Hindawi Publishing Corporation
International Journal of Vehicular Technology
Volume 2014, Article ID 719413, 11 pages
http://dx.doi.org/10.1155/2014/719413

Research Article

Driving Posture Recognition by Joint Application of Motion History Image and Pyramid Histogram of Oriented Gradients

Chao Yan,[1] Frans Coenen,[2] and Bailing Zhang[1]

[1] *Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, SIP, Suzhou 215123, China*
[2] *Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK*

Correspondence should be addressed to Bailing Zhang; bailing.zhang@xjtlu.edu.cn

Received 3 August 2013; Accepted 30 October 2013; Published 28 January 2014

Academic Editor: Aboelmagd Noureldin

In the field of intelligent transportation system (ITS), automatic interpretation of a driver's behavior is an urgent and challenging topic. This paper studies vision-based driving posture recognition in the human action recognition framework. A driving action dataset was prepared by a side-mounted camera looking at a driver's left profile. The driving actions, including operating the shift lever, talking on a cell phone, eating, and smoking, are first decomposed into a number of predefined action primitives, that is, interaction with shift lever, operating the shift lever, interaction with head, and interaction with dashboard. A global grid-based representation for the action primitives was emphasized, which first generate the silhouette shape from motion history image, followed by application of the pyramid histogram of oriented gradients (PHOG) for more discriminating characterization. The random forest (RF) classifier was then exploited to classify the action primitives together with comparisons to some other commonly applied classifiers such as *k*NN, multiple layer perceptron, and support vector machine. Classification accuracy is over 94% for the RF classifier in holdout and cross-validation experiments on the four manually decomposed driving actions.

1. Introduction

In China, the number of personal-use automobiles has continued to grow at a rapid rate, reaching the number 120,890,000 in 2012. According to the World Health Organization (WHO), there is an estimated number of 250,000 deaths due to road accidents every year, making it the leading cause of death for people aged 14 to 44. Unsafe and dangerous driving accounts for the death of more than one million lives and over 50 million serious injuries worldwide each year [1]. The WHO also estimates that traffic accidents cost the Chinese economy over $21 billion each year. One of key contributing factors is reckless driving [1]. It is a proven fact that drivers who are reaching for an object such as a cell-phone are three times more likely to be involved in a motor vehicle accident, while actually using a cell-phone increases the risks to six times as likely.

In order to reduce unsafe driving behaviors, one of the proposed solutions is to develop a camera-based system to monitor the activities of drivers. This is particularly relevant for long-distance truck and bus drivers. For example, in many countries, including China, it is illegal for drivers to be using their cell-phone whilst driving. Drivers who violate the restriction face civil penalties. However, how to automatically distinguish between safe and unsafe driving actions is not a trivial technical issue. Since most commercial drivers operate alone, most of their driving behaviors are not directly observable by others. Such barriers will disappear when in-vehicle technologies become available to observe driver behaviors. An emerging technology that has attracted wide attention is the development of driver alertness monitoring systems which aims at measuring driver status and performance to provide in-vehicle warnings and feedback to drivers. Truck and bus fleet managers are particularly interested in such systems to acquire sound safety management. They can regularly track their driver outcomes and provide prevention of crashes, incidents, and violations.

Vision-based driving activity monitoring is closely related to human action recognition (HAR), which is an important area of computer vision research and applications. The goal

of the action recognition is to classify image sequences to a human action based on the temporality of video images. Much progress has been made on how to distinguish actions in daily life using cameras and machine learning algorithms. HAR has no unique definition; it changes depending on the different levels of abstraction. Moeslund et al. [2] proposed different taxonomies, that is, action primitive, action, and activity. An action primitive is a very basic movement that can be described at the decomposed level. An action is composed of action primitives that describes a cyclic or whole-body movement. Activities consist of a sequence of actions participated by one or more participates. In the recognition of drivers' action, the context is usually not taken into account, for example, the background environment variation outside the window and interactions with another person or moving object. Accordingly, this paper only focuses on partial body movements of the driver.

There exists some works on driver activity monitoring. To monitor a driver's behavior, some of the works focused on the detection of driver alertness through monitoring the eyes, face, head, or facial expressions [3–7]. In one study, the driver's face was tracked and yaw orientation angles were used to estimate the driver's face pose [8]. The Fisherface approach was applied by Watta et al. to represent and recognise the driver's seven poses, including looking over the left shoulder in the left rear-view mirror, at the road ahead, down at the instrument panel, at the centre rear-view mirror, at the right rear-view mirror, or over the right shoulder [9]. In order to minimize the influence of various illumination and background, Kato et al. used a far infrared camera to detect the driver face direction such as leftward, frontward, and rightward [10]. Cheng et al. presented a combination of thermal infrared and color images with multiple cameras to track important driver body parts and to analyze driver activities such as steering the car forward, turning left, and turning right [11]. Veeraraghavan et al. used the driver's skin-region information to group two actions; grasping the steering wheel and talking on a cell phone [12, 13]. Zhao et al. extended and improved Veeraraghavan's work to recognise four driving postures, that is, grasping the steering wheel, operating the shift lever, eating, and talking on a cell phone [14, 15]. Tran et al. studied driver's behaviors by foot gesture analysis [16]. Other works focused on capturing the driver's attention by combining different vision-based features and physical status of the vehicle [17–22].

The task of driver activity monitoring can be generally studied in the human action recognition framework, the emphasis of which is often on finding good feature representations that should be able to tolerate variations in viewpoint, human subject, background, illumination, and so on. There are two main categories of feature descriptions: global descriptions and local descriptions. The former consider the visual observation as a whole while the latter describe the observation as a collection of independent patches or local descriptors. Generally, global representation is derived from silhouettes, edges, or optical flow. One of the earliest global representation approaches, called motion history image (MHI), was proposed by Bobick and Davis [23], which extract silhouettes by using background subtraction and aggregate difference between subsequence in an action sequence. Other global description methods include the R transform [24], contour-based approach [25, 26], and optical flow [27–30]. The weakness of global representation includes the sensitivity to noise, partial occlusions, and variations in viewpoint. Instead of global representation, local representation describes the observation as a collection of space-time interesting points [31] or local descriptors, which usually does not require accurate localisation and background subtraction. Local representation has the advantage of being invariant to different of viewpoint, appearance of person, and partial occlusions. The representative local descriptor is the space-time interest point detectors proposed by Laptev and Lindeberg [31], which however has the shortcoming of only having a small number of stable interest points available in practice. Some of their derivations have been proposed, for example, extracted space-time cuboids [32].

In this paper, we studied drivers' activity recognition by comprehensively considering action detection, representation, and classification. Our contributions include three parts. The first part is our deviation from many published works on drivers' posture based on static images from drivers' action sequence, which has the potential problem of confusion caused by similar postures. It is very possible that two frames of vision-similar posture are extracted from two completely different action image sequences. For example, the moment/frame that a driver moves the cell phone across his or her mouth can be confused as eating. Following the action definition in [2] which is based on the combination of basic movements, we regard driving activity as space-time action instead of static space-limited posture. The main driving activity we considered are hand-conducted actions such as eating and using a cell phone.

The second contribution of this paper is our proposal of the driving action decomposition. Generally, the driving actions that take place in the drivers seat are mainly performed by hand, which include but are not limited to eating, smoking, talking on the cell phone, and operating the shift lever. These actions or activities are usually performed by shifting the hand position, which is confined to the drivers seat. Following the train of thought in [2], we regard the actions or activities as a combination of a number of basic movements or action primitives. We created a driving action dataset similar to the SEU dataset [14], with four different types of action sequences, including operating the shift lever, responding to a cell phone call, eating and smoking. The actions are then decomposed into four action primitives, that is, hand interaction with shift lever, hand operating the shift lever, hand interaction with head, and hand interaction with dashboard. Upon the classification of these action primitives, the driving actions involving eating, smoking, and other abnormal behaviors can be accordingly recognised as a combination of action primitives [33].

The last contribution of this paper is the proposal of a global grid-based representation for the driving actions, which is a combination of the motion history image (MHI) [23] and pyramid histogram of oriented gradients (POHG) [34], and the application of random forest classifier (RF) for the driving actions recognition. Encoding the region

of interest in the drivers seat is a natural choice as there are few noises and no partial occlusions in the video. The action silhouettes were first extracted to represent action primitives by applying MHI to aggregate the difference between subsequent frames. To have better discrimination than MHI alone, the pyramid histogram of oriented gradient of the MHI was calculated as the features for further training and classification. PHOG is a spatial pyramid extension of the histogram of gradients (HOG) [35] descriptors, which has been used extensively in computer vision. After the comparison of several different classification algorithms, the random forest (RF) classifier was chosen for the driving action recognition, which offers satisfactory performance.

The rest of the paper is organized as follows. Section 2 gives a brief introduction on our driving posture dataset creation and the necessary preprocessing. Section 3 and Section 4 review the motion history image and the pyramid histogram of oriented gradients, with explanation of how they are applied in driving posture description, respectively. Section 5 introduces the random forest classifier and other three commonly used classification methods for comparison. Section 6 reports the experiment results, followed by conclusion in Section 7.

2. Driving Action Dataset Creation and Preprocessing

A driving action dataset was prepared which contains 20 video clips in 640×424@24 fps. The video was recorded using a Nikon D90 camera at a car park in the Xi'an Jiaotong-Liverpool University. Ten male drivers and ten female drivers participated in the experiment by pretending to drive in the car and conducting several actions that simulated real driving situations. Five predefined driving actions were imitated, that is, turning the steering wheel, operating the shift lever, eating, smoking, and using a cell phone.

There are five steps involved in simulating the driving activities by each participant.

Step 1. A driver first grasps the steering wheel and slightly turns the steering wheel.

Step 2. The driver's right hand moves to shift the lever and operates it for several times before moving back to the steering wheel.

Step 3. The driver takes a cell phone from the dashboard and responds to a phone call and then puts it back by his or her right hand.

Step 4. The driver takes a cookie from the dashboard and eats it using his or her right hand.

Step 5. For male drivers, he takes a cigarette from the dashboard, and puts it into his mouth and then uses a lighter to light the cigarette and then puts it back on the dashboard.

This experiment extracted twenty consecutive picture sequences from the video of the dataset for further experimentation.

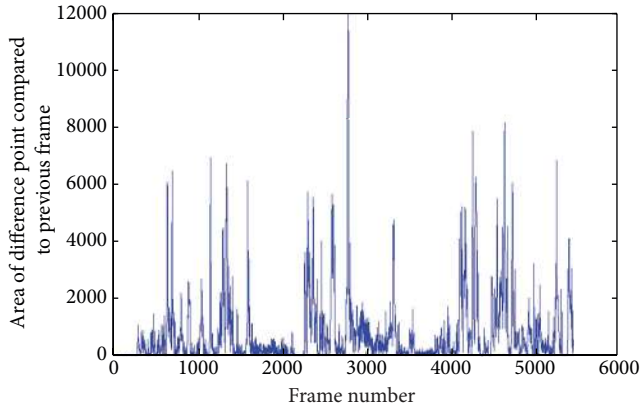

Figure 1: The vertical axis stands for area of difference point compared to previous frame by applying Otsu's thresholding method [26]; the horizontal axis stands for the frame number.

2.1. Action Detection and Segmentation. Being similar to many intelligent video analysis systems, action recognition should start with motion detection in a continuous video stream for which many approaches are available. Among the popular approaches, frame differencing is the simplest and most efficient method which involves taking the difference between two frames to detect the object. Frame differencing is widely applied with proven performance, particularly when a fixed camera is used to observe dynamic events in a scene.

