Volume 4, Issue 3, March 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

On the Selection of Appropriate Kernel Function for SVM in Face Recognition

Justice Kwame Appati*, Gideon Kwadzo Gogovi, Gabriel Obed Fosu

Department of Mathematics
Kwame Nkrumah University
of Science and Technology.
Kumasi-Ghana, India

Abstract—Biometric researchers have paid critical attention to feature extraction since without this process it is virtually impossible to perform recognition. The weaker the features, the weaker the recognition rate, likewise, the stronger the features the more likely to achieve a better recognition rate. However, the later is not always the case since after feature extraction, a classifier is required to test how robust and invariant this features extracted. This classifiers depends highly on the choice of appropriate kernel functions. We determined an appropriate kernel function suitable for the face94 database. We showed that, the selection of Multilayer Perceptron (MLP) kernel function with weight 1 and bias -1 is appropriate for this selected database since it recorded the highest recognition rate. The Support Vector Machine was used as the classifier for this research.

Keywords— Kernel Function; Support Vector Machine; Biometric; Feature Extraction; Face Recognition

I. INTRODUCTION

Biometric, being the authentication and identification of human, based on their physiological or behavioural characteristics such as voice, face, fingerprint and gait has been in existence for quite a long time. Recent trend shows that more work is still in session as far as these characteristics are concern in the field of biometric [1 - 3]. However, each characteristic aforementioned possesses its own peculiar pros and cons and its choice is solely dependent on the problem at hand.

In the past years, Vapnik and his co-workers [4] proposed the Support Vector Machines (SVMs) as an effective method for purpose pattern recognition and classification. From their proposed method, SVM finds the hyper-plane that separates the largest possible fraction of points of the same class to one side while maximizing the distance from either class to the hyper-plane, given that, the set of points belongs to two classes. According to Vapnik [4], this hyper-plane is called Optimal Separating Hyper-plane (OSH) which minimizes the risk of misclassifying not only the object in the training set but also the unseen object of the test. Osuna *et al* [5] train SVM for face detection, where the discrimination is between two classes: face and non-face, each with thousands of examples. Pontil and Verri [6] also use the SVMs to recognize 3D objects which are obtained from the Columbia Object Image Library (COIL) [7]. However, the appearances of these objects are explicitly different and hence the discriminations between them are not too difficult.

Roobaert *et al* [8] repeat the experiments and argue that even a simple matching algorithm can deliver nearly the same accuracy as SVMs. They concluded that the advantage of using SVMs is not obvious. However, it is difficult to discriminate or recognize different persons (hundreds or thousands) by their faces [9] because of the similarity of faces. Guodong *et al*, in their research focus on the face recognition problem and show that the discrimination functions learned by SVMs can give much higher recognition accuracy than the popular standard eigenface approach [10]. They represent face images using the eigenfaces and performed feature extraction on them of which the discrimination functions between each pair are learned by SVMs.

In this paper, face as biometric characteristic was considered and as can be seen, researchers have worked extensively on face feature extraction techniques and applying SVM as a classifier during recognition and classification. However, to the best of our knowledge, little has been done to investigate the effect of the various standard kernel functions available that aid the SVM to arrive at a better recognition rate during algorithm performance analysis. Since the kernel function is the heart of SVM, this paper seeks to identify an appropriate kernel function suitable for the selected face database in the application of SVM as a classifier during face recognition after feature extraction stage.

II. MODEL

A. *Support Vector Machines*

Support Vector Machines (SVMs) are supervised learning algorithm for classification and is used for building models to predict whether a new object belong to a particular class or the other given a training sample with each belonging to a specified group. In a linear support vector machine, a set of N support vector z_1, z_2, ..., z_N and weight w_1, w_2, ..., w_N are given in the computation of SVM which is given by Equation (1). However, using kernels, the original formulation for the SVM is rewritten as shown in Equation (2):

$$F(x) = \sum_{i=1}^{N} w_i < z_i, x > + b \qquad (1)$$

$$F(x) = \sum_{i=1}^{N} w_i k(z_i, x) + b \qquad (2)$$

B. *The Kernel trick*

The Kernel trick in the language of machine learning provides a useful mathematical tool for bridging any algorithm that depends strictly on the dot product between two vectors from linearity to non-linearity. Owing to this kernel trick, mapping does not need to be computed hence, any algorithm that can be expressed as inner product just need a replacement of that inner product with inner product from another space which is referred as the Kernel Function. Using this function, the algorithm can then be transformed into higher-dimension to make classification possible. The kernel function as an inner product in feature space is represented as:

$$k(x, y) = < \varphi(x), \varphi(y) > \qquad (3)$$

C. *Kernel Methods*

These are class of algorithms for pattern analysis and recognition which works by mapping data into higher dimensional space with the hope that data could be easily separated or better structured. However, for this to be possible, the kernel function must have the following property that is:

