WAVELET SUPPORT VECTOR MACHINE FOR FACE RECOGNITION

Chen-Chia Chuang
Department of Electrical Engineering, National Ilan University, Ilan County, Taiwan, R.O.C.

Ping-Lun Liao
Department of Computer Science and Information Engineering, National Formosa University, Yunlin County, Taiwan, R.O.C.

Jin-Tsong Jeng
Department of Computer Science and Information Engineering, National Formosa University, Yunlin County, Taiwan, R.O.C, tsong@nfu.edu.tw

Follow this and additional works at: https://jmstt.ntou.edu.tw/journal

Part of the Controls and Control Theory Commons

WAVELET SUPPORT VECTOR MACHINE FOR FACE RECOGNITION

Acknowledgements

WAVELET SUPPORT VECTOR MACHINE FOR FACE RECOGNITION

Chen-Chia Chuang[1], Ping-Lun Liao[2], and Jin-Tsong Jeng[2]

ABSTRACT

In this paper a tool system with wavelet support vector machine (WSVM) under dimension reduction for face recognition is proposed. Eigenfaces and fisherfaces are the major methods used to reduce the dimension of face images in the proposed face recognition tool system. At the same time, noise interference image cases, namely Gaussian noise, with no ears and no eyes are also considered in this paper. Finally, the implementation of proposed system is based on object-oriented design software; the dimension reduction and wavelet kernel support vector machine are applied to the proposed tool system that can deal with face recognition problems in different environments.

I. INTRODUCTION

Artificial intelligence has been in rapid development during the past few decades; in this field there are many areas of research such as reasoning, inference systems, and machine learning. At present machine learning is a popular area of artificial intelligence; machine learning is a mathematical theory based on empirical data to design and develop algorithms, and these algorithms allow computers to simulate human learning ability [12]. In general, there are some different architectures of machine learning such as supervised learning, unsupervised learning, and semi-supervised learning. Some of the most used critical techniques in machine learning includes the Bayes network, decision trees, Gaussian processes, the Markov model, support vector machines (SVM), neural networks, sequence analysis, and regression models. At the same time, some applications with machine learning include data mining, voice recognition, image processing, robots, the field of biology, and automatic control.

SVM is one of the popular machine learning methods in the area of artificial intelligence and was proposed by Vapnik [18], and based on the statistical learning theory (SLT). The SLT in the SVM is used to infer the rules of the events from observed nature phenomenon or simulation results. SVM is very suitable in approximating high dimensionality space. SVM can use a quadratic programming (QP) problem that can be guaranteed to find a global extremism solution under fixed hyperparameters, as the SVM models are closely related with classical multilayer perceptron neural networks. Using a kernel function, the SVM is an alternative training method for radial basis function and multi-layer perceptron classifiers. The weights of SVM are found by solving a QP problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training. One of the popular tools in SVM is libsvm [3]. The SVM with libsvm has two major parts, the support vector classification (SVC) and support vector regression (SVR). In this paper, the wavelet kernel is added to SVC and SVR in the libsvm.

Face recognition has attracted tremendous attention in the computer vision community since 1981. Two dimension reduction methods, eigenface [1] and fisherface [13], were independently proposed for extracting features. The eigenface method was proposed by Kirby [9], and it assumed that a human face could be constructed by linear combination of a set of facial basis features which was solved by principle components analysis (PCA) [15]. The fisherface method applied the advantages of linear discriminant analysis (LDA) to maximize the distance between different faces within the projected space. That is, this method improved the correct rate of recognition. Since these techniques were discovered, many more new techniques have been surveyed [6, 10] for facial recognition. In this paper, wavelet kernel support vector machine is applied to the proposed system that can deal with the classification of face recognition problems with different noises under different kernels. At the same time, different noise interference image cases are also considered in the proposed system. The proposed system is implemented by using object-oriented design software. The selection of hyperparameters in the proposed wavelet SVM (WSVM) approach is based on our previous results [4, 7] for the different face recognition problems.
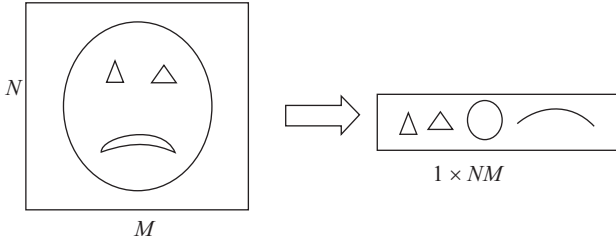
Fig. 1. Basic concept of eigenfaces.

II. MATHEMATICAL PRELIMINARIES FOR FACE RECOGNITION

1. Eigenface Method

Firstly, each $N \times M$ image array is reshaped into an $1 \times NM$ vector as shown in Fig. 1. A new vector is assigned the value x_k, where k is the index of the image. Then, the total scatter matrix is calculated by the following equation:

$$S_r = \sum_{k=1}^{N_t} (x_k - \mu)(x_k - \mu)^T, \tag{1}$$

where N_t = total number of images, T is a transpose of a matrix and

$$\mu = \frac{1}{N} \sum_{k=1}^{N_t} x_k = Mean. \tag{2}$$

The eigenvectors of this matrix are then found using single value decomposition.

$$S_T = UDV^T. \tag{3}$$

Sorting the columns of matrix U corresponding to the largest values in diagonal matrix D, one receives a list of the most significant eigenfaces in descending order. Feature vectors can then be established using the following equation:

$$y_k = Ux_k, \quad where \quad k = 1, 2, ..., N. \tag{4}$$

2. Fisherface Method

The fisherface method is similar to the eigenface method. Unlike the eigenface method [13], the fisherface method determines both the between-class scatter matrix (S_B) and the within-class scatter matrix (S_W). In order to determine S_B and S_W, it uses the x_k values to calculate both the class mean μ_k and the mean of all samples μ. That is,

$$\mu = \frac{1}{N} \sum_{k=1}^{N_t} x_k, \tag{5}$$

$$\mu_k = \frac{1}{N} \sum_{m=1}^{N_k} x_{k_m}, \tag{6}$$

where N_k = number of images in class k, and x_{k_m} = image at index m of class k.

The between-class scatter matrix (S_B) becomes

$$S_B = \sum_{k=1}^{C_f} N_k (\mu_k - \mu)(\mu_k - \mu)^T, \tag{7}$$

where C_f = the number of classes and the within-class scatter matrix (S_W) becomes

$$S_W = \sum_{k=1}^{C_f} \sum_{x \in X_k} (x_k - \mu)(x_k - \mu)^T. \tag{8}$$

The optimal eigenvectors (U_{opt}) can be found by Eq. (9).

$$U_{opt} = \arg\max \frac{\left| U^T S_B U \right|}{\left| U^T S_W U \right|} = [u_1, u_2, ..., u_m]. \tag{9}$$

This equation can then be simplified into a generalized eigenvalue equation as the following:

$$S_B u_i = \lambda_i S_B u_i, \quad i = 1, 2, ..., m. \tag{10}$$

Feature vectors can then be established by using Eq. (11).

$$y_k = U^T x_k, \quad where \quad k = 1, 2, ..., m. \tag{11}$$

3. General Discriminant Analysis Method

The general discriminant analysis (GDA) method starts off with the same steps as the fisherface method. Two matrices are constructed, the between-class scatter matrix S_B,

$$S_B = \sum_{k=1}^{C_f} N_k (\mu_k - \mu)(\mu_k - \mu)^T \tag{12}$$

and the within-class scatter matrix S_W

$$S_W = \sum_{k=1}^{C_f} \sum_{x \in X_k} (x_k - \mu)(x_k - \mu)^T. \tag{13}$$

The optimal eigenvectors (U_{opt}) differ from fisherface method, and are defined by the following equation:

$$U_{opt} = \arg\max \frac{\left| U^T S_B U \right|}{\left| U^T S_B U + U^T S_W U \right|} = [u_1, u_2, ..., u_m]. \tag{14}$$

This equation is then simplified into the following generalized eigenvector equation:

$$S_B u_i = \lambda_i (S_B + S_W) u_i, \quad i = 1, 2, ..., m. \quad (15)$$

Feature vectors can be established by using Eq. (16)

$$y_k = U^T x_k, \quad where \ k = 1, 2, ..., m. \quad (16)$$

4. Gabor Filter

In a Gabor filter, each point is represented by local Gabor filter responses. A 2-D Gabor filter is obtained by modulating a 2-D sine wave with a Gaussian envelope. This paper follows the notation by Hamamoto [5] and Lim *et al.* [11].

$$f(x, y, \theta_k, \lambda) = \exp\left(-\frac{1}{2}\left(\frac{(x\cos\theta_k + y\sin\theta_k)^2}{\sigma_x^2} + \frac{(-x\sin\theta_k + y\cos\theta_k)^2}{\sigma_y^2}\right)\right)$$
$$\cdot \exp\left(\frac{2\pi(x\cos\theta_k + y\sin\theta_k)^2}{\lambda}\right), \quad (17)$$

where σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y-dimensions, respectively. λ and θ_k are the wavelength and orientation, respectively. The spread of the Gaussian envelope is defined using the wavelength λ. θ_k is defined by

$$\theta_k = \frac{\pi}{n}(k-1), \quad k = 1, 2, ..., n, \quad (18)$$

where n denotes the number of orientations. For the sampling point (X, Y), the Gabor filter response, denoted as $g(.)$, is defined by the following equation:

$$g(X, Y, \theta_k, \lambda) = \sum_{x=-X}^{N-X-1} \sum_{y=-Y}^{N-Y-1} I(X+x, Y+y)f(x, y, \theta_k, \lambda), \quad (19)$$

where $I(x, y)$ denotes an $N \times N$ grayscale image.

III. PROPOSED WAVELET SVM TOOL SYSTEM

1. Support Vector Classification

The aim of support vector classification is to devise a computational efficient way of learning 'good' separating hyperplane in a high dimensional feature space. SVM has introduced the concept of maximum margin to solve this problem. In the SVM theory [17, 18], the separable cases are defined as

$$y(x) = \text{sgn}[w^T x + b], \quad (20)$$

and

$$\begin{cases} w^T x_k + b \geq 1, & if \ y_k = +1, \\ w^T x_k + b \leq 1, & if \ y_k = -1. \end{cases} \quad (21)$$

Eqs. (20) and (21) can be simplified to

$$y_k(w^T x_k + b) \geq 1, \quad k = 1, ..., N_s. \quad (22)$$

N_s is the total number of sampling data. The convex optimization theory is trying to solve the SVC problem. The steps of SVC under optimization theory are the following:

Step 1: The problem in primal space is

$$Minimize \quad \frac{1}{2} w^T w$$

$$subject \quad to \quad y_k(w^T x_k + b) \geq 1, \quad k = 1, ... N_s \quad (23)$$

Step 2: The Lagranian form

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{k=1}^{N_s} \alpha_k \left[y_k(w^T x_k + b) - 1 \right], \quad (24)$$

where α is the non-negative Lagrange multiplier. Using the Lagrange method, Eq. (24) becomes

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{k=1}^{N_s} \alpha_k y_k x_k, \\ \dfrac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_{k=1}^{N_s} \alpha_k y_k = 0. \end{cases} \quad (25)$$

Replace Eq. (24) with Eq. (25),

$$L(\alpha) = \sum_{k=1}^{N_s} \alpha_k - \frac{1}{2} \sum_{k=1}^{N_s} \sum_{i=1}^{N_s} y_k y_i x_k^T x_i \alpha_k \alpha_i \quad (26)$$

can be obtained.

Step 3: Now, the problem becomes a dual problem,

$$Maximize \quad L(\alpha) = \sum_{k=1}^{N_s} \alpha_k - \frac{1}{2} \sum_{k=1}^{N_s} \sum_{i=1}^{N_s} y_k y_i x_k^T x_i \alpha_k \alpha_i$$

$$Subject \quad to \quad \sum_{k=1}^{N_s} \alpha_k y_k = 0, \quad (27)$$

where $\alpha_k \geq 0$ for $k = 1, ..., N_s$.

Step 4: Assume there is an optimal solution: $\vec{\alpha} = (\vec{\alpha}_1, \vec{\alpha}_2, ..., \vec{\alpha}_{N_s})^T$ for Eq. (27). Substitute $\vec{\alpha}$ for Eq. (25). The optimal weight vector becomes

$$\vec{w} = \sum_{k=1}^{N_s} \vec{\alpha}_k y_k x_k. \tag{28}$$

Using the Karush-Kuhn-Tucker (KKT) theorem, the optimal solution for $\vec{\alpha}$ satisfies

$$\vec{\alpha}_k [y_k (w^T x_k + b) - 1] = 0, \quad k = 1, ..., N_s. \tag{29}$$

The decision function is

$$f(x) = \operatorname{sgn}(\vec{w}^T x + \vec{b}) = \operatorname{sgn}\left(\sum_{k-1}^{N_s} \vec{\alpha}_k y_k x_k^T x + \vec{b} \right). \tag{30}$$

2. Support Vector Regression

Dataset for the SVR can be represented as the following

$$D = \left\{ (x_1, y_1), ..., (x_k, y_k), ..., (x_{N_s}, y_{N_s}) \right\}, x_k \in R^n, \ y_k \in R. \tag{31}$$

And its nonlinear function

$$f(x) = \langle \omega, \varphi(x) \rangle + b, \tag{32}$$

where $\varphi(\cdot): R^n \to R^{n^h}$ is the nonlinear function and maps input space into high dimensional space, b is the bias term, and n^h is the infinite dimensional space. The optimization problem

$$\min \tfrac{1}{2} \| \omega \|^2 + C \sum_{k=1}^{N_s} (\xi_k + \xi_k^*),$$

$$\text{s.t.} \begin{cases} y_k - \langle \omega, \varphi(x_k) \rangle - b \le \varepsilon + \xi_k, \\ \langle \omega, \varphi(x_k) \rangle + b - y_k \le \varepsilon + \xi_k^*, \\ \xi_k, \xi_k^* \ge 0, \end{cases} \tag{33}$$

which has ε-insensitive loss function that is used to check the distance between the regression equation and the training sample.

$$\left| y - f(x, \omega) \right|_\varepsilon = \begin{cases} 0, & if \left| y - f(x, \omega) \right| \le \varepsilon, \\ \left| y - f(x, \omega) \right| - \varepsilon, & otherwise, \end{cases}$$

$$\left| y - f(x, \omega) \right| - \varepsilon = \xi,$$

$$\left| y - f(x, \omega) \right| - \varepsilon = \xi^*. \tag{34}$$

Moreover, ε is the precision of the approximate, ξ and ξ^* are slack variables which measure the deviation of training samples outside the ε-insensitive area. C expresses the tolerate degree of the bias quality and is a regularization constant.

Eq. (35) is a Lagrange function for the SVR as follows

$$L = \frac{1}{2} \| \omega \|^2 + C \sum_{k=1}^{N_s} (\xi_k + \xi_k^*)$$

$$- \sum_{k=1}^{N_s} \alpha_k \left(\varepsilon + \xi_k - y_k + \langle \omega, \varphi(x_k) \rangle + b \right)$$

$$- \sum_{k=1}^{N_s} \alpha_k^* \left(\varepsilon + \xi_k^* - y_k - \langle \omega, \varphi(x_k) \rangle - b \right)$$

$$- \sum_{k=1}^{N_s} (\eta_k \xi_k + \eta_k^* \xi_k^*), \tag{35}$$

where α, α^*, η and $\eta^* \ge 0$ are Lagrange multipliers. The corresponding dual is found by differentiating with respect to ω, b, ξ and ξ^*, the following:

$$\begin{cases} \dfrac{\partial L}{\partial \omega} = 0 \quad \to \quad \omega = \sum_{k=1}^{N_s} (\alpha_k - \alpha_k^*) \varphi(x_k), \\[2mm] \dfrac{\partial L}{\partial b} = 0 \quad \to \quad \sum_{k=1}^{N_s} (\alpha_k^* - \alpha_k) = 0, \\[2mm] \dfrac{\partial L}{\partial \xi_k} = 0 \quad \to \quad C - \alpha_k - \eta_k = 0, \\[2mm] \dfrac{\partial L}{\partial \xi_k^*} = 0 \quad \to \quad C - \alpha_k^* - \eta_k^* = 0. \end{cases} \tag{36}$$

Hence, the optimization problem in Eq. (33) can be transformed into a dual problem by the following:

$$\max_{\alpha, a^*} Q = -\frac{1}{2} \sum_{k,l=1}^{N_s} (\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*) \langle \varphi(x_k), \varphi(x_l) \rangle$$

$$- \varepsilon \sum_{k=1}^{N_s} (\alpha_k + \alpha_k^*) + \sum_{k=1}^{N_s} y_k (\alpha_k - \alpha_k^*), \tag{37}$$

$$\text{s.t.} \begin{cases} \sum_{k=1}^{N_s} (\alpha_k - \alpha_k^*) = 0, \\ \alpha_k, \alpha_k^* \in [0, C]. \end{cases}$$

According to the KKT condition [4], the variable b can be calculated. For the SVR, this can be represented as Eqs. (38) and (39),

$$\begin{cases} \alpha_k \left[\varepsilon + \xi_k - y_k + \langle \omega, \varphi(x_k) \rangle + b \right] = 0, \\ \alpha_k^* \left[\varepsilon + \xi_k^* + y_k - \langle \omega, \varphi(x_k) \rangle - b \right] = 0, \end{cases} \tag{38}$$

$$\begin{cases} \eta_k \xi_k = (C - \alpha_k)\xi_k = 0, \\ \eta_k^* \xi_k^* = (C - \alpha_k^*)\xi_k^* = 0. \end{cases} \quad (39)$$

In Eq. (38), as $|f(x_k) - y_k| \geq \varepsilon$, the Lagrange variable cannot be zero. From Eqs. (38) and (39), the vectors map into $\alpha_k^{(*)} \in (0, C)$ and $\xi_k^{(*)} = 0$. Hence, b can be calculated by the following:

$$b = y_k - \langle \omega, \varphi(x_k) \rangle - \varepsilon \quad for\ \alpha_k \in (0, C),$$

$$b = y_k - \langle \omega, \varphi(x_k) \rangle + \varepsilon \quad for\ \alpha_k^* \in (0, C). \quad (40)$$

The function evaluation of the SVM nonlinear function can be represented as

$$f(x) = \sum_{k=1}^{N_s} (\alpha_k - \alpha_k^*) \langle \varphi(x), \varphi(x_k) \rangle + b. \quad (41)$$

In Eq. (41), the inner product of the basis function $\langle \varphi(x), \varphi(x_k) \rangle$ can be replaced with the kernel function $K(x, x_k)$

$$f(x) = \sum_{k=1}^{\#_{SV}} (\alpha_k - \alpha_k^*) K(x, x_k) + b,$$

$$s.t.\ 0 \leq \alpha_k^* \leq C, \quad 0 \leq \alpha_k \leq C. \quad (42)$$

Only some of $(\alpha_k - \alpha_k^*)$ are not zeros and the corresponding vectors x_k are called support vectors (SVs). That is, $\#_{SV} \leq N_s$, where $\#_{SV}$ is the number of SVs for the training data. The kernel function is defined by Eq. (43)

$$K(x, x_k) = \sum_{j=1}^{m_s} g_j(x) g_j(x_k), \quad (43)$$

where m_s is the number of dimensions for x. There are four kernel functions $K(\cdot, \cdot)$ namely, linear, sigmoid, radial basis function, and polynomial in the libsvm for SVM. The basic idea of a kernel method [16] is mapping input data into another space. For example, $\Phi: R^2 \rightarrow R^3$ is the following equation

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) = \left(x_1^2, \sqrt{2} x_1 x_2, x_2^2 \right). \quad (44)$$

$$\ker(f) = \langle f(x_1) \bullet f(x_2) \rangle, \quad (x_1, x_2) \in X \times X. \quad (45)$$

$\ker(f)$ is the definition of nonlinear kernel in SVM. In this paper, a wavelet kernel is added to SVC and SVR in the libsvm. In general, a wavelet is a small wave which has amplitude that starts out at zero, increases, and then decreases back to zero [2, 19]. Each wavelet in Eq. (46) is derived from a zero-mean 'mother' function ψ through two linear transformations: (1) dilatation by a scale parameter, and (2) translation by 'u'.
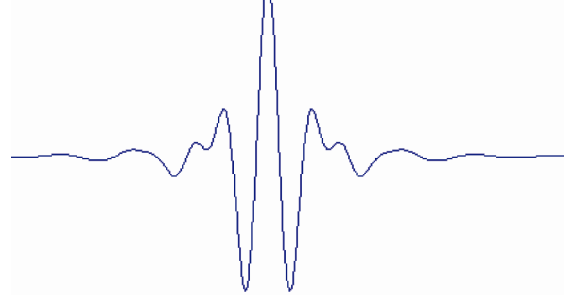

Fig. 2. Meyer wavelet.

These parameters determine the width of the window and hence define the resolution of the transform, as illustrated in Fig. 2.

$$\psi_{s,u}(x) = \frac{1}{\sqrt{s}} \cdot \psi(\frac{x - u}{s}) \quad (46)$$

Eqs. (27) and (30) can be rewrite in kernel form

$$Maximize \quad L(\alpha) = \sum_{k=1}^{N_s} \alpha_k - \frac{1}{2} \sum_{k=1}^{N_s} \sum_{i=1}^{N_s} y_k y_i \alpha_k \alpha_i K(x_k, x_i)$$

$$Subject \quad to \quad \sum_{k=1}^{N_s} \alpha_k y_k = 0, \quad (47)$$

and

$$f(x) = \text{sgn}(\vec{w}^T x + \vec{b}) = \text{sgn}\left(\sum_{k-1}^{N_s} \vec{\alpha}_k y_k K(x, x_k) + \vec{b} \right). \quad (48)$$

According to [19], the wavelet kernel is

$$K(x, x') = \prod_{i=1}^{m_s} \left(\cos\left(1.75 \times \frac{(x_i - x_i')}{a} \right) \exp(-\frac{\|x_i - x_i'\|^2}{2a^2}) \right). \quad (49)$$

The decision function for classification of the proposed tool system is

$$f(x) = \text{sgn}\left(\sum_{i=1}^{N_s} \alpha_i y_i \prod_{j=1}^{m_s} h\left(\frac{x^j - x_i^j}{a_i} \right) + b \right). \quad (50)$$

where $h(x) = \cos(1.75x) \exp(-\frac{x^2}{2})$.

The estimate function for regression of the proposed tool system is

$$f(x) = \sum_{i=1}^{\#sv} (\alpha_i - \alpha_i^*) \prod_{j=1}^{m_s} h\left(\frac{x^j - x_i^j}{a_i} \right) + b. \quad (51)$$
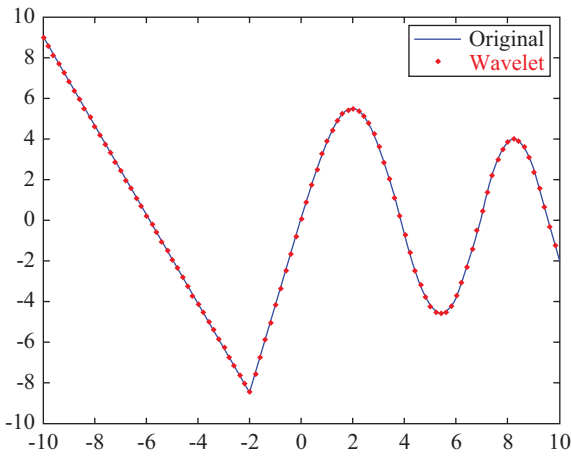
Fig. 3. Single-variable function approximation with the proposed system.

IV. PROPOSED TOOL FOR FACE RECOGNITION

A smart client bridges the gap between web applications and desktop applications. We can enjoy the superb desktop user interface as well as the advantages of web pages (such as networking capability, remote access to information). A smart client utilizes the advantages of a thin client (no installation, automatic update) with rich client (high-performance, high productivity). In addition to Microsoft's .NET platform, Adobe's Flash, Sun's Java Applet, as well as Webstart and other platforms can be used to develop smart client applications [8]. Using the .NET framework to develop a smart client is more convenient than other tools. In this paper, .NET framework 3.5 was chosen for development of the proposed tool system for smart client with C#.

Firstly, the two cases for testing the proposed WSVM tool in this paper are regression problems. Regression cases includes single-variable function and two-variables function for use with the proposed WSVM tool system. The mean square error (MSE) is used to measure the performance of the results.

Case 1: Regression of a single-variable [19] function
The single-variable function is defined as

$$f(x) = \begin{cases} -2.186x - 12.864, & -10 \le x < -2 \\ 4.246x, & -2 \le x < 0 \\ 10e^{-0.05x - 0.5} \cdot \sin((0.03x + 0.7)x), & 0 \le x \le 10. \end{cases} \tag{52}$$

In this case, the MSE value is 9.97892026230995e-05. Fig. 3 shows the results of simulation, where the red dot is the prediction of wavelet SVM and the blue solid line is the original curve of the function. The parameters C and a are 10 and 0.001, respectively.

Case 2: Regression of two-variables [19] function
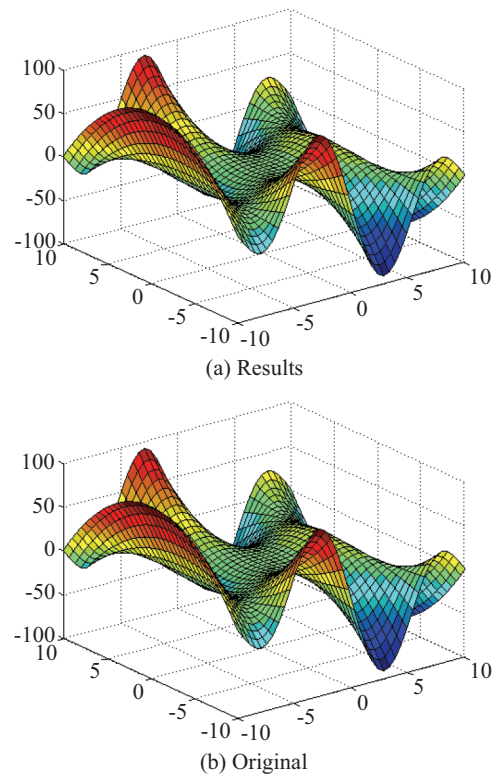The two-variables function is defined as

(a) Results

(b) Original

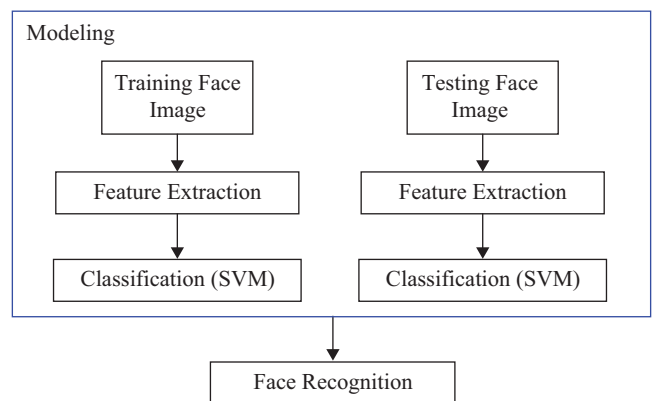**Fig. 4. Two-variables function approximation with the proposed system.**

Fig. 5. The flow chart of the proposed face recognition system.

$$f(x) = (x_1^2 - x_1^2)\sin(0.5x_1) \tag{53}$$

over the domain [-10, 10] by [-10, 100]. In this case, the MSE value is 0.000453. Fig. 4 shows the results diagram, where (a) is the prediction of wavelet SVM, and (b) is the original curve of the function. The parameters C and a are 500 and 3, respectively. The two testing regression functions above show that the proposed tool system has better performance in regression.

Secondly, for the face recognition problem, it is necessary to model the system into training and testing phases. Fig. 5 is the flow chart of the proposed method. In the training phase,
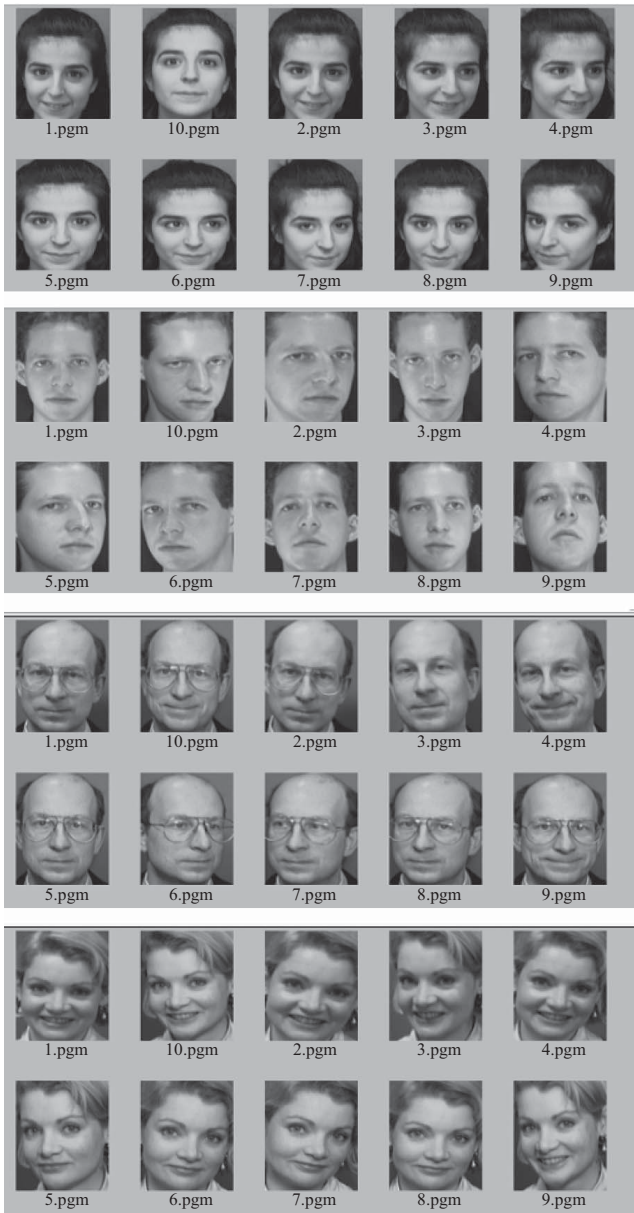
Fig. 6. **ORL face database image.**

Fig. 7. **Noise interference images with mean= 0.1 and variance= 0.01.**

Fig. 8. **Noise interference images with no ears information.**

Fig. 9. **Noise interference images with no eyes information.**

the system learns the knowledge of the image for face recognition. In the testing phase, the system uses this knowledge to validate the accuracy rate. If the accuracy rate is acceptable, the model is used for recognition. The images of the simulation are from the ORL Face Database [14]. There are 40 people and 10 pictures for each person. The image is 92 * 112 pixel 8-bit gray level and a **P**ortable **G**ray **M**ap (*.pgm) format (See Fig. 6). 200 images were used for the training phase and another 200 images were used for the testing phase.