With each driving action sequence, the frame differences between two adjacent image frames are first calculated, followed by thresholding operation to identify moving objects. Otsu's thresholding method [26] was chosen, which minimizes the intraclass variance of the black and white pixels. The existence of moving objects will be determined by evaluating whether there exist connected regions in the binary image. And the location of the moving objects can be further calculated based on the total areas and coordinate of connected regions. The details can be illustrated by Figure 1.

In the following section, the segmented actions images are further manually labelled into four different categories of action primitives based on the trajectory of driver's right hand as shown in Figure 2. The first category of action is moving to shift lever with the right hand from the steering wheel or moving back to the steering wheel from the shift lever. The second category of action is operating the shift lever with the right hand. The third category of action is moving to the dashboard from the steering wheel with the right hand or moving back to the steering wheel from the dashboard with the right hand. The fourth category of action is moving to the head from the steering wheel or moving back to the steering wheel from the head with the right hand.

3. Motion History Image (MHI)

Motion history image (MHI) approach is a view-based temporal template approach, developed by Bobick and Davis [23], which is simple but robust in the representation of movements and is widely employed in action recognition,

motion analysis, and other related applications [36–38]. The motion history image (MHI) can describe how the motion is moving in the image sequence. Another representation called motion energy image (MEI) can demonstrate the presence of any motion or a spatial pattern in the image sequence. Both MHI and MEI templates comprise the motion history image (MHI) template-matching method.

Figure 3 shows a movement in driving. The first row is some key frames in a driving action. The second and third rows are the frame differences and the corresponding binary images from applying Otsu's thresholding. The fourth and fifth rows are cumulative MEI and MHI images, respectively. MHI's pixel intensity is a function of the recency of motion in a sequence where brighter values correspond to more recent motion. We currently use a simple replacement and linear decay operator using the binary image difference frames. The formal definitions are briefly explained below:

$$\operatorname{diff}(x,y,t) = \begin{cases} \operatorname{zeros}(x,y,1), & \text{if } t = 1, \\ \left|I(x,y,t-1) - I(x,y,t)\right|, & \text{otherwise}, \end{cases} \tag{1}$$

where $\operatorname{diff}(x,y,t)$ is a difference image sequence indicating the difference compared to previous frame. Let

$$D(x,y,t) = \begin{cases} 0, & \text{if } \operatorname{diff}(x,y,t) < \text{threshold}, \\ 1, & \text{otherwise}, \end{cases} \tag{2}$$

where $D(x,y,t)$ is binary images sequence indicating region of motion. Then the motion energy image is defined as

$$\operatorname{MEI}(x,y,t) = \bigcup_{t=\text{action\_start\_frame}}^{t=\text{action\_end\_frame}} D(x,y,t). \tag{3}$$

Both motion history images and motion energy images were introduced to capture motion information in images [23]. While MEI only indicates where the motion is, motion history image $\operatorname{MHI}(x,y,t)$ represents the way the object moving, which can be defined as

$$\operatorname{MHI} = \begin{cases} 255, & \text{if } D(x,y,t) = 1, \\ \max\left\{0, \operatorname{MHI}(x,y,t-1) - 1\right\}, & \text{if } D(x,y,t) \neq 1,\ \dfrac{255}{\text{pic}_{\text{seq}_{\text{length}}}} \leq 1, \\ \max\left\{0, \operatorname{MHI}(x,y,t-1) - \operatorname{floor}\left(\dfrac{255}{\text{pic\_seq\_length}}\right)\right\}, & \text{if } D(x,y,t) \neq 1,\ \dfrac{255}{\text{pic}_{\text{seq}_{\text{length}}}} > 1. \end{cases} \tag{4}$$

The result is a scalar-valued image where latest moving pixels are the brightest. MHI can represent the location, the shape, and the movement direction of an action in a picture sequence. As MEI can be obtained by thresholding the MHI above zero, we will only consider features derived from MHI in the following.

After the driving actions were detected and segmented from the raw video dataset, motion history images were extracted for each of the four decomposed action sets. Figure 4 demonstrates how the motion history image is calculated to represent movements for each decomposed action sequence. In the figure, the left column and the middle column are the start and end frames of a decomposed action snippet, respectively. The right column is the MHI calculated for the corresponding action snippet.

4. Pyramid Histogram of Oriented Gradients (PHOG)

Motion history image MHI is not appropriate to be directly exploited as features for the purpose of comparison or classification in practical applications. In the basic MHI method [23], after calculating the MHI and MEI, feature vectors are calculated employing the seven high-order Hu moments. Then these feature vectors are used for recognition. However, Hu's moment invariants have some drawbacks, particularly limited recognition power [39]. In this paper, the histogram of oriented gradients feature is extracted from the MHI as the suitable features for classification.

In many image processing tasks, the local geometrical shapes within an image can be characterized by the distribution of edge directions, called histograms of oriented gradients (HOG) [35]. HOG can be calculated by evaluating a dense grid of well-normalized local histograms of image gradient orientations over the image windows. HOG has some important advantages over other local shape descriptors; for example, it is invariant to small deformations and robust in terms of outliers and noise.

The HOG feature encodes the gradient orientation of one image patch without considering where this orientation originates from in this patch. Therefore, it is not discriminative enough when the spatial property of the underlying structure of the image patch is important. The objective of a newly proposed improved descriptor pyramid histogram of oriented gradients (PHOG) [34] is to take the spatial property of the local shape into account while representing an image by HOG. The spatial information is represented by tiling the image into regions at multiple resolutions, based on spatial pyramid matching [40]. Each image is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction. The number of points in each grid cell is then recorded.

Figure 2: Four manually decomposed action primitives.

The number of points in a cell at one level is simply the sum over those contained in the four cells it is divided into at the next level, thus forming a pyramid representation. The cell counts at each level of resolution are the bin counts for the histogram representing that level. The soft correspondence between the two point sets can then be computed as a weighted sum over the histogram intersections at each level.

