Vision-based Driver Behaviour Analysis

Chao Yan

School of Electrical Engineering, Electronics & Computer Science
University of Liverpool

This dissertation is submitted for the degree of
Doctor of Philosophy

March 2016

I would like to dedicate this thesis to my loving parents ...

Acknowledgements

I would like to express my deep gratitude to Dr. Bailing Zhang and Prof. Frans Coenen, my supervisors, for their insightful suggestions, valuable help and constant support during my phD study.

My grateful thanks are also extended to Prof. Yong Yue, Dr. Wenjin Lv and Dr. Ka lok Man for their useful critiques and advice.

I am particularly grateful for funding by Xi'an Jiaotong Liverpool University Graduate Scholarship and supporting by Xi'an Jiaotong Liverpool University Campus.

I would like to offer my special thanks to my colleagues, Rongqiang Qian, Zhao Wang and Yizhang Xia, for fruitful suggestions and being my great friends.

Finally, I most gratefully acknowledge my mother and my wife for their encouragement and love.

Abstract

With the ever-growing traffic density, the number of road accidents is anticipated to further increase. Finding solutions to reduce road accidents and to improve traffic safety has become a top-priority for many government agencies and automobile manufactures alike. It has become imperative to the development of Advance Driver Assistance Systems (ADAS) which is able to continuously monitor, not just the surrounding environment and vehicle state, but also driver behaviours.

Dangerous driver behaviour including distraction and fatigue, has long been recognized as the main contributing factor in traffic accidents. This thesis mainly presents contributing research on vision based driver distraction and fatigue analysis and pedestrian gait identification, which can be summarised in four parts as follows.

First, the driver distraction activities including operating the shift lever, talking on a cell phone, eating, and smoking, are explored to be recognised under the framework of human action recognition. Computer vision technologies including motion history image and the pyramid histogram of oriented gradients, are applied to extracting discriminate feature for recognition. Moreover, A hierarchal classification system which considers different sets of features at different levels, is designed to improve the performance than conventional "flat" classification.

Second, to solve the effectiveness problem in poor illuminations and realistic road conditions and to improve the performance, a posture based driver distraction recognition system is extended, which applies convolutional neural network (CNN) to automatically learn and predict pre-defined driving postures. The main idea is to monitor driver arm patterns with discriminative information extracted to predict distracting driver postures.

Third, supposing to analysis driver fatigue and distraction through driver's eye, mouth and ear, a commercial deep learning facial landmark locating toolbox (Face++ Research Toolkit) is evaluated in localizing the region of driver's eye, mouth and ear and is demonstrated robust performance under the effect of illumination variation and occlusion in real driving condition. Then, semantic features for recognising different statuses of eye, mouth and ear on image patches, are learned via CNNs, which requires minimal domain knowledge of the problem.

Finally, works on pedestrian subject identification using convolutional neural networks(CNNs) and multi-task learning model(MTL), is presented additionally. Gait identification is strongly motivated by the demands of security that require automatically identifying person at a distance. This be particularly relevant with respect to police/detective vehicle that is tracking criminal.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

Unsafe and dangerous driving accounts for the death of more than one million lives and over 50 million serious injuries worldwide each year [12]. The U.S. National Highway Traffic Safety Administration(NHTSA) data indicates that 1.6 million nonfatal injuries, and 40 thousands fatalities, resulted from traffic accidents in 2012, with up to 80% of them due to driver inattention [13]. In Europe, up to 20% of accidents are caused by driver drowsiness. Moreover, in it [14] was estimated that the worldwide vehicle population would increase to 1.2 billion in 2014. With the ever-growing traffic density, the number of road accidents is anticipated to further increase. Finding solutions to reduce road accidents and improve traffic safety has become a top-priority for many government agencies and automobile manufactures alike.

Statistics show that one of the leading causes of fatal or injury-causing traffic accident is the diminishment of the driver vigilance level. The main contributing factors may either be fatigue or distraction. Scientific research has been conducted to estimate the level of sleep deprivation in relation to traffic accidents[15, 16]. The development of Intelligent Driver Assistance Systems(IDAS) ,that continuously monitor, not just the surrounding environment and vehicle state, but also driver behaviours, have attracted increasing worldwide attention[17]. IDAS are seen to be particularly relevant with respect to long-distance drivers as they often drive alone. Usage if IDAS that 'flag-up' important information outside of a vehicle, such as driving lane indicators and traffic signