymbol{x}$ is a n-dimensional vector data $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$.

The Naive Bayes classifier is based on equation (2) and assumes that each feature be statistically independent [2]. This assumption results in simpler calculation cost and efficient data processing. By combining equation (1) and equation (2), the Naive Bayes classifier can be summarized as the following equation:

$$k = argmax_k\, P(c_k) \prod_{i=0}^{n} P(x_i|c_k) \tag{3}$$

where the denominator $P(x)$ is omitted since the value is the same for all class.

The Naive Bayes classifier is often referred as the maximum a posteriori (MAP) decision rule. Note that the assumption of statistically independence in each feature sometimes does not hold in certain cases and causes problems in some practical cases [3]. However, various applications and experimental studies show that training schemes based on the MAP decision rule with the Naive Bayes assumptions yield an optimal classifier even when the assumption does not hold.

The CNN [4] is utilized as the local classifier in this paper. The CNN is an unsupervised competitive learning algorithm based on the classical k-means clustering algorithm. It finds the centroids of clusters at each presentation of the data vector. The CNN first introduces definitions of the winner neuron and the loser neuron. When a data xi is given to the network at the epoch *(k)*, the winner neuron at the epoch *(k)* is the neuron with the minimum distance to xi. The loser neuron at the epoch *(k)* to x_i is the neuron that was the winner of xi at the epoch *(k-1)* but is not the winner of x_i at the epoch (*k*). The CNN updates its weights only when the status of the output neuron for the presenting data has changed when compared to the status from the previous epoch.

When an input vector x is presented to the network at epoch n, the weight update equations for winner neuron *j* and loser neuron *i* in CNN can be summarized as follows [4]:

$$w_j(n+1) = w_j(n) + \frac{i}{N_j+1}\left[x(n) - w_j(n)\right]$$
$$w_i(n+1) = w_i(n) - \frac{1}{N_i+1}\left[x(n) - w_i(n)\right] \tag{4}$$

where $w_j(n)$ and $w_i(n)$ represent the weight vectors of the winner neuron and the loser neuron, iteration, respectively.

The CNN has several advantages over conventional algorithms such as SOM or k-means algorithm when used for clustering and unsupervised competitive learning. The CNN requires neither a predetermined schedule for learning gain nor the total number of iterations for clustering. It always converges to sub-optimal solutions while conventional algorithms such as SOM may give unstable results depending on the initial learning gains and the total number of iterations. Note that the CNN was designed for deterministic data because the distance measure used in the CNN is the quadric (Euclidean) distance. More detailed description on the CNN can be found in [4].

Conventional classifiers calculate the local classification decision probability and uses the information for parameters for making a global classification decision. However, the proposed NBC calculates the classification probability by using equation (1) directly when the model adopts a Naive Bayes classifier as its local classifier. When the feature value is a continuous value, the proposed NBC estimates the probability that a feature vector component is classified as its class with the following probability density function:

$$P(x_i|c_k) = \frac{1}{\sqrt{2\pi}\sigma_{c_k}} e^{-\left(x_i-\mu_{c_k}\right)^2/2\sigma^2_{c_k}} \tag{5}$$

where the probability density function is formed during the training stage of local classifiers with the mean, μ_{c_k}, and standard deviation, σ_{c_k}, of each i-th class data for each feature vector component x_k

Note that $P(\boldsymbol{x}|c_k)$ can be calculated by using equation (2) with each dimension independently because NBC adopts the Naive Bayes classifies as its local classifiers.

During training procedure, the probability density function, $P(x_i|c_k)$ for each dimension of feature vector on each local classifier is first calculated and the $P(\boldsymbol{x}|c_k)$ for each local classifier is found by using equation (2). Once the probability density function is found for each local classifier, the training procedure for the global classifier is terminated. The decision making procedure for a given data is that the feature vectors are pass through corresponding local classifiers and the class for the given data is found by the following equation:

$$Class(\boldsymbol{x}) = argmax_k \frac{1}{N} \sum_{j=1}^{N} P(c_{ij}) \prod_{i=1}^{n} P(x_i|c_{jk}) \qquad (6)$$

III. Experiments And Results

For experiments, image data sets are collected from the Caltech image data set. The Caltech image data set consists of different image classes (categories) in which each class contains different views of an object. The Caltech image data were collected by the Computational Vision Group and are available at the following website :

http://www.vision.caltech.edu/html-files/archive.html

Figure 1 shows examples of airplane, car, motorbike, and bike images used in our experiments. Each class consists of 200 images with different views resulting in a total of 800 images in the data set. Before any further processing for feature extraction, the entire image sets were converted to grey scale data with the same resolution