The resolution of an MHI image is 640×480. An MHI is divided into small spatial cells based on different pyramid levels. We follow the practice in [34] by limiting the number of levels to $L = 3$ to prevent overfitting. Figure 5 shows that the pyramid at level l has $2^n \times 2^n$ cells.

The magnitude $m(x, y)$ and orientation $\theta(x, y)$ of the gradient on a pixel (x, y) are calculated as follows:

$$(x, y) = \sqrt{g_x(x, y)^2 + y_y(x, y)^2},$$
$$\theta(x, y) = \arctan \frac{g_x(x, y)}{g_y(x, y)}, \quad (5)$$

where $g_x(x, y)$ and $g_y(x, y)$ are image gradients along the x and y directions. Each gradient orientation is quantized into K bins. In each cell of every level, gradients over all the pixels are concatenated to form a local K bins histogram. As a result, a ROI at level l is represented as a $K2^l2^l$ dimension vector. All the cells at different pyramid levels are combined to form a final PHOG vector with dimension of $d = K\sum_{l=0}^{L} 4^l$ to represent the whole ROI.

The dimension of the PHOG feature (e.g., $d = 680$ when $K = 8; L = 3$) is relatively high. Many dimension reduction methods can be applied to alleviate the problem. We employ the widely used principal component analysis (PCA) [41] due to its simplicity and effectiveness.

5. Random Forest (RF) and Other Classification Algorithms

Random forest (RF) [42] is an ensemble classifier using many decision tree models, which can be used for classification or regression. A special advantage of RF is that the accuracy and variable importance information is provided with the results. Random forests create a number of classification trees. When an vector representing a new object is input for classification, it was sent to every tree in the forest. A different subset of the training data are selected (≈2/3), with replacement, to train each tree, and remaining training data are used to estimate error and variable importance. Class assignment is made by the number of votes from all of the trees.

RF has only two hyperparameters, the number of variables M in the random subset at each node and the number

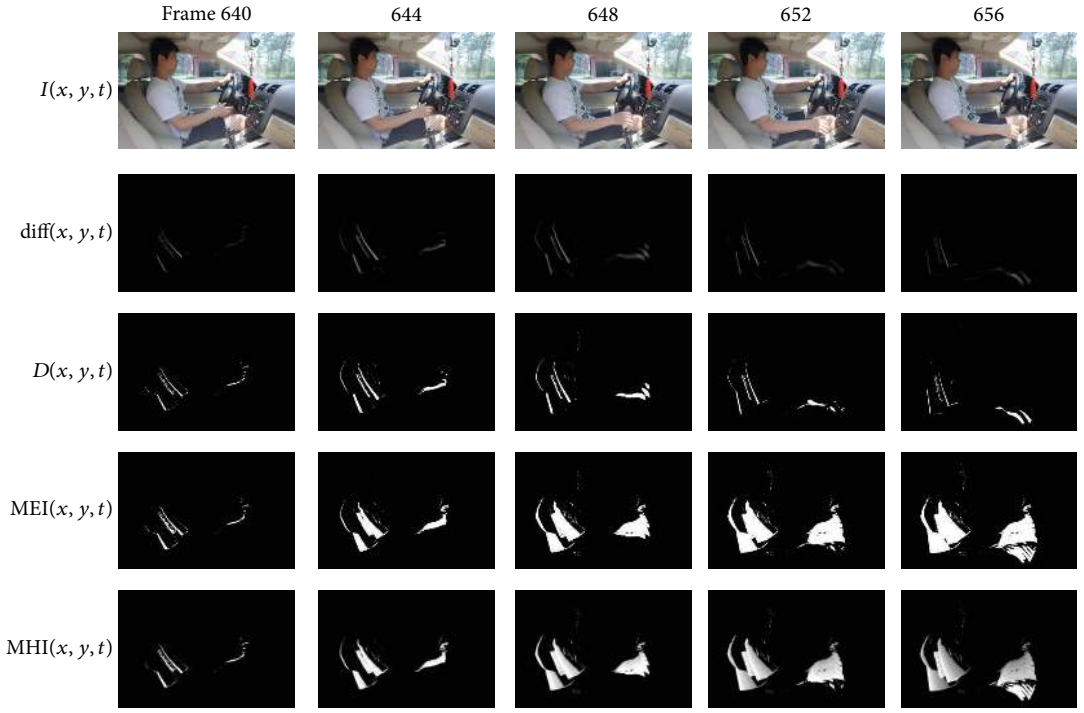

Figure 3: Example of the driver's right hand moving to shift lever from steering wheel. The first row is some key frames in a driving action. The second row is the corresponding frame difference images. The third row is binary images resulted from thresholding. The forth row is cumulative motion energy images. The fifth row is cumulative motion history images.

of trees T in the forest [42]. Breima's RF error rate depends on two parameters: the correlation between any pair of trees and the strength of each individual tree in the forest. Increasing the correlation increases the forest error rate while increasing the strength of the individual trees decreases this misclassification rate. Reducing M reduces both the correlation and the strength M is often set to the square root of the number of inputs.

When the training set for a particular tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample set.

The RF algorithm can be summarised as follows.

(1) Choose parameter T, which is the number of trees to grow.

(2) Choose parameter m, which is used to split each node, and $m = M$, where M is the number of input variables and m is held constant while growing the forest.

(3) Grow T trees. When growing each tree do the following.

  (i) Construct a bootstrap sample of size n sampled from $S_n = (X_i, y_i)\ (i = 1)^n$ with replacement and grow a tree from this bootstrap sample.

  (ii) When growing a tree at each node, select m variables at random and use them to find the best split.

  (iii) Grow the tree to a maximal extent. There is no pruning.

(4) To classify point X collect votes from every tree in the forest and then use majority voting to decide on the class label.