- Continuity of the function
- Symmetry
- Positive (semi-) definite

D. *Appropriate Choice of Standard Kernel Functions*

In machine learning, choosing an appropriate kernel highly depends on the problem at hand. It is possible to fine tune the parameters of a kernel function, however, it sometimes become a tedious and cumbersome task to handle hence the need for automatic kernel selection.

Below are list of standard kernel functions available which will be studied in this paper and these are [11]:

1. Linear Kernel

Linear kernels are the simplest kernel functions given by the inner product <*x*, *y*> plus an optional constant *c*. Algorithms using a linear kernel are usually equivalent to their non-kernel counterparts.

$$k(x, y) = x^T y + c \qquad (4)$$

2. Polynomial Kernel

Polynomial kernels are non-stationary kernel suitable for problems where all its training data are normalised. This kernel has adjustable parameters thus alpha, a constant term *c* and the polynomial degree *d*.

$$k(x, y) = (\alpha x^T y + c)^d \qquad (5)$$

3. Gaussian Kernel

Gaussian kernels are examples of Radial Basis Function (RBF) kernel. They also have adjustable parameters such as sigma which plays a major role in the performance of the kernel. An overestimation of this parameter causes the kernel behave almost linear and the higher-dimensional projection will start to lose its non-linear power. On the other hand, an underestimation of the parameter makes the kernel function lack regularization hence making the decision boundary highly sensitive to noise in training data.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\delta^2} \right) \qquad (6)$$

4. Hyperbolic Tangent (Sigmoid) Kernel

The Hyperbolic Tangent Kernel which is also known as the Sigmoid Kernel or the MultiLayer Perceptron (MLP) kernel developed from the field of Neural Networks, where the bipolar sigmoid function is often used as an activation function for artificial neurons. There are two adjustable parameters for this kernel, the slope alpha and the intercept constant *c*.

$$k(x, y) = \tanh(\alpha x^T y + c) \qquad (7)$$

E. *Matching*

The final feature vector of the test face image is determined and compared with the trained feature vector. The Euclidean Distance is then computed between trained face database and the test face data. The image matched index which recorded the list Euclidean Distance is selected with the assumption that, it is the correct index for the test image. This is then verified with the original index of the test image. If the indices are the same then a match is recorded otherwise miss-match.

$$\|X - A\|_2 = \sqrt{\sum_{i=1}^{n}(x_i - a_i)^2} \quad (8)$$

where:

X is the feature vector of the trained image.
A is the feature vector of the test image
n is the number of elements in the feature vector
x_i is the element of the trained feature vector
a_i is the element of the test feature vector

III. Analysis and Discussion

A. *Source of Data*

Facial images used for this study were taken from Faces94 database which is composed of 152 individuals with 20 been female, 112 male and 20 male staff kept in separate directories. These separate directories were merged into one to achieve the different lighting effect. The subjects were sited at approximately the same distance from the camera and were asked to speak while a sequence of twenty images was taken and this was to introduce moderate and natural facial expression variation. Each facial image was taken at a resolution of 180 × 200 pixels in the portrait format on a plain green background and index 1 to 20 prefixed with a counter. In all, a total of 3040 images were created which make up the database for the study.