In order to describe the characteristics of each class, the feature vectors should be able to discriminate the images from different categories while producing similar feature values on the images from the same category in order to classify the image data in the same class. The following features are employed in the experiments:

A. Discrete cosine transform (DCT)

Discrete cosine transform (DCT) is a tool to convert an image into frequency components and has been successfully applied to the image compression problem [5]. DCT can decompose an image signal into the underlying spatial frequencies, and the DCT coefficients of an image signal can be used as new features that have the ability to represent the regularity and some textural features of the image signal. In order to describe the DCT texture information of image signals, a localized image representation method computes the DCT coefficients at different points of interest in the image signal.

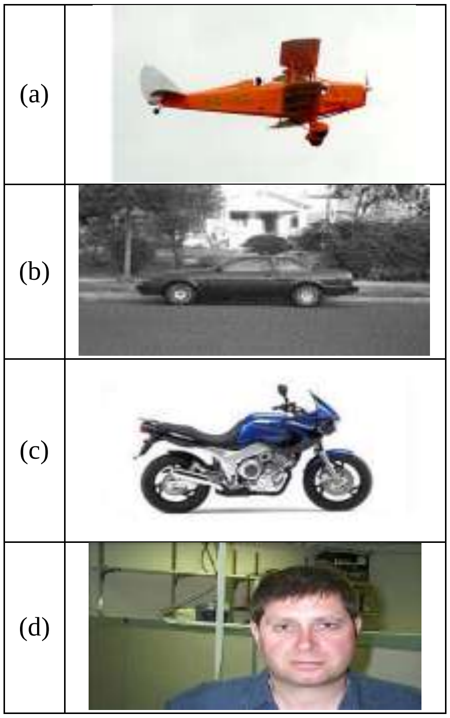

Fig 1: Example of Caltech image data set:(a) Airplane (b) Car (c) Motorbike (d) Face

For the localized representation, the DCT coefficients are first calculated with 8×8 window at the upper left corner of the satellite image. The 8×8 sliding window for calculating the next DCT coefficients is then shifted by an increment of 2 pixels horizontally and vertically. The DCT coefficients from each block consist of 64 dimensional coefficients and 32 lowest frequency DCT coefficients from each image block are used as a feature for the local block in our experiments because most of the energy is concentrated in these frequency region. Therefore, the DCT feature vectors used in our experiments are 32 dimensional vectors.

B. Local Binary Pattern (LBP)

The local binary pattern (LBP) is one of the most widely used feature extraction methods for describing image textures including points, lines, and surfaces because of its texture representation capability and computational simplicity [6]. The most important advantageous feature of LBP in practical uses is that the LBP is very robust to monotonic data value variations caused by illumination changes. The LBP operator labels the pixels of an image signal by applying a threshold on the neighborhoods of each pixel with the center value and producing the result as a binary number. As shown in Fig. 4, at a given pixel position (x_c,y_c), the LBP is defined as an ordered set of binary comparisons of pixel intensities between the center pixel and its predetermined S_p surrounding pixels (in this case, S_p = 8).

Uniform local binary pattern (ULBP) is a variant of a local binary pattern (LBP) and has at most two circular 0 to 1 or 1 to 0 transitions. ULBP can reduce the length of the feature histogram and improve the classifier performance by using LBP

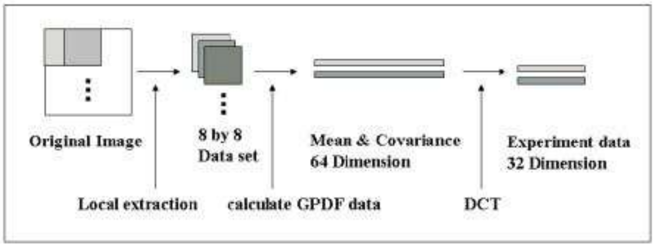

Fig 2: DCT feature extraction

TABLE I
COMPARISON OF CLASSIFICATION ACCURACY ON CNN CLASSIFIER WITH DIFFERENT FEATURE VECTORS (MEAN AND STANDARD VARIATION)

Feature	Accuracy(%)	
	Average	Std. Dev.
DCT	71.8	7.68
ULBP	67.4	10.12
CovD	63.2	9.81
WPT	58.5	11.28

features [6]. While the size of the LBP used in our experiments is 8 and there exist possible cases, the ULBP for this case allows only 59 cases; the resulting ULBP produces 59-dimensional feature vectors.