In this paper, we also compared the accuracy of RF and several popular classification methods, including k-nearest neighbor (kNN) classifier, multilayer perceptron (MLP), and Support Vector Machines (SVM) on the driving action datasets.

5.1. Other Classification Methods

5.1.1. k–Nearest Neighbor Classifier. k-nearest neighbour (kNN) classifier, one of the most classic and simplest classifier in machine learning, classifies object based on the minimal distance to training examples in feature space by a majority vote of its neighbours [41]. As a type of lazy learning, kNN classifier does not do any distance computation or comparison until the test data is given. Specifically, the object is assigned to the most common class among its k nearest neighbours. For example, the object is classified as the class of its nearest neighbour if k equals 1. Theoretically, the error rate of kNN algorithm is infinitely close to Bayes error while the training set size is infinity. However, a satisfactory performance of kNN algorithm prefers a large number of training data set which results in expensive computation in practical.

5.1.2. Multilayer Perceptron Classifier. In neural network, multilayer perceptron (MLP) is an extension of the single

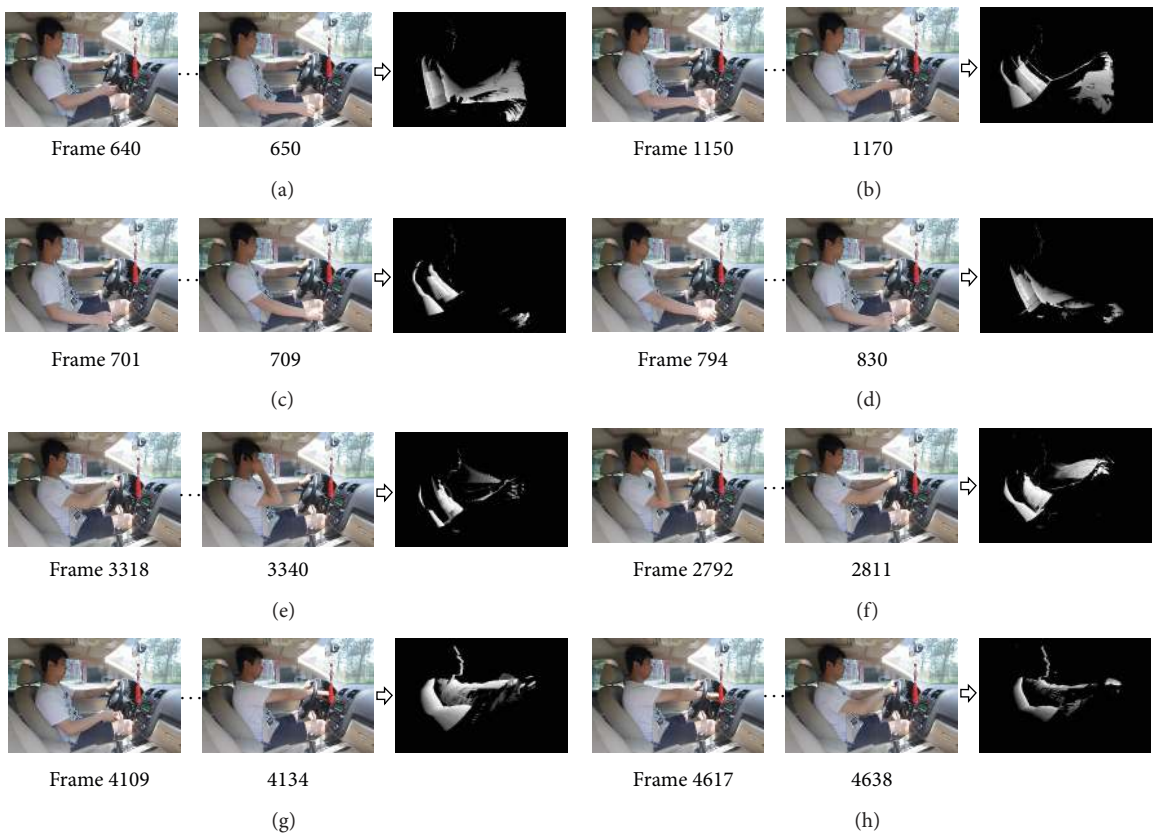

Figure 4: MHIs for different driving actions. (a) Right hand moving to shift lever. (b) Right hand moving back to steering wheel from shift lever. (c) Right hand operating the shift lever. (d) Operating the shift lever. (e) Right hand moving to head from steering wheel. (f) Right hand moving back to steering wheel. (g) Right moving back to steering wheel from dashboard. (h) Right hand moving to dashboard from steering wheel.

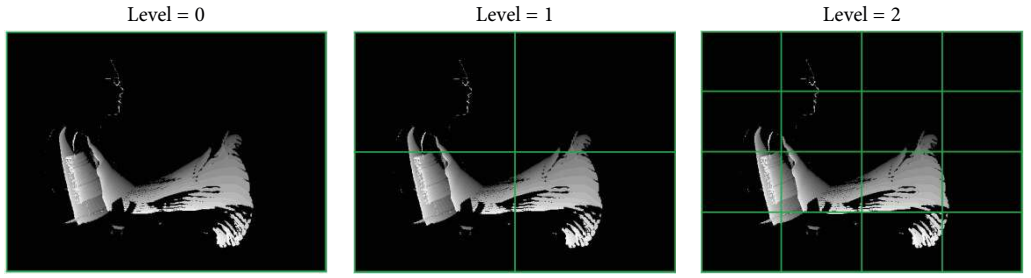

Figure 5: A schematic illustration of PHOG. At each resolution level, PHOG consists of a histogram of orientation gradients over each image subregion.

layer linear perceptron by adding hidden layers in between [41]. An MLP is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes, that is, the input layer, single or multiple hidden layer, and an output layer. An MLP classifier is usually trained by the error backpropagation algorithm.