The proposed method used Gaussian white noise to generate noise in this paper. The mean= 0.1 and variance= 0.01 of noise interference images are shown in Fig. 7. Noise interference facial images with no ears and no eyes information are also considered and shown in Figs. 8 and 9, respectively.
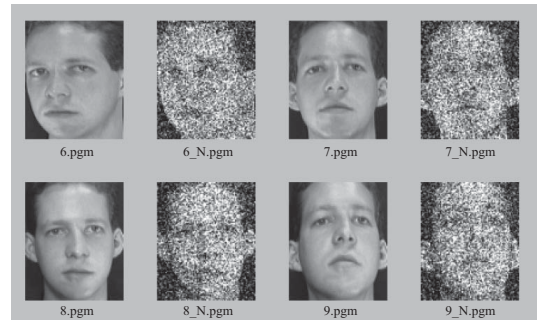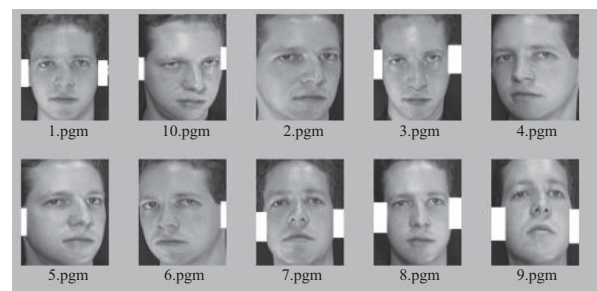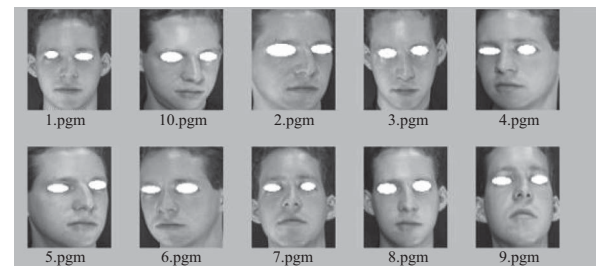
V. RESULTS

In this paper, the proposed tool systems used a .NET framework to develop smart client tools. That is, .NET framework 3.5 was chosen as the development tool for a smart client with C#. Because the feature extraction methods were used to reduce the dimension of the images, two dimension sizes were chosen for the simulation: 200 and 400. Also considered in this paper were different environments, namely noise interference images, and face images with no ears, and face images with no eyes. We also compared the different kernels: linear, polynomial, radial basis function (RBF), and wavelet kernel in SVM. Tables 1~4 list the accuracy of different kernels in SVM under different reduced dimension approaches and different noise interference conditions at the 200 dimension size. Tables 5~8 list the accuracy of different kernels in SVM under different reduced dimension approaches and different noise interference conditions at the 400 dimension size.

Table 1. No noise interference case for 200 dimension size.

Kenel\Feature	Eigenfaces	Fishfaces	GDA	Gabor
Linear	93.50%	100.00%	97.00%	87.00%
Polynomial	93.50%	100.00%	97.00%	87.00%
RBF	92.00%	99.50%	90.00%	82.00%
Wavelet	94.00%	100.00%	97.00%	87.00%

Table 2. Gaussian white noise (0.1 mean with 0.01 variance) for 200 dimension size.

Kernel\Feature	Eigenfaces	Fishfaces	GDA	Gabor
Linear	89.00%	100.00%	96.50%	57.50%
Polynomial	89.50%	100.00%	95.50%	57.50%
RBF	85.50%	100.00%	90.00%	66.00%
Wavelet	89.00%	100.00%	97.00%	57.50%

Table 3. No ears case for 200 dimension size.

Kernel\Feature	Eigenfaces	Fishfaces	GDA	Gabor
Linear	95.50%	100.00%	88.00%	81.50%
Polynomial	95.50%	100.00%	88.00%	82.50%
RBF	89.50%	98.50%	83.50%	75.00%
Wavelet	95.50%	100.00%	89.00%	81.50%

Table 4. No eyes case for 200 dimension size.

Kernel\Feature	Eigenfaces	Fishfaces	GDA	Gabor
Linear	92.00%	100.00%	81.00%	81.50%
Polynomial	92.00%	100.00%	81.50%	82.00%
RBF	83.50%	96.50%	71.00%	81.00%
Wavelet	92.00%	100.00%	82.50%	82.00%

Table 5. No noise interference case for 400 dimension size.

