

Research Article

Support Vector Machine for Behavior-Based Driver Identification System

Huihuan Qian,^{1,2} Yongsheng Ou,¹ Xinyu Wu,¹ Xiaoning Meng,² and Yangsheng Xu^{1,2}

¹Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

²Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong

Correspondence should be addressed to Yongsheng Ou, yoo205@lehigh.edu

Received 1 November 2009; Accepted 5 March 2010

Academic Editor: Zeng-Guang Hou

Copyright © 2010 Huihuan Qian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present an intelligent driver identification system to handle vehicle theft based on modeling dynamic human behaviors. We propose to recognize illegitimate drivers through their driving behaviors. Since human driving behaviors belong to a dynamic biometrical feature which is complex and difficult to imitate compared with static features such as passwords and fingerprints, we find that this novel idea of utilizing human dynamic features for enhanced security application is more effective. In this paper, we first describe our experimental platform for collecting and modeling human driving behaviors. Then we compare fast Fourier transform (FFT), principal component analysis (PCA), and independent component analysis (ICA) for data preprocessing. Using machine learning method of support vector machine (SVM), we derive the individual driving behavior model and we then demonstrate the procedure for recognizing different drivers by analyzing the corresponding models. The experimental results of learning algorithms and evaluation are described.

1. Introduction

Automobiles are by now indispensable to our personal lives, as well as to the activities of business, public services, and even national security, but the problem of car thefts is a reality and it threatens the automobile security seriously. According to the National Insurance Crime Bureau, a vehicle is stolen every 25 seconds in the U.S.A. Each year, over 1.2 million vehicles are stolen across the country, causing the loss of 8 billion US dollars. Therefore, the work on vehicle security is significant.

Not surprisingly, many classical intelligent technologies are already well established within the automotive industry for vehicle security. GM has developed the *OnStar* system [1], which can supply the vehicle information to the infrastructure once the vehicle owner reported for a theft. However, it has no solution to how to detect the theft by vehicle automatically. There are a large number of vehicle companies which have commercialized their intelligent electronic car keys [2], however, the level of “intelligence” is only limited to activating the vehicle remotely. The deeper research for intelligent recognition of a vehicle theft will boost the vehicle intelligence significantly.

In the past decade, a number of groups have participated in the research of intelligent vehicles, which have led to projects including ARGO [3], ROVA [4], and MOSFET [5]. These vehicles mainly apply machine vision techniques to perform road-condition analysis and assist people in driving.

In recent years, the paradigm of learning human behaviors has attracted considerable amount of attention. It is difficult to describe the desired instructions into specific and proper code statements. In the past decade, several researchers have proposed various experimental designs and applications [6–8]. Significant research towards learning skills directly from humans on intelligent vehicles has been conducted primarily by the Navlab group at the CMU [9] and our group at the CUHK [10, 11].

Support vector machine (SVM) has recently become popular in the machine learning. SVM is a new learning-by-example paradigm spanning a broad range of classification, regression, and density estimation problems. This systematic approach motivated by statistical learning theory combines ideas from various scientific branches such as mathematical programming, exploiting the quadratic programming for convex optimization, functional analysis,

indicating adequate methods for kernel representations, and machine learning theory, exploring the large maximum classifiers concept [12]. It was first introduced by Vapnik and co-workers and is described in more detail in [13, 14]. The roots of this approach, the so-called support vector (SV) methods of constructing the optimal separating hyperplane for pattern recognition, were already presented and had been used in machine learning in [15]. The SV technique was generalized for nonlinear separating surfaces in [16], and was further extended for constructing decision rules in the nonseparable case [17]. The training task involves optimization of a convex cost function conveying to a technique without local minima.

SVM has been applied to many areas, such as pattern recognition, regression, equalization [18, 19]. It is adopted in applications such as dynamic robot control [20], space robot control [21], image classification [22], human dynamic gait recognition [23], and so on.

In this paper, we focus on the research of utilizing dynamic human behavior models for vehicle security (preventing vehicles from being stolen) application. A methodology of modeling dynamic human behaviors is proposed. By learning from driving performances, the intelligent classifier can be embedded into an IC-based car key, through which the vehicle security system can identify valid drivers based on the ways the vehicle are driven and the drivers behave. When an illegitimate driver come to use the car and the demonstrated driving behaviors do not match the specified model, the car will be enabled to automatically stop running and deliver alarm signals accordingly. In [24, 25], we constructed a steering-by-wire vehicle and its steering interphase, which will be able to work as an on-road test platform for vehicle security.

We highlight the following aspects of our system in this paper: First, live biometrical features in dynamic human behaviors are adopted in the system, which brings the enhanced security to the proposed security system. Second, since we collect the signals directly from human driving controls, which include steering, acceleration and braking, we do not utilize other car dynamics and environmental variables such as the car's yaw angle with respect to the road, lateral offset to the road's center, and the road curvature. Therefore, no complicated sensor is required, which brings to the system robust and efficient performance in realtime. Third, the intelligent security system is easy to install on a normal vehicle by adding on functional modules. No complicated requirements means little space and time needed for system installation and drivers are likewise not distracted by the addition of the in-car system. Finally, we develop a methodology to capture and analyze the characteristics of human behaviors into computational representations. It is easily scalable for other applications.

The paper is organized as follows. Firstly, technical descriptions of the experimental hardware and software platforms are presented. In Section 3, we then extract features from the data collected from the experimental platform. Thereafter, we utilize SVM to classify the features of human driving behavior data in Section 4. Section 5 studies and

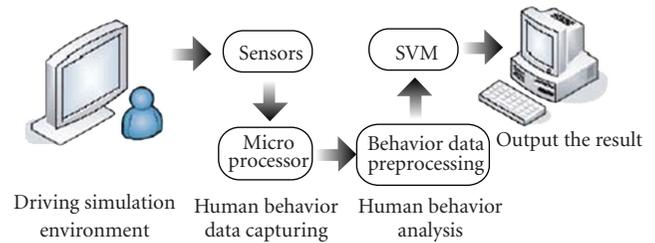


FIGURE 1: Architecture of the experimental system.