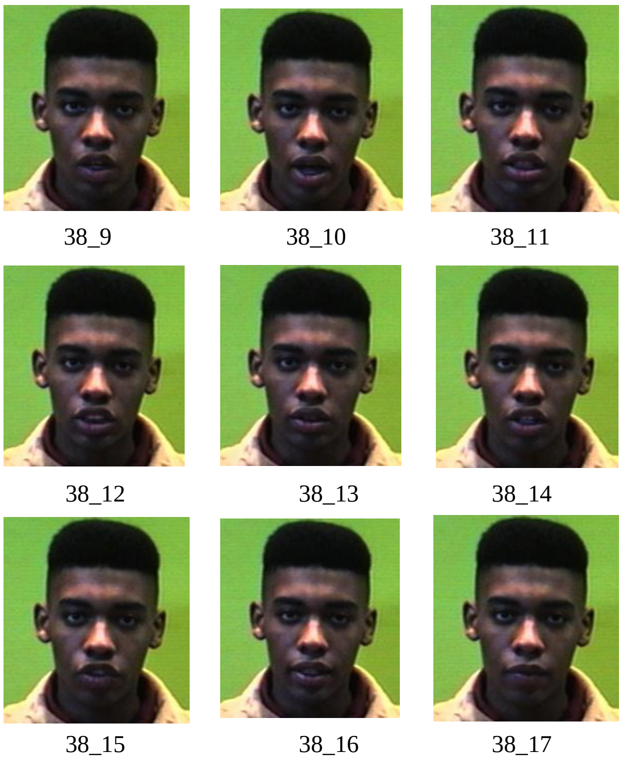

Fig. 1. Sample of face image from Database

B. *Computer Architecture*

With the chosen face dataset, a computer program was written to perform face recognition on a machine with the following specification.

- Lenovo Thinkpad x230
- 2.6 GHz Intel Core i5 (Processor)
- 4GB of RAM (Memory)
- 500 GB (Hard Disk)

having Ubuntu13.04 (64bits) installed on it as the Operating System (OS).

C. *Performance Analysis*

The efficiency of the various standard kernel functions was examined by testing the SVM on the face94 database. Before this performance was carried out, a radon transform feature vector containing 12,160 features with four features for each face image was partitioned into two disjoint set, that is, the trained features and the test features. This partition resulted in 2,432 features for test data and 9,728 features for trained data. That is, 608 and 2432 face images for test and train respectively. Each face feature in the test data is matched against all the face features in the train dataset while the Euclidean distance is taken. Out of the many distance taken, the minimum distance is selected along with it image index to be used for the computation of recognition rate. The table below show the various kernel functions used, their recognition rate as well as time taken in second by each kernel function.

Table 1. Performance of Kernel Function

Kernel Function	Recognition Rate	Time in Seconds
Linear	97.70%	60.14
Quadratic	92.27%	55.32
Polynomial	89.97%	233.37
Radial Basis	94.41%	54.85
Multilayer Perceptron(MLP)	99.84%	55.46

IV. CONCLUSIONS

In this paper, we identified an appropriate kernel function suitable for the face94 database. We applied the SVM as a classifier during the face recognition. We observed that, among the five standard Kernel functions selected, Multilayer perceptron recorded the highest recognition rate (99.84%) when applied to the selected database. Although the linear kernel gives quite a resonalbe recognition rate (97.70%), it will cost as approximately additional 5sec to as compared to the MLP to achieve such recognition rate. However, in the absence of MLP, Linear will be preferred to Radial Basis on the basis of recognition rate since is higher though Radial basis spends less seconds to achieve 94.41% recognition rate.

REFERENCES

[1] A. Samal and P. A. Iyengar. Automatic recognition and analysisof human faces and facial expressions: A survey. PatternRecognition, 25:65–77, 1992.V. N. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, 1998.

[2] D. Valentin, H. Abdi., A. J. O'Toole, and G. W. Cottrell. Connectionist models of face processing: A survey. *Pattern Recognition*, 27:1209–1230, 1994

[3] R. Chellappa, C. L.Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proc. IEEE*, 83:705–741, May 1995

[4] V. N. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, 1998.

[5] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. *Proc. CVPR*, 1997.

[6] M. Pontil and A. Verri. Support vector machines for 3-d object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:637–646, 1998.

[7] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *Int. Journal of Computer Vision*, 14:5–24, 1995.

[8] D. Roobaert, P. Nillius, and J. Eklundh. Comparison of learning approaches to appearance-based 3d object recognition with and without cluttered background. *ACCV2000*, to appear.

[9] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural network approach. *IEEE Trans. Neural Networks*, 8:98–113, 1997.

[10] M. A. Turk and A. P. Pentland. Eigenfaces for recognition. *J. Cognitive Neurosci.*, 3(1):71–86, 1991.

[11] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. "A Practical Guide to Support Vector Classification". Dept of Computer Sci. National Taiwan Uni, Taipei, 106, Taiwan http://www.csie.ntu.edu.tw/~cjlin 2007