5.1.3. Support Vector Machine. Support vector machine (SVM) is one of the most commonly applied supervised learning algorithms. A SVM is formally defined by a separating hyperplane which is in a high or infinite dimensional space. Given labeled training data, SVM will generate an optimal hyperplane to categorize new examples. Intuitively, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. And the optimal separating hyperplane maximizes the margin of the training data.

6. Experiments

6.1. Holdout Experiment. We choose the two standard experimental procedures, namely, holdout approach and the cross-validation approach, to verify the driving action recognition

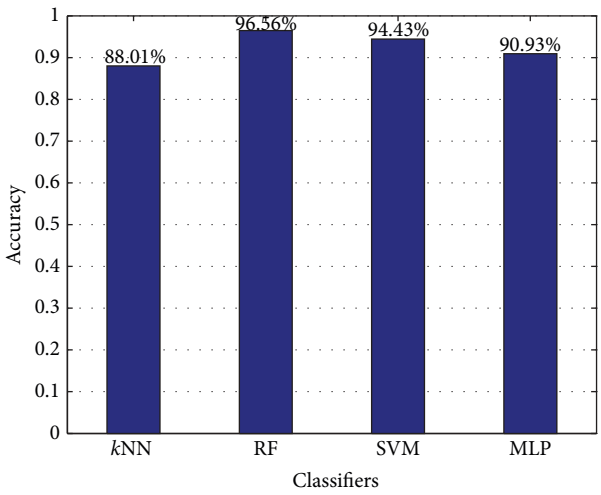

Figure 6: Bar plots of classification rates from holdout experiment with 80% of data are used for training and the remaining for testing.

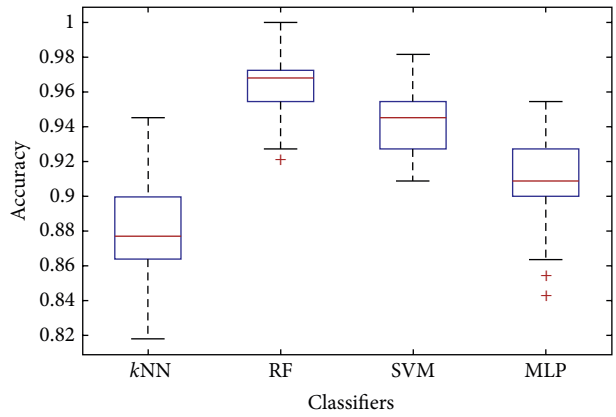

Figure 7: Box plots of classification rates from holdout experiment with 80% of data are used for training and the remaining for testing.

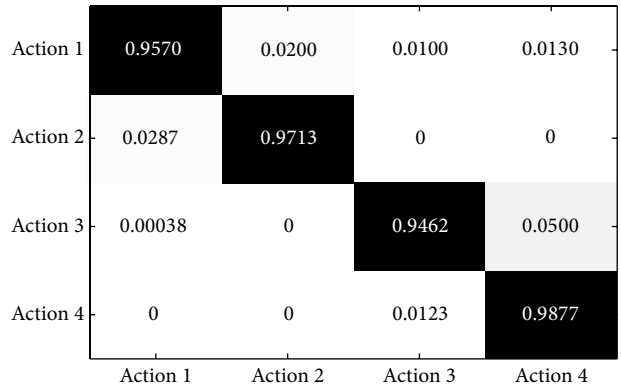

Figure 8: Confusion matrix of RF classification result from the holdout experiment.

performance using RF classifier and the PHOG feature extracted from MHI. Other three classifiers, kNN, MLP, and SVM, will be compared.

In the holdout experiment, 20% of the PHOG features are randomly selected as testing dataset, while the remaining 80% of the features are used as training dataset. The holdout experiment is usually repeated 100 times and the classification results are recorded. In each holdout experiment cycle, the same training and testing dataset are applied to the four different classifiers simultaneously to compare their performance.

Generally, classification accuracy is one of the most common indicators used to evaluate the performance of the classification. Figures 6 and 7 are the bar plots and box plots of the classification accuracies from the four classifiers with the same decomposed driving actions. The results are the average from 100 runs. The average classification accuracies of kNN classifier, RF classifier, SVM classifier, and MLP classifier are 88.01%, 96.56%, 94.43%, and 90.93%, respectively. It is obvious that the RF classifier performs the best among the four classifiers compared.

To further evaluate the performance of RF classifier, confusion matrix is used to visualize the discrepancy between the actual class labels and predicted results from the classification. Confusion matrix gives the full picture at the errors made by a classification model. The confusion matrix shows how the predictions are made by the model. The rows correspond to the known class of the data, that is, the labels in the data. The columns correspond to the predictions made by the model. The value of each of element in the matrix is the number of predictions made with the class corresponding to the column, for example, with the correct value as represented by the row. Thus, the diagonal elements show the number of correct classifications made for each class, and the off-diagonal elements show the errors made. Figure 8 shows the confusion matrix from the above experiment for the RF classifier.

In the figure, classes labelled as one, two, three, and four correspond to hand interaction with shift lever, operating the shift lever, interaction with head, and interaction with dashboard, respectively. In the confusion matrix, the columns are the predicted classes while the rows are the true ones. For the RF classifier, the average classification rate of the four driving actions is 96.56%. The respective classification accuracies for the four driving actions are 95.7%, 97.13%, 94.62% and 98.77% in holdout experiment, respectively. It shows that the classes one and two tend to be easily confused with each other, with error rate of about 2% and 2.87%, respectively. On the other hand, the error rates from the confusion between classes three and four lie between 1.2% and 5%.