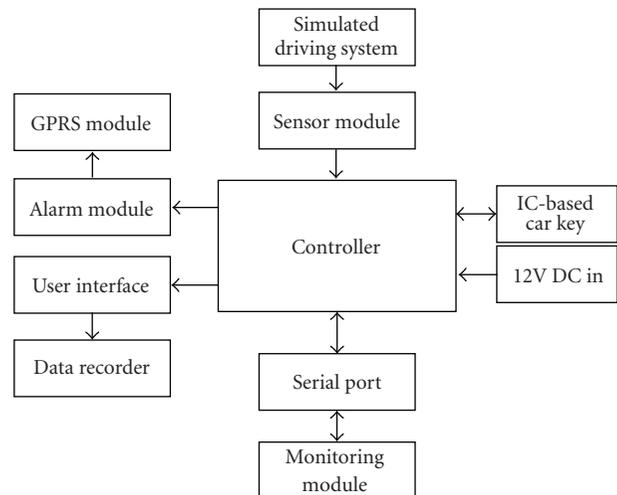


FIGURE 2: Architecture of the system hardware implementation.

analyzes the experimental results. Section 6 concludes the paper.

2. Experimental Platform and Data Collection

2.1. Overview. In this section, the technical descriptions of the implemented hardware and software platforms are presented. Figure 1 shows the architecture of the experimental intelligent vehicle security system. We design an experimental platform which consists of three parts, (1) a real-time graphic simulator offering full controls including steering as well as the brake and acceleration pedals, (2) a sensory system with a processor circuit board to capture human driving behavior data, and (3) an analysis system to model and identify human behavior. While the human subject is “driving” on the driving simulator in part 1, the signals of his/her driving performance are captured by part 2 and then transferred to part 3 for modeling processes.

Figure 2 shows the hardware architecture of the proposed intelligent vehicle security system. The sensory module is implemented in the vehicle (we use simulated driving system instead) to measure the values of steering, braking, and acceleration. The data is sent to the monitoring module in realtime with the driver models embedded in the IC-based car key. Then the recognition result from the monitoring module is sent to user interface in car and to the driving data recorder as well. If the illegitimate driver is detected,

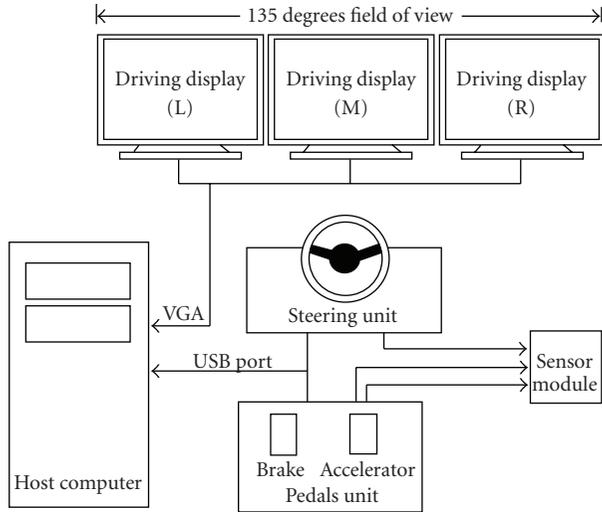


FIGURE 3: Diagram of the simulated driving system.

the alarm module will be active and the alarm message can be delivered through the wireless communication network.

2.2. Driving Simulation Subsystem. Figure 3 shows the diagram of proposed driving simulation subsystem. In this system, a simulated driving environment which bears substantial resemblance to a comparable real driving task is developed. Although some aspects of a real driving task cannot be modeled well in a simulated driving environment, including driving control reality variable road conditions, and, we choose a simulated driving task because it embodies the qualities which meet that criteria for comparing and identifying individual driving behavior. Moreover, the focus of this paper is the analysis of human behaviors itself.

In the simulation subsystem, we apply one PC to offer the simulated driving environment for the driver, including rendered 3-D graphics display as well as realistic controls, which are the steering wheel and pedals for acceleration and brake. We adopt a set of commercial racing game controllers from Logitech (in Figures 4(a) and 4(b)) and the racing game named "Need for Speed: Underground" from Electronic Arts. The racing game provides the features of 3-D road scenes with dynamic traffic stream (in Figure 4(c)). Human subject uses the front view and game controller to drive in the simulated environment, as if sitting in the driving seat in a real car. In this experiment, the host PC used to run the racing game is based on Pentium 4 2.0 G and 1024 MB RAM, the graphics card is Matrox Parhelia 128 MB which supports 3 monitors at a time for wide angle display.

2.3. Data Sensing and Capturing Subsystem. In the data capturing subsystem, a processor circuit board (in Figure 4(d)) is utilized to sense and gather the individual driving behavior data from the driving environment simulation subsystem. With the driving control sensing device, 3 analog signals are gathered and then processed by an A/D converter at the sampling time of 100 ms and then the digitized values are

sent to the microcomputer (ATmega8535L). The received data $X(t)$ can be represented by

$$X(t) \triangleq \{a(t), b(t), c(t)\}, \quad (1)$$

where $a(t)$ represents the normalized acceleration value, $b(t)$ represents the normalized braking value, and $c(t)$ represents the normalized steering value.

The $X(t)$ is a time sequence of 3-Dimensional data (shown in Figure 5) and is transferred to the human behavior analysis subsystem through the RS232 port for further processing.

2.4. Data Analysis Subsystem. In the human behavior analysis subsystem, the methods introduced in the following sections are applied to the retrieved human behavior data. For our goal in identifying the drivers from their driving skills, the human behavior model library of each driver is generated from the corresponding behavior data input. Once the models are ready, we implement them as the classifier in the system in response to the real-time individual driving performance.

Before modeling human driving behaviors, we apply data preprocessing methods towards data collected from the previous subsystem. Fast Fourier transform (FFT), principal component analysis (PCA), and independent component analysis (ICA) are investigated in this paper. The output of this data preprocessing module is for the support vector machine (SVM) modeling and evaluation. These methodologies are presented as follows.

3. Feature Extraction

In this section, we apply data preprocessing methods towards data collected from the aforementioned experimental platform. It is necessary and important to apply data reduction and feature selection in data preprocessing for human behaviors modeling because failures in feature selection reduces the efficiency of the system performance significantly, even bad feature selection causes the failure of whole recognition procedures. Among several feature extraction methods, fast Fourier transform (FFT), principal component analysis (PCA), and independent component analysis (ICA) are investigated in this paper.