C. Covariance Descriptor(Cov.D)

The covariance feature vector has been widely used in object recognition and pattern classification problems for its advantageous features including robustness against noises and low dimensionality when compared with other feature descriptors [7]. For our experiments, the covariance feature vectors formed with the first derivative (G_x, G_y), the second derivative (G_{xx}, G_{yy}), and the angle of image I as shown in equation (10) are used.

$$\boldsymbol{F}(x,y) = [\, G_x, G_y, G_{xx}, G_{yy},\ \sqrt{G_x^2 + G_y^2}, tan^{-1}\frac{G_y}{G_x}\,] \qquad (10)$$

D. Wavelet Packet Transform (WPT)

The wavelet transform is a useful multi-resolution analysis tool and it has been widely applied to texture analysis and classification. The features of images, such as edges of an object, can be projected by the wavelet coefficients in low-pass and high-pass sub-bands. In our experiments, we used 6 step 2-D wavelet packet transform and produced 68 dimensional feature vector for each image.

The dimensions of each feature vector obtained from DCT, ULBP, CovD, and WPT are 36, 59, 36, and 68, respectively. Throughout the experiments, the 10-fold cross-validation method is adopted to deal with the small sample size of our datasets. That is, the datasets are divided randomly into ten roughly equal parts. The first nine parts are used for training the classifiers, and the remaining part is used for evaluating the classifiers. The above process is repeated ten times so that each part is used once as the test dataset.

Table I summarizes the average classification accuracy among three individual classifiers with a separate feature vector for CNN classifier. As can be seen from Table I, individual classifiers with separate features show limited classification accuracy, while the individual classifier with a DCT feature gives the most accurate classification performance among the four individual classifiers. Based on these observations, DCT feature is selected for further experiments.

In order to describe the texture information of images, a localized image representation method is employed. The localized representation method represents the content of the image by a collection of local features extracted from the image.

These features are computed at different points of interest in the image. Afterwards, the Gaussian distribution, wherein the mean vector and the covariance matrix are estimated from all local feature vectors obtained from the image, is used to represent the content of the image. Localized representation maintains the dimensions of the feature vectors tractable by adjusting the sizes of blocks. This method is consequently more robust to occlusions and clutters. In order to obtain the texture information from the image, conventional texture descriptors based on a frequency domain analysis such as Gabor filters [8] and wavelet filters [9] are often used. However, these algorithms often induce a high computational load for feature extraction and are not suitable for real-time applications. In this paper, the Discrete Cosine Transform (DCT) is adopted for extracting the texture information from each block of the image [10]. The DCT transforms the image from the spatial domain into the frequency domain. For the localized representation, images are transformed into a collection of 8×8 blocks. Each block is then shifted by an increment of 2 pixels horizontally and vertically. The DCT coefficients of each block are then computed and returned in 64 dimensional coefficients. Only the 32 lowest frequency DCT coefficients that are visible to the human eye are kept. Therefore, the feature vectors that are obtained from each block have 32 dimensions. In order to calculate the GPDF for the image, the mean vector and the covariance matrix are estimated from all blocks obtained from the image. Finally, a GPDF with a 32-dimensional mean vector and a 32 × 32 covariance matrix is used to represent the content of the image. Figure 2 summarizes the data processing procedure used in our experiment.

Table II is a summary of classification accuracies among different classifiers when DCT is adopted as the feature for classifiers. Note that the Naïve Bayes-based classifier outperforms other classifiers. The classification accuracy for CNN, FCM, MLPNN, and NB is 71.8%, 66.5%, 72.6%, and 77.2% on average, respectively.

Table III summarizes the training time required for different classifiers for the image dataset. Since Multi-layered Perceptron type Neural Network (MLPNN) adopts Error Back-Propagation learning algorithm adaptively, the training times for MLPNN are estimated with the maximum of training times for the four classifiers. As can be seen from Table III, Naïve Bayes requires the minimum training time among the four classifiers. Note that this training time advantage of Naïve Bayes comes from the fact that Naïve Bayes is not iterative training algorithm.