6.2. k-Ford Cross-Validation. The second part of our experiment is to use k-ford cross-validation to further confirm the classification performance of the driving actions. In k-fold cross-validation, the original sets of data will be portioned into k subsets randomly. One subset is retained as the validation data for testing while the remaining $k - 1$ subsets are used as training data. The cross-validation process will then be repeated k times, which means that each of the k

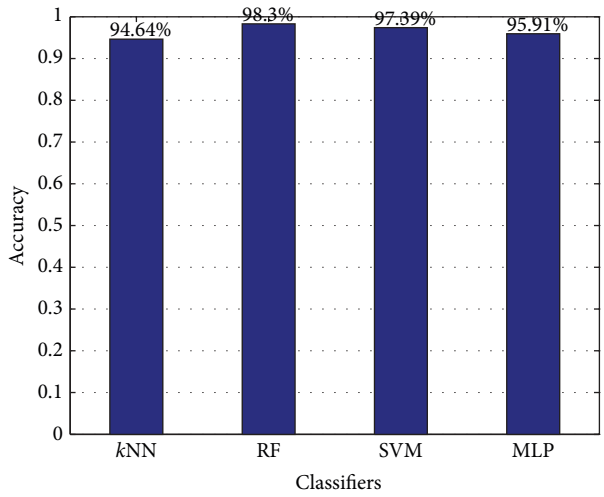

Figure 9: Bar plots of classification rates from 10-fold cross-validation.

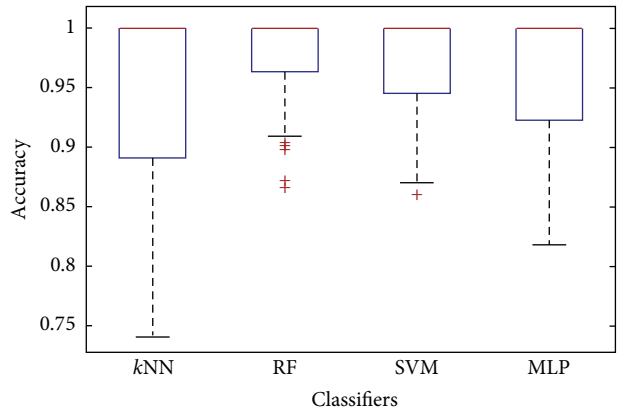

Figure 10: Box plots of classification rates from 10-fold cross-validation.

subsamples will be used exactly once as the validation data. The estimation can be the average of the k results. The key property of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once. We chose 10-fold cross-validation in the experiment, which means that nine of the ten splitted sets are used for training and the remaining one is reserved for testing.

The evaluation procedure is similar to the holdout experiment. The cross-validation experiment was also conducted 100 times for each of the classification methods. Each time the PHOG feature extracted from the driving action dataset was randomly divided into 10 folders. The average classification accuracies of the 100 repetitions are shown in the bar plots of Figure 9 and box plots of Figure 10. The average classification accuracies of k-NN classifier, RF classifier, SVM classifier, and MLP classifier are 94.64%, 98.30%, 97.39%, and 95.91%, respectively. From the bar plots, box plots and confusion matrix in Figures 9, 10, and 11, the RF classifier clearly outperforms other three classifiers compared.

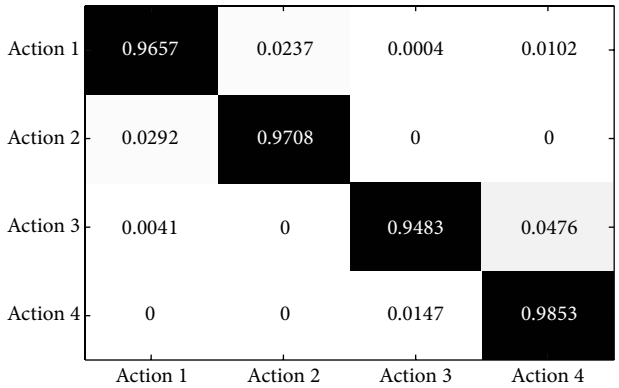

Figure 11: Confusion matrix of RF classification from 10-fold cross-validation experiment.

7. Conclusion

In this paper, we proposed an efficient approach to recognise driving action primitives by joint application of motion history image and pyramid histogram of oriented gradients. The proposed driving action primitives lead to the hierarchical representation of driver activities. The manually labelled action primitives are jointly represented by motion history image and pyramid histogram of oriented gradient (PHOG). The random forest classifier was exploited to evaluate the classification performance, which gives an accuracy of over 94% from the holdout and cross-validation experiments. This compares favorably over some other commonly used classifications methods.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The project is supported by Suzhou Municipal Science and Technology Foundation Grants SS201109 and SYG201140.

References

[1] "WHO World report on road traffic injury prevention," http://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/en/.

[2] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.

[3] X. Fan, B.-C. Yin, and Y.-F. Sun, "Yawning detection for monitoring driver fatigue," in *Proceedings of the 6th International Conference on Machine Learning and Cybernetics (ICMLC '07)*, pp. 664–668, Hong Kong, August 2007.

[4] R. Grace, V. E. Byrne, D. M. Bierman et al., "A drowsy driver detection system for heavy vehicles," in *Proceedings of the 17th Digital Avionics Systems Conference*, vol. 2, pp. I36/1–I36/8, Bellevue, Wash, USA, 1998.

[5] E. Wahlstrom, O. Masoud, and N. Papanikolopoulos, "Vision-based methods for driver monitoring," in *Proceedings of the IEEE Intelligent Transportation Systems*, vol. 2, pp. 903–908, Shanghai, China, October 2003.

[6] A. Doshi and M. M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 453–462, 2009.

[7] J. J. Jo, S. J. Lee, H. G. Jung, K. R. Park, and J. J. Kim, "Vision-basedmethod for detecting driver drowsiness and distraction in driver monitoringsystem," *Optical Engineering*, vol. 50, no. 12, pp. 127202–127224, 2011.

[8] X. Liu, Y. D. Zhu, and K. Fujimura, "Real-time pose classication for driver monitoring," in *Proceedings of the IEEE Intelligent Transportation Systems*, pp. 174–178, Singapore, September 2002.