Kernel\Feature	Eigenfaces	Fishfaces	GDA	Gabor
Linear	93.50%	100.00%	77.00%	87.00%
Polynomial	93.50%	100.00%	77.00%	87.00%
RBF	92.00%	97.50%	75.00%	82.00%
Wavelet	93.50%	100.00%	77.00%	87.00%

Table 6. Gaussian white noise (0.1 mean with 0.1 variance) for 400 dimension size.

Kernel\Feature	Eigenfaces	Fishfaces	GDA	Gabor
Linear	82.50%	99.50%	82.00%	31.00%
Polynomial	82.50%	99.50%	82.00%	31.00%
RBF	77.50%	93.50%	77.50%	50.00%
Wavelet	82.00%	99.50%	82.50%	51.00%

Table 7. No ears case for 400 dimension size.

Kernel\Feature	Eigenfaces	Fishfaces	GDA	Gabor
Linear	95.50%	100.00%	67.00%	81.50%
Polynomial	95.50%	100.00%	67.00%	82.50%
RBF	90.50%	97.00%	63.00%	78.50%
Wavelet	95.00%	99.50%	67.00%	82.50%

Table 8. No eyes case for 400 dimension size.

Kernel\Feature	Eigenfaces	Fishfaces	GDA	Gabor
Linear	92.00%	99.50%	47.00%	81.50%
Polynomial	92.00%	99.50%	47.00%	82.00%
RBF	88.50%	89.50%	41.00%	81.00%
Wavelet	92.00%	99.50%	47.00%	81.50%

From the above results, the eigenface and the fisherface methods under the proposed tool system have better performance in the face recognition problem, as eigenface and fisherface methods exhibit noise resistance. Hence, the proposed tool system based on eigenface and fisherface methods can deal with different cases that have better performance on face recognition problems at different noise levels.

VI. CONCLUSION

In this paper, a tool system with wavelet SVM for face recognition was developed. Eigenfaces and fisherfaces were applied to reduce the dimensions of an image, as eigenface and fisherface methods exhibit noise resistance. Hence, the proposed tool based on eigenface and fisherface methods can deal with different cases that have better performance on face recognition problems at different noise levels. The implementation of the proposed tool systems are based on object-oriented design software and wavelet kernel SVM, and can deal with face recognition problems at the different conditions in this paper.

REFERENCES

1. Brunelli, R. and Poggio, T., "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 10, pp. 1042-1052 (1993).
2. Burrus, C. S., Gopinath, R. A., and Guo, H., *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall, New York (1997).
3. Chang, C. C. and Lin, C. J., LIBSVM: A Library for Support Vector Machines, Online from http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
4. Fu, Y. Y., Wu, C. J., Ko, C. N., and Jeng, J. T., "Radial basis function networks with hybrid learning for system identification with outliers," *Applied Soft Computing*, Vol. 11, No. 3, pp. 3083-3092 (2011).
5. Hamamoto, Y., "A gabor filter-based method for fingerprint identification," in: Jain, L., Halici, U., Hayashi, I., Lee, S., and Tsutsui, S. (Eds.), *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, CRC Press, Boca Raton, FL, pp. 137-151 (1999).
6. Jafri, R. and Arabnia, R. J., "A survey of face recognition techniques," *Journal of Information Processing Systems*, Vol. 5, No. 2, pp. 41-68 (2009).
7. Jeng, J. T., "Hybrid approach of selecting hyper-parameters of support

vector machine for regression," *IEEE Transactions on Systems, Man, and Cybernatics Part B: Cybernetics*, Vol. 36, No. 3, pp. 699-709 (2006).

8. Jeng, J. T., Liao, P. L., and Chuang, C. C., "Intelligent information-integrated systems for smart home," *International Journal of Intelligent Systems Science and Technology*, Vol. 1, No. 1, pp. 41-46 (2009).

9. Kirby, M. and Sirovich, L., "Application of the karhunen-loeve procedure for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, pp. 103-108 (1990).

10. Li, S. Z. and Jain, A. K., *Handbook of Face Recognition*, Springer, Berlin (2011).

11. Lim, R., Reinders, M. J. T., and Thiang, "Facial landmark detection using a gabor filter representation and a genetic search algorithm," *Proceeding of ASCI Conference* (2000).

12. Mitchell, T., *Machine Learning*, McGraw-Hill, New York (1997).

13. Moghaddam, B. and Pentland, A., "Face recognition using view-based and modular eigenspaces," *Automatic Systems for the Identification and Inspection of Humans, SPIE,* Vol. 2277, No. 12 (1994).

14. ORL Face Database from http://www.uk.research.att.com/facedatabase.html.

15. Safari, M., Harandi, M. T., and Araabi, B. N., "A SVM-based method for face recognition using a wavelet PCA representation of faces," *International Conference on Image Processing (ICIP)*, pp. 853-856 (2004).

16. Schlkopf, B. and Smola, A. J., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, London (2001).

17. Taylor, J. S. and Cristianini, N., *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, London (2000).

18. Vapnik, V., *The Nature of Statistical Learning Theory*, Springer, Berlin (1995).

19. Zhang, Q. H. and Benveniste, A., "Wavelet networks," *IEEE Transactions on Neural Networks*, Vol. 3, pp. 889-898 (1992).