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY ON DIFFERENT CLASSIFIER WITH DCT FEATURE VECTORS (MEAN AND STANDARD VARIATION)

Classifier	Accuracy(%)	
	Average	Std. Dev.
Centroid Neural Network	71.8	7.68
Fuzzy C-Means	66.5	9.62
Multi-Layer Perceptron Neural Network	72.6	8.04
Naïve Bayes	77.2	7.16

TABLE III
COMPARISON OF TRAINING TIME

Classifier	Training time (s)
Centroid Neural Network	1.22
Fuzzy C-Means	1.86
Multi-Layer Perceptron Neural Network	4.22
Naïve Bayes	0.42

IV. CONCLUSION

A classification model for image data is proposed by using Naïve Bayes classifier is proposed in this paper. Since the Naive Bayes classifier does not require any excessive training procedure commonly required in most of the artificial neural network architectures, the resulting classifier can yield an appropriate classification decision with very limited computational efforts. The proposed classifier utilizes the Naive Bayes classifier for minimizing the training time over the conventional classifiers while yielding accurate classification results by adopting the advantage of the probability concepts of the Naive Bayes classifier. In order to evaluate the performance of the proposed classifier, experiments on Caltech image data sets are carried out. The performance of the proposed classifier is compared with those of conventional classifiers such as CNN, FCM, and MLPNN in terms of training speed and classification accuracy. When the proposed classifier is compared with the conventional classifiers in terms of training time and classification accuracy, the results show that the proposed classifier outperforms the conventional classifiers in terms of both training speed and classification accuracy. The advantage of training speed of the proposed classifier over the conventional classifiers is an advantageous feature in practical applications. Further research on how to overcome the assumption of the independence of individual feature dimension in the Naive Bayes classifier will be a subject in future research.

ACKNOWLEDGMENT

This work was supported by the IT R&D program of The MKE/KEIT (10040191, The development of Automotive Synchronous Ethernet combined IVN/OVN and Safety control system for 1Gbps class).

REFERENCES

[1] M. Jang, D.Park, "Stochastic Classifier Integration Model," *International Journal of Applied Engineering Research,* vol. 11, no.2, pp. 809-814, 2016.

[2] D. Lowd, P. Domingos, "Naive Bayes models for probability estimation," *in proc. of the 22th International Conference on Machine Learning*, 2005, pp. 529-536.

[3] D. Lewis, "Naive Bayes at forty: The independence assumption in information retrieval," *Lecture Notes in Computer Science*, vol. 1398, pp. 4-15, June 2005.

[4] D.C. Park. "Centroid Neural Network for Unsupervised Competitive Learning", *IEEE Transaction on Neural Networks*, vol. 11, no.8, pp.520-528, March 2000.

[5] G. Strang. "The discrete cosine transform", *SIAM Rev*, vol. 41, no. 1, pp. 135-147, 1999

[6] Z. Guo, L. Zhang, D. Zhang. "A completed modeling of local binary pattern operator for texture classification", *IEEE Transaction on Image Processing*, vol. 19, no.6, pp.1657-1663, March 2010.

[7] O. Tuzel, F. Porikli, P. Meer. "Region covariance: A fast descriptor for detection and classification", *in proc. of Ninth European Conf. Computer Vision*, vol. 2, 2006, pp. 589-600.

[8] J. Daugman," Complete Discrete 2D Gabor Transform by Neural Networks for Image Analysis and Compression," *IEEE Trans. Acoust.,Speech, and Signal Processing*, vol. 36, pp 1169-11179, 1988.

[9] C. Pun,M. Lee," Extraction of Shift Invariant Wavelet Features for Classification of Images with Different Sizes," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol.26, no.9, pp. 1228-1233, 2004.

[10] Y. Huang, R. Chang, " Texture Features for DCT-Coded Image Retrieval and Classification," *in proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 6, pp.3013-3016, 1999

Dong-Chul Park received the B.S. degree in electronics engineering from Sogang University, Seoul, Korea, in 1980, the M.S. degree in electrical and electronics engineering from the Korea Advanced Institute of Science and Technology, Seoul, Korea, in 1982, and the Ph.D. degree in electrical engineering, with a dissertation on system identifications using artificial neural networks, from the University of Washington (UW), Seattle, in 1990. From 1990 to 1994, he was with the Department of Electrical and Computer Engineering, Florida International University, The State University of Florida, Miami. Since 1994, he has been with the Department of Electronics Engineering, MyongJi University, Korea, where he is a Professor. From 2000 to 2001, he was a Visiting Professor at UW. He is a pioneer in the area of electrical load forecasting using artificial neural networks. He has published more than 140 papers, including 40 archival journals in the area of neural network algorithms and their applications to various engineering problems including financial engineering, image compression, speech recognition, time-series prediction, and pattern recognition. Dr. Park was a member of the Editorial Board for the IEEE TRANSACTIONS ON NEURAL NETWORKS from 2000 to 2002. He is a Senior Member of IEEE.