[9] P. Watta, S. Lakshmanan, and Y. Hou, "Nonparametric approaches for estimating driver pose," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 2028–2041, 2007.

[10] T. Kato, T. Fujii, and M. Tanimoto, "Detection of driver's posture in the car by using far infrared camera," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 339–344, Parma, Italy, June 2004.

[11] S. Y. Cheng, S. Park, and M. M. Trivedi, "Multi-spectral and multi-perspective video arrays for driver body tracking and activity analysis," *Computer Vision and Image Understanding*, vol. 106, no. 2-3, pp. 245–257, 2007.

[12] H. Veeraraghavan, N. Bird, S. Atev, and N. Papanikolopoulos, "Classifiers for driver activity monitoring," *Transportation Research C*, vol. 15, no. 1, pp. 51–67, 2007.

[13] H. Veeraraghavan, S. Atev, N. Bird, P. Schrater, and N. Papanikolopoulos, "Driver activity monitoring through supervised and unsupervised learning," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pp. 895–900, Vienna, Austria, September 2005.

[14] C. H. Zhao, B. L. Zhang, J. He, and J. Lian, "Recognition of driving postures by contourlet transform and random forests," *IET Intelligent Transport Systems*, vol. 6, no. 2, pp. 161–168, 2012.

[15] C. H. Zhao, B. L. Zhang, X. Z. Zhang, S. Q. Zhao, and H. X. Li, "Recognition of driving postures by combined features and random subspace ensemble of multilayer perception classiffer," *Journal of Neural Computing and Applications*, vol. 22, no. 1, Supplement, pp. 175–184, 2000.

[16] C. Tran, A. Doshi, and M. M. Trivedi, "Modeling and prediction of driver behavior by foot gesture analysis," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 435–445, 2012.

[17] J. C. McCall and M. M. Trivedi, "Visual context capture and analysis for driver attention monitoring," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC '04)*, pp. 332–337, Washington, DC, USA, October 2004.

[18] K. Torkkola, N. Massey, and C. Wood, "Driver inattention detection through intelligent analysis of readily available sensors," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC '04)*, pp. 326–331, Washington, DC, USA, October 2004.

[19] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proceedings of the 14th IEEE International Intelligent Transportation Systems Conference (ITSC '11)*, pp. 1609–1615, Washington, DC, USA, October 2011.

[20] A. V. Desai and M. A. Haque, "Vigilance monitoring for operator safety: a simulation study on highway driving," *Journal of Safety Research*, vol. 37, no. 2, pp. 139–147, 2006.

[21] G. Rigas, Y. Goletsis, P. Bougia, and D. I. Fotiadis, "Towards driver's state recognition on real driving conditions," *International Journal of Vehicular Technology*, vol. 2011, Article ID 617210, 14 pages, 2011.

[22] M. H. Kutila, M. Jokela, T. Mäkinen, J. Viitanen, G. Markkula, and T. W. Victor, "Driver cognitive distraction detection: feature estimation and implementation," *Proceedings of the Institution of Mechanical Engineers D*, vol. 221, no. 9, pp. 1027–1040, 2007.

[23] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[24] Y. Wang, K. Huang, and T. Tan, "Human activity recognition based on R transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, Minneapolis, Minn, USA, June 2007.

[25] H. Chen, H. Chen, Y. Chen, and S. Lee, "Human action recognition using star skeleton," in *Proceedings of the International Workshop on Video Surveillance and Sensor Networks (VSSN '06)*, pp. 171–178, Santa Barbara, Calif, USA, October 2006.

[26] W. Liang and D. Suter, "Informative shape representations for human action recognition," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 1266–1269, Hong Kong, August 2006.

[27] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Proceedings of the International Conference on Computer Vision*, vol. 2, pp. 726–733, Nice, France, October 2003.

[28] A. R. Ahad, T. Ogata, J. K. Tan, H. S. Kim, and S. Ishikawa, "Motion recognition approach to solve overwriting in complex actions," in *Proceedings of the 8th IEEE International Conference on Automatic Face and Gesture Recognition (FG '08)*, pp. 1–6, Amsterdam, The Netherlands, September 2008.

[29] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 288–303, 2010.

[30] S. Danafar and N. Gheissari, "Action recognition for surveil-lanceapplications using optic ow and SVM," in *Proceedings of the Asian Conferenceon Computer Vision (ACCV '07)*, pp. 457–466, Tokyo, Japan, November 2007.

[31] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 432–439, Nice, France, October 2003.

[32] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg, "Local velocity-adapted motion events for spatio-temporal recognition," *Computer Vision and Image Understanding*, vol. 108, no. 3, pp. 207–229, 2007.

[33] S. Park and M. Trivedi, "Driver activity analysis for intelligent vehicles: issues and development framework," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 644–649, Las Vegas, Nev, USA, June 2005.

[34] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR '07)*, pp. 401–408, Amsterdam, The Netherlands, July 2007.

[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 886–893, San Diego, Calif, USA, June 2005.

[36] G. R. Bradski and J. Davis, "Motion segmentation and pose recognition with motion history gradients," in *Proceedings of the 5th IEEE Workshop on Applications of Computer Vision*, pp. 238–244, Palm Springs, Calif, USA, 2000.

[37] O. Masoud and N. Papanikolopoulos, "A method for human action recognition," *Image and Vision Computing*, vol. 21, no. 8, pp. 729–743, 2003.

[38] H. Yi, D. Rajan, and L.-T. Chia, "A new motion histogram to index motion content in video segments," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1221–1231, 2005.

[39] J. Flusser, T. Suk, and B. Zitov, *Moments and Moment Invariants in Pattern Recognition*, John Wiley & Sons, Chichester, UK, 2009.

[40] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 2169–2178, New York, NY, USA, June 2006.

[41] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classiffcation*, Wiley-Interscience, 2nd edition, 2000.

[42] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.