3.1. Fast Fourier Transform. To determine the extent of preprocessing human behaviors, we consider factors such as the existence of a preprocessing algorithm, its necessity, its complexity, and its generality. We select fast Fourier transform. In fact, if we have a function given by $x(t)$ whose Fourier transform is $X(f)$, when we shift $x(t)$ by a constant time, T , that is, $x(t - T)$, its Fourier transformation is

$$X(f)e^{-j2\pi fT}, \quad (2)$$

that is, time shifting affects phase only; the magnitude remains constant throughout.

Although the Fourier transform does not explicitly show the time localization of frequency components, the time



FIGURE 4: (a) Steering wheel; (b) Acceleration and brake pedals; (c) Simulated driving scene; (d) Processor circuit board.

localization can be presented by suitably prewindowing the signal in time domain [26]. Accordingly, short time Fourier transform (STFT) [27] of a signal $x(t)$ is defined as

$$\text{STFT}_x^y(t, f) = \int_{\tau} [x(\tau)\gamma(\tau - t)]e^{-j2\pi f\tau} d\tau. \quad (3)$$

STFT at time t is the Fourier transform of the signal $x(\tau)$ multiplied by a shifted analysis window $\gamma(\tau - t)$ centered around t . All integrals are from $-\infty$ to ∞ . Because multiplication by the relatively short window $\gamma(\tau - t)$ effectively suppresses the signal outside a neighborhood around the analysis time point $t = \tau$, the STFT is simply a local spectrum of the signal $x(\tau)$ around analysis window t . The windows can be overlapped to prevent loss of information. Although human behavior is a nonstationary stochastic process over a long interval, it can be considered stationary over a short time interval. Thus, STFT should give a good spectral representation of the human behaviors during that time interval.

3.2. Principal Component Analysis. The method can be described in brief as follows suppose that we have two sets of training samples: A and B . The number of training samples in each set is N . Φ_i represents each eigenvector produced by

principal component analysis (PCA). Each of the training samples, including positive samples and negative samples, can be projected into an axis extended by the corresponding eigenvector. By analyzing the distribution of the projected $2N$ points, we can roughly select the eigenvectors which have more human behavior information. The following is a detailed description of the process.

- (1) For a certain eigenvector Φ_i , compute its mapping result according to the two sets of training samples. The result can be described as $\lambda_{i,j}$ ($1 \leq i \leq M$, $1 \leq j \leq 2N$).
- (2) Train a classifier f_i using a simple method such as Perception or Neural Network which can separate $\lambda_{i,j}$ into two groups: specific valid driver or others with a minimum error $E(f_i)$.
- (3) If $E(f_i) < \theta$, then we delete this eigenvector from the original set of eigenvectors.

M is the number of eigenvectors and $2N$ is the total number of training samples. θ is the predefined threshold. The remaining eigenvectors are selected.

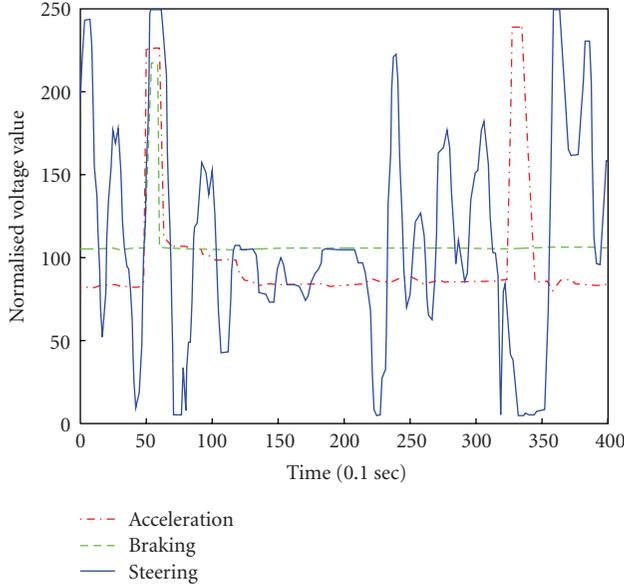


FIGURE 5: Human driving behavior data.

3.3. Independent Component Analysis. Apart from PCA, we also propose using independent component analysis (ICA) to reduce the dimensions of the data inputs for human behavior modeling. Independent component analysis is a statistical method which transforms an observed multidimensional vector into components that are statistically as independent as possible.

A fixed-point algorithm is employed for independent component analysis [28]. The goal of the ICA algorithm is to search for a linear combination of the prewhitened data $x'_i(t)$, where $y = w^T x'(t)$, such that the negentropy (non-gaussianity) is maximized. w is assumed to be bounded to have a norm of 1 and g' is the derivative of g . The fixed point algorithm [28] is as follows:

- (1) Generate an initial random vector w_{k-1} , $k = 1$
- (2) $w_k = E\{x'g(w_{k-1}^T x')\} - E\{g'(w_{k-1}^T x')\}w_{k-1}$
- (3) $w_k = w_k / \|w_k\|$
- (4) Stop if converged ($\|w_k - w_{k-1}\|$ is smaller than a certain defined threshold). Otherwise, increment k by 1 and return to step (2).

If the process converges successfully, the vector w_k produced can be converted to one of the underlying independent components by $w_k^T x'(t)$, $t = 1, 2, \dots, m$. Due to the whitening process, the columns of B are orthonormal. By projecting the current estimates on the subspace orthogonal to the columns of the matrix B which are found previously, we are able to retrieve the components one after the other.

4. Learning via Support Vector Machine

In this paper, SVM is applied within the framework of modeling human behaviors for intelligent vehicle security application. Inherent complexities and the nonlinearity of

human dynamic behavior make mathematical modeling difficult, hindering the use of conventional methods for process modeling and condition monitoring.

4.1. Mathematical Description. In SVM, the basic idea is to map the data X into a high-dimensional feature space \mathcal{F} via a nonlinear mapping Φ , and to do linear classification or regression in this space

$$f(\bar{x}) = (\omega \cdot \Phi(\bar{x})) + b \quad \text{with } \Phi: R^N \rightarrow \mathcal{F}, \omega \in \mathcal{F}, \quad (4)$$

where b is a threshold. Thus, linear regression in a high-dimensional (feature) space corresponds to nonlinear regression in the low-dimensional input space R^N . Note that the dot product in (4) between ω and $\Phi(\bar{x})$ would have to be computed in this high-dimensional space (which is usually intractable), if we are not able to use the kernel that eventually leaves us with dot products that can be implicitly expressed in the low-dimensional input space R^N . Since Φ is fixed, we determine ω from the data by minimizing the sum of the empirical risk $R_{\text{emp}}[f]$ and a complexity term $\|\omega\|^2$, which enforces flatness in feature space

$$R_{\text{reg}}[f] = R_{\text{emp}}[f] + \lambda \|\omega\|^2 = \sum_{i=1}^l C(f(\bar{x}_i) - y_i) + \lambda \|\omega\|^2, \quad (5)$$

where l denotes the sample size $(\bar{x}_1, \dots, \bar{x}_l)$, $C(\cdot)$ is the penalty term and λ is a regularization constant. For a large set of loss function, (5) can be minimized by solving a quadratic programming problem, which is uniquely solvable. It can be shown that the vector ω can be written in terms of the data points

$$\omega = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \Phi(\bar{x}_i), \quad (6)$$

with α_i, α_i^* being the solution of the aforementioned quadratic programming problem. α_i and α_i^* have an intuitive interpretation as forces pushing and pulling the estimate $f(\bar{x}_i)$ towards the measurements y_i . Taking (6) and (4) into account, we are able to rewrite the whole problem in terms of dot products in the low-dimensional input space

$$\begin{aligned} f(\bar{x}) &= \sum_{i=1}^l (\alpha_i - \alpha_i^*) (\Phi(\bar{x}_i) \cdot \Phi(\bar{x})) + b \\ &= \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(\bar{x}_i, \bar{x}) + b, \end{aligned} \quad (7)$$

where α_i, α_i^* are Lagrangian multipliers, and \bar{x}_i are support vectors.

In (7), we introduce a kernel function $K(\bar{x}_i, \bar{x}_j) = \Phi(\bar{x}_i) \cdot \Phi(\bar{x}_j)$. As explained in [29], any symmetric kernel function K satisfying Mercer's condition corresponds to a dot product in some feature space.

4.2. Approach. We propose to use SVM to model human driving behaviors. We consider human driving data $X(t)$ as the input vector of human dynamic behavior features as the time sequence. Since the SVM has an ability for classification, we use the human behavior data to “train” the SVM classifiers in the human dynamic behavior features space. For each driver, we want to design an SVM model to separate them from the other drivers. The task is to build up models across individuals by using the SVM training procedure. Then, since a dual class SVM model is capable of classifying two different classes of data only, in our applications, more than one dual class SVM is utilized and they are organized in a hierarchical manner. If there are n drivers to be recognized, $n - 1$ SVMs will be connected together as shown in Figure 6.

In any predictive learning task, such as classification, an appropriate representation of examples as well as the model and parameter estimation method should be selected to obtain a high level of performance of the learning machine. Traditional statistical approach to estimating models from data is based on parametric estimation. The basic fact that an assumption of an underlying dependency with a simple known parametric form is an ensuing need, limits its applicability in practice. Recent approaches allow a wide class of models of varying complexity to be chosen. The task of learning then amounts to selecting the model of optimal complexity and estimating parameters from training data. Under the SVM approach, the parameters to be usually chosen are the following.

- (1) The penalty term C which determines the tradeoff between the complexity of the decision function and the number of training examples misclassified.
- (2) The mapping function Φ .
- (3) The kernel function such that $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$.

5. Experimental Study

In this section, we conduct experiments based on the proposed methodology for recognition of driver identities by analyzing the driving performances. In order to estimate the performance of the proposed system, we invite 7 human subjects to attend the experiment, who are Meng, Ou, Ye, Huang, Wang, Wu, and Shen. They are asked to “drive” on the designed experimental platform individually. The raw data of their driving behaviors is collected by the Data Sensing and Capturing Subsystem. The data recorded is to be analyzed by the Data Analysis Subsystem aforementioned. Our objective is to identify the driving data by trained SVM models. We use the accuracy rate of the SVM classifications to evaluate the performance of the proposed system. The experimental results of applying different data preprocessing methods and choosing different parameters of SVM when modeling human driving behaviors are shown in what follows.

5.1. Preprocess Data Analysis. In the first series of experiments, we run different data preprocessing methods for the optimization. The raw data is captured at a rate of 10 Hz and overlapping windows are applied on the data to cut the data into segments. Each segment is 40 seconds long and can be considered as a matrix of size 400×3 .

We then apply FFT, PCA, and ICA to reduce the input size to the SVM for classification. The following steps are performed on each data segment.

- (1) Apply FFT of order 20 to transform each column of data of size 400 into 20, so the result retrieved is a matrix of size 20×3 , as each data segment. Then, align the data into a single row as a 1×60 vector.
- (2) Divide the raw data matrix into 10 parts by time sequence and align these 40×3 matrix to a 40×30 matrix. Extract two features from the gained data matrix using PCA. With 2 PCs, a 2×30 feature retrieved is aligned to form a 1×60 vector.
- (3) Divide the raw data matrix into 10 parts by time sequence and align these 40×3 matrix to a 40×30 matrix. Extract two features from the gained data matrix using ICA. With 2 ICs, a 2×30 feature retrieved is aligned to form a 1×60 vector.

We compare the data preprocessing using PCA and ICA with FFT. We simply train an SVM to distinguish one tester from all testing data, which is Meng’s, to evaluate the performances of three methods of feature selection and data reduction. We have 2 groups of data containing 348 raw data segments totally, 104 segments representing the behaviors of driver Meng (the authorized driver) and 244 segments representing non Meng (the unauthorized drivers). These segments are sent to SVM for training and testing. Due to the aforementioned rules, each segment is processed to a 1×60 vector as the input to SVM.

Three data preprocessing methods are tested independently and the SVM testing results are shown in Table 1 for comparing the ability of feature selection of each method. As seen, FFT achieves the best performance among those three feature selections and data reduction methods, so we choose FFT as the data preprocessing for the further procedures.

Next we examine the different parameters of FFT to the classification results. Table 2 shows the test results of classification using different sampling times (length of data segment) and different FFT orders (size of input vector). Different sampling times from 10 seconds to 160 seconds and different FFT orders of 5, 10, and 20 are conducted. When using the sampling time t and FFT orders n , it means each data segment is a matrix of size $10t \times 3$ as the original sampling rate of the hardware is 10 Hz. The FFT transforms each column of data of size $10t$ into n and the result retrieved is a matrix of size $n \times 3$. Then the data is aligned into a single row as a $1 \times 3n$ vector for the SVM training and testing.

Table 2 shows the average success rate of identifying driver Meng from all testing data, using different sampling times and FFT orders. Different combination of sampling time and FFT order affects different classification rates.

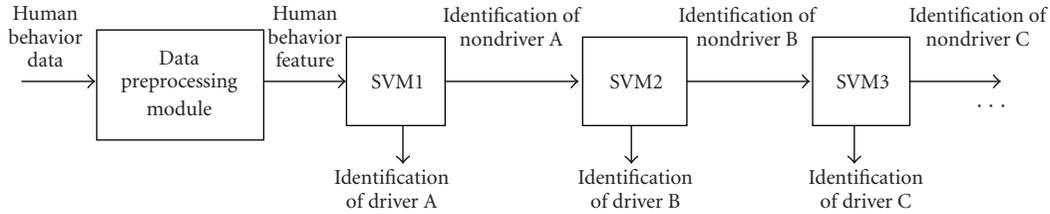


FIGURE 6: Block diagram of SVMs network classifier.

TABLE 1: Test results using different preprocessing methods.

	Number of errors (authorized driver)	Number of errors (unauthorized drivers)	Average success rate
FFT	22	46	80.46%
PCA	39	82	65.23%
ICA	47	79	63.79%

TABLE 2: Test results using different sampling time and FFT orders.

Sampling time	FFT (order 5)	FFT (order 10)	FFT (order 20)
10 sec.	64.92%	66.83%	64.68%
20 sec.	66.51%	66.51%	60.29%
40 sec.	69.23%	69.23%	78.85%
80 sec.	63.46%	55.77%	51.92%
160 sec.	65.38%	53.85%	65.38%

Shorter sampling time and smaller FFT order may cause the loss of important features involved in the human behavior signal, while longer time and larger order may lead to a mixed redundant signal to the classification procedure, both of them lower than the recognition rate. By comparing the success rate, sampling time at 40 seconds and FFT order of 20 achieve the best results. In addition, the data segment length of 40 seconds, namely, the time interval of the system examining the current driver, is efficient for not causing huge data segments for computation and providing adequate sampling frequency for testing driving performance in realtime.

5.2. Models Design. Training SVM requires the selection of parameters which influence the ensuing model performance. Therefore, to achieve a good model those parameters have to be chosen correctly. Examples, as stated earlier, are (1) cost function C and (2) the mapping function Φ . In the first part of our experiments, we have considered Gaussian radial basis function (RBF) as the kernel function. The RBF kernel is very advantageous in complex nonseparable classification problems due to its ability of nonlinear input mapping. It has the property that $\Phi(x) \cdot \Phi(y) = \exp(-\gamma \|x - y\|^2)$, and subsequently γ (defined as $1/2\sigma^2$ being the kernel width) is an important parameter to be chosen.

In this series of experiments, we run the SVM classifier with several values of C and γ somehow trying to determine which combination of parameters might be the best for a “good” model. That is, the one that could better express

the causal relation among variables which govern the quality within the driving platform. This is accessed through the evaluation of performance accuracy. One possible way is to divide the original data into a data training set and into a validation data set for model evaluation. Figure 7 shows the testing accuracy as a function of kernel γ both parameterized with C for identification of driver Meng. Figure 8(a) shows the variation of number of SVs versus γ and C and Figure 8(b) shows the variation of number of learning iterations versus γ and C . The stopping tolerance for solving the optimization problem is set to 10^{-3} .

In the second part of our experiments, we consider a polynomial kernel as the kernel function. It has the property that $\Phi(x) \cdot \Phi(y) = (\gamma \times x' \times y + c)^d$, where $c = 0$ and $d = 3$ as the default settings in this experiment, and subsequently γ (defined as $1/2\sigma^2$ being the kernel width) is an important parameter to be chosen.

Figure 9 shows the testing accuracy as a function of kernel γ both parameterized with C for identification of driver Meng. Figure 10(a) shows the variation of number of SVs versus γ and C and Figure 10(b) shows the variation of number of learning iterations versus γ and C . The stopping tolerance for solving the optimization problem is set to 10^{-3} .

From the results shown above, larger C corresponds to a smaller number of SVs as well as a larger number of training iterations. Larger γ corresponds to poorer balanced classification, which means the defective classifier model causing the lower accuracy on one side of the classification destination but higher accuracy on the other side, as well as the larger C doing so. Further explanation is required for these results taking into account both C and γ parameters. For nonseparable data, the penalty term $C \sum_{i=1}^l (f(\bar{x}_i - y_i))$ is able to reduce the training errors in the working data set. Therefore, the margin is an indicator of the generalization accuracy. In the absence of a method to compute the best tradeoff between the regularization term and the training errors, the balance sought by the SVMs technique is hard to find. Thus, a larger C corresponds to a higher penalty of training errors and clearly overfitting occurs. On the other

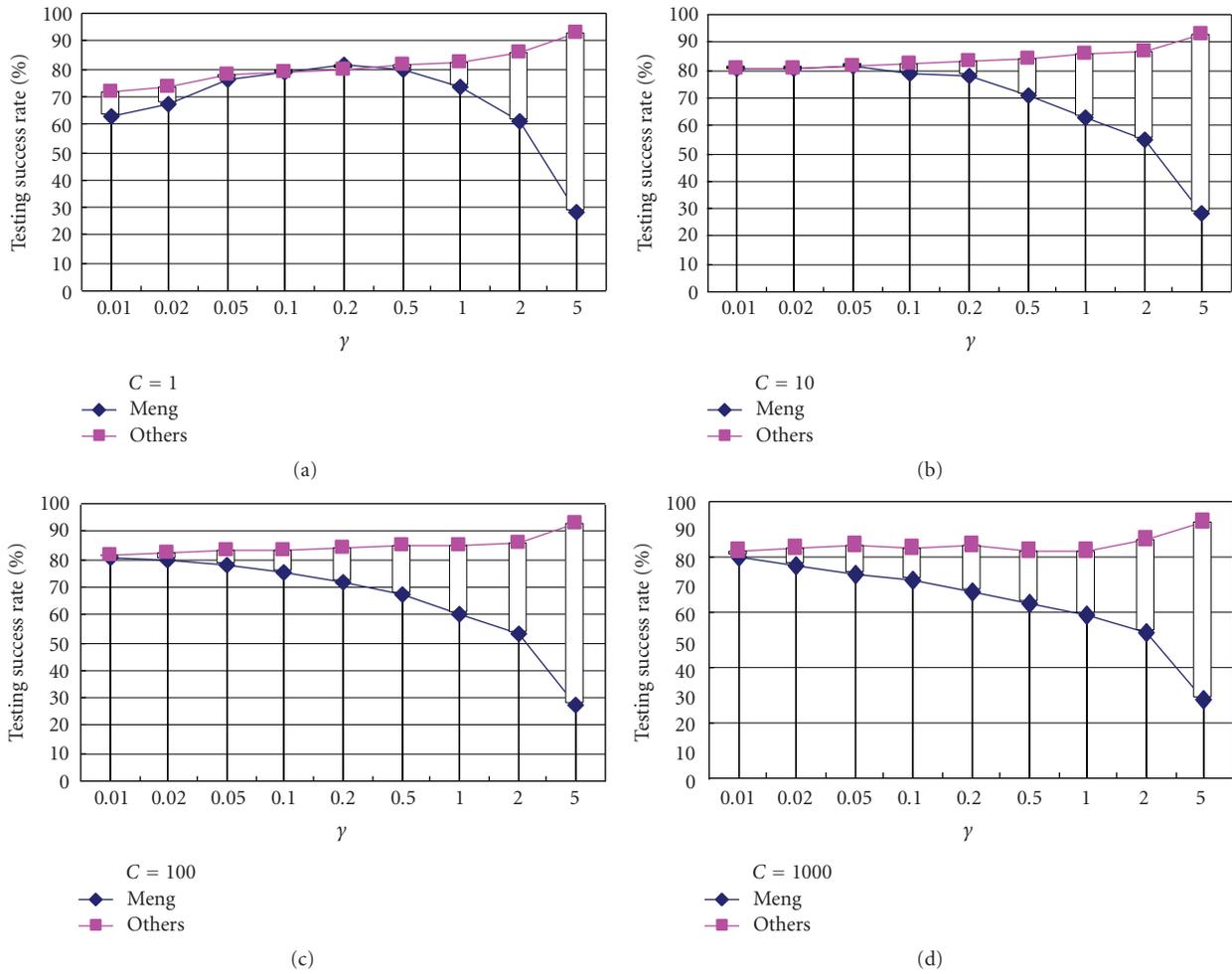


FIGURE 7: Success rate of model Meng versus Others parameterized with γ .

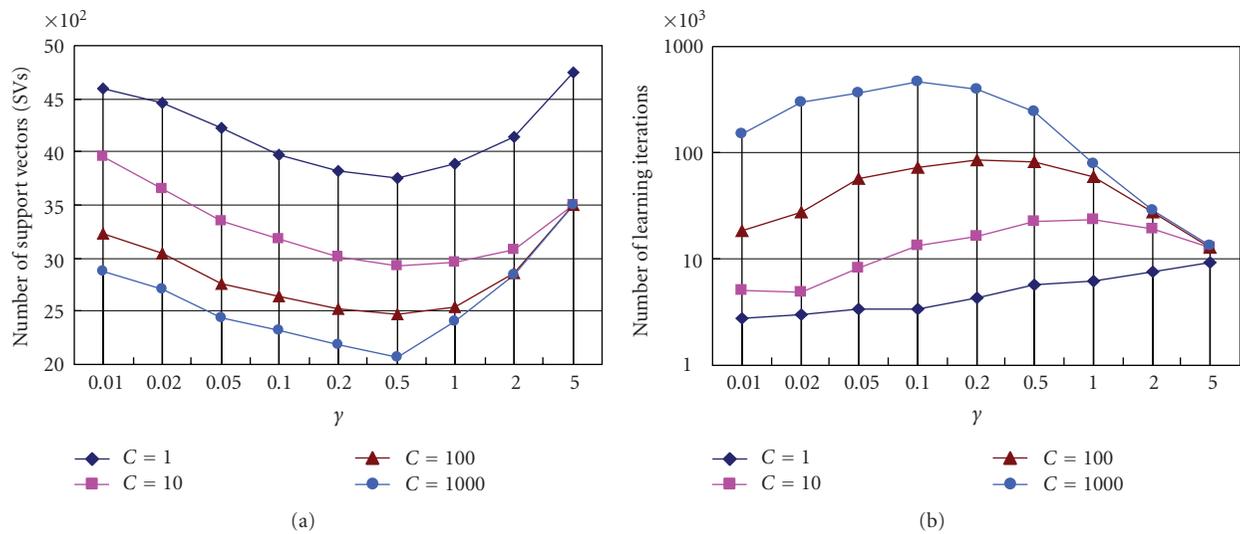


FIGURE 8: (a) Number of SVs versus γ ; (b) Number of learning iterations versus γ (model Meng versus Others parameterized with C).

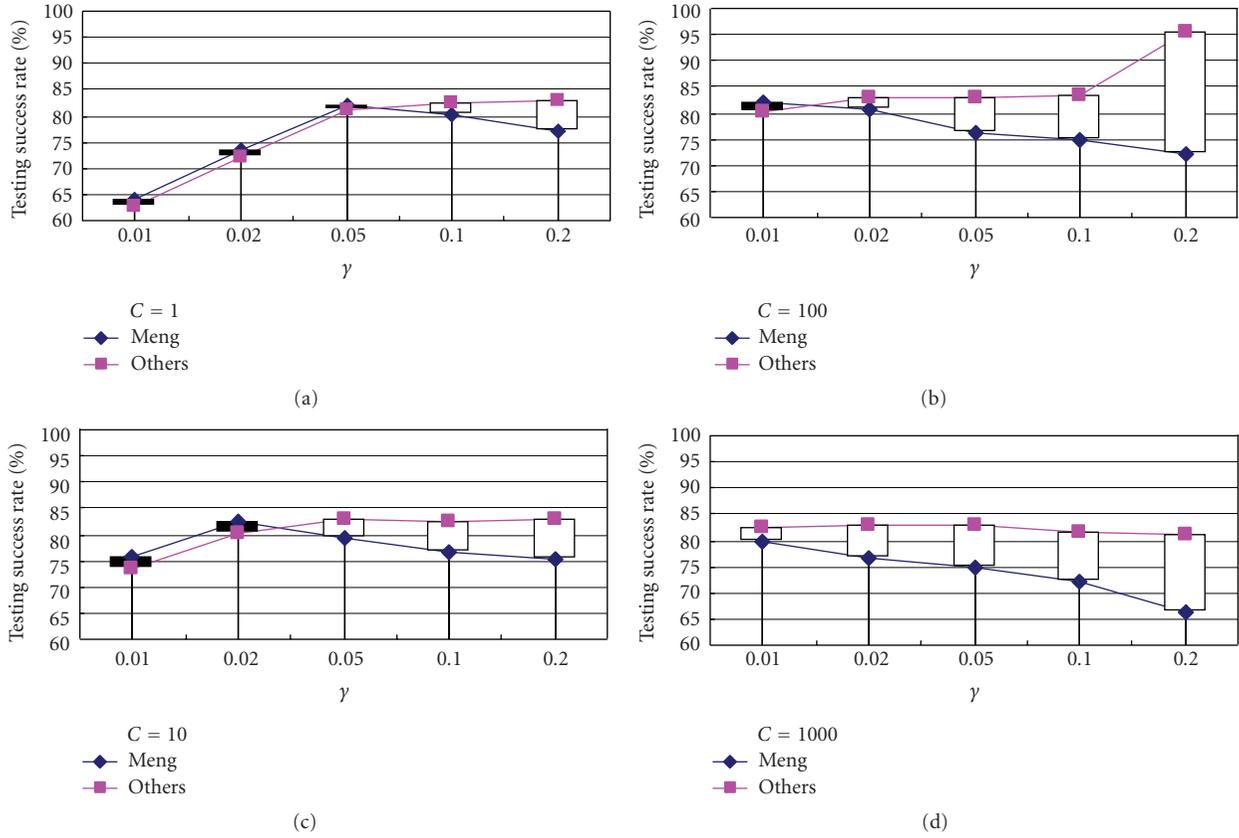


FIGURE 9: Success rate of model Meng versus Others parameterized with γ .

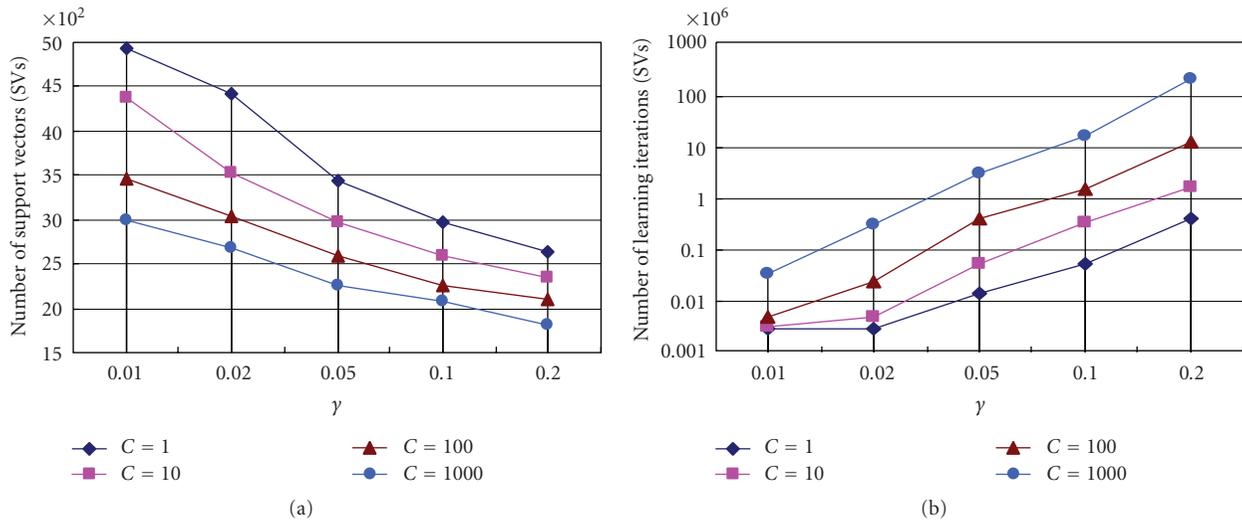


FIGURE 10: (a) Number of SVs versus γ ; (b) Number of learning iterations versus γ (model Meng versus Others parameterized with C).

hand, the higher when the kernel parameter γ becomes, the greater the variety of the decision boundaries that can be formed originating a more complex model. The added flexibility initially decreases the generalization error as the model can better fit the data. However, there is the danger that this can lead to overfitting as well.

Due to the requirements of the proposed system, we aim to achieve a high classification accuracy as well as low computational consumption. The aim of the identification system is for the vehicle to judge if it is his own driver, so we set the Meng's success rate with higher priority. It is found experimentally that $C = 10$ and $\gamma = 0.02$ in the polynomial

TABLE 3: Test results on driver identifications via SVM.

Meng	Ou	Ye	Huang	Wang	Wu	Shen
81.45%	93.62%	77.08%	84.62%	79.31%	90.00%	89.47%

kernel has the highest success rate for Meng. With the same method, we train 7 SVM models for all 7 human subjects and utilize the SVM network to form a multiple classifier. To test with our testing samples and evaluate with the identification accuracy, we derive the final results in Table 3. As seen, we achieve the average success rate over 85%.

5.3. Discussion. In this section, we demonstrate that SVM is a feasible parametric model for our proposed application. The first aspect investigated is to use preprocessing methods for feature extraction from large human dynamic behavior data for modeling purposes. The extension of the implementation is to the data sets in a larger scale and different methods of problem multiclass formulation. The feature extraction method based on FFT is found to be able to give the best data reduction results compared to PCA and ICA in the presented experiments of modeling human driving behaviors through SVM. FFT establishes a one-to-one mapping between the time domain and the frequency domain and preserves information from the original signal, ensuring that important features are not lost as a result of the transformation. Under the experimental criteria in this paper, FFT is proved to have a better performance to model human dynamic behavior for driver identification than PCA and ICA. Although PCA and ICA are often used for input reduction, it is not always useful because the variance of a signal is not necessarily related to the importance of the variable. Human behavior contains much signals at lower frequencies and FFT can retain the energy at this frequent area but PCA and ICA work bad as there are too many isotropically distributed clusters. By reducing the redundancy in the input data, the training process of the human driving behavior model becomes more efficient. After the unnecessary information is removed from the inputs, not only the key characteristics of the human behavior data can be retained, but also the modeling power of the SVM is actually improved.

Besides choosing preprocessing methods, the SVM model design is an important issue in this section. We have discussed the application of the multiclass SVM's classifiers and compared them with different SVM parameters to identify different drivers. The basic idea of SVM is to determine the structure of the classifier by minimizing the bounds of the training error and generalization error. The SVs close to the boundary decision surface determine the efficacy of the classifier. Based on the results from our application, SVM with polynomial kernel achieves the better performance. Our results demonstrate that the SVMs have the potential to obtain a reliable distinction among our testing human subjects, individual identification can be recognized with the multiclass SVM's classifiers with a success rate of over 85%, which verifies that the proposed SVM modeling method is valid and useful against the vehicle thefts problem.

6. Conclusion and Future Work

In this paper, we focus on the research of utilizing dynamic human behavior models for vehicle security (preventing vehicles from being stolen) application. By learning from driving performances, the intelligent classifier can be embedded into an IC-based car key, through which the vehicle security system can identify the valid drivers based on the ways the vehicle are driven and the drivers behave.

6.1. Conclusions. We proposed the innovative idea on driver identification system for detecting vehicle theft based on dynamic human behaviors. The dynamic and stochastic feature is difficult to be handled by traditional mathematical methods. We compared FFT, PCA and ICA in the data preprocessing, and proved FFT has better performance to process human dynamic behavior. Thereafter, machine learning method based on SVM is applied. We discussed the application of the multi-class SVM classifiers and compared the performance of different SVM parameters. SVM with polynomial kernel performs better than other functions.

6.2. Future Work. Choosing the best parameters, especially if a systematic approach is not used and/or the problem knowledge do not aid for proper selection, can be time consuming since we have to rely upon guessing and trial-and-error techniques. Therefore, an interactive grid search model selection method can be studied, which may further enhance the accuracy.

Acknowledgments

This work is partially supported by Hong Kong RGC CUHK417605, Hong Kong ITF ITP/003/09AP, GHP/006/09SZ, the grant from Key Laboratory of Robotics and Intelligent System, Guangdong Province (2009A060800016), the Knowledge Innovation Program of the Chinese Academy of Sciences Grant No. KG CX2-YW-156, and the grant from Shenzhen Hong Kong Innovation Circle.

References

- [1] "General Motor, OnStar," 2005, http://www.onstar.com/us_english/jsp/index.jsp.
- [2] http://en.wikipedia.org/wiki/Electronic_key.
- [3] A. Broggi, *Automatic Vehicle Guidance: The Experience of the ARGO Autonomous Vehicle*, World Scientific, Singapore, 1999.
- [4] S. M. Smith, "ALTRUISM: interpretation of three-dimensional information for autonomous vehicle control," *Engineering Applications of Artificial Intelligence*, vol. 8, no. 3, pp. 271–280, 1995.
- [5] S. L. M. Beauvais, C. Kreucher, and S. Lakshmanan, "Building world models for mobile platforms using heterogeneous sensors fusion and temporal analysis," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC '97)*, pp. 230–235, 1997.
- [6] H. Asada and S. Liu, "Transfer of human skills to neural net robot controllers," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2442–2448, 1991.

- [7] M. Nechyba, *Learning and Validation of Human Control Strategy*, Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, Pa, USA, 1998.
- [8] Y. Xu and Y. Ou, *Control of Single Wheel Robots*, Springer Tracts in Advanced Robotics, Springer, Berlin, Germany, 2005.
- [9] D. A. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*, Kluwer Academic, Boston, Mass, USA, 1994.
- [10] M. Nechyba and Y. Xu, “Learning and transfer of human real-time control strategies,” *Journal of Advanced Computational Intelligence*, vol. 1, no. 2, pp. 137–154, 1997.
- [11] M. C. Nechyba and Y. Xu, “Stochastic similarity for validating human control strategy models,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 3, pp. 437–451, 1998.
- [12] R. Herbrich, *Learning Kernel Classifiers: Theory and Algorithms*, MIT Press, Cambridge, Mass, USA, 2002.
- [13] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [14] B. Schölkopf, C. Burges, and A. Smola, *Advances in Kernel Methods Support Vector Learning*, MIT Press, Cambridge, Mass, USA, 1999.
- [15] V. Vapnik and A. Chervonenkis, “A note on one class of perceptrons,” *Automatic Remote Control*, vol. 25, 1964.
- [16] B. Boser, I. Guyon, and V. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory (COLT '92)*, pp. 144–152, 1992.
- [17] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [18] D. J. Sebald and J. A. Bucklew, “Support vector machine techniques for nonlinear equalization,” *IEEE Transactions on Signal Processing*, vol. 48, no. 11, pp. 3217–3226, 2000.
- [19] T. B. Trafalis and H. Ince, “Support vector machine for regression and applications to financial forecasting,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN '00)*, vol. 6, pp. 348–353, Como, Italy, 2000.
- [20] Y. Ou, H. Qian, and Y. Xu, “Support vector machine based approach for abstracting human control strategy in controlling dynamically stable robots,” *Journal of Intelligent and Robotic Systems*, vol. 55, no. 1, pp. 39–54, 2009.
- [21] P. Huang and Y. Xu, “SVM-based learning control of space robots in capturing operation,” *International Journal of Neural Systems*, vol. 17, no. 6, pp. 467–477, 2007.
- [22] X. Wu, Y. Ou, H. Qian, and Y. Xu, “A real time face classification and counting system,” *International Journal of Information Acquisition*, vol. 5, no. 1, pp. 1–10, 2008.
- [23] M. Chen, B. Huang, and Y. Xu, “Intelligent shoes for abnormal gait detection,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 2019–2024, Pasadena, Calif, USA, May 2008.
- [24] H. Qian, T. L. Lam, W. Li, C. Xia, and Y. Xu, “System and design of an omni-directional vehicle,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 389–394, Bangkok, Thailand, February 2008.
- [25] T. L. Lam, H. Qian, Y. Xu, and G. Xu, “Omni-directional steer-by-wire interface for four wheel independent steering vehicle,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1383–1388, 2009.
- [26] W. H. Press, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 2nd edition, 1992.
- [27] F. Hlawatsch and G. F. Boudreaux-Bartels, “Linear and quadratic time-frequency signal representations,” *IEEE Signal Processing Magazine*, vol. 9, no. 2, pp. 21–67, 1992.
- [28] A. Hyvarinen, “Fast and robust fixed-point algorithms for independent component analysis,” *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.
- [29] S. Bernhard, C. J. C. Burges, and A. Smola, *Advanced in Kernel Methods Support Vector Learning*, MIT Press, Cambridge, Mass, USA, 1998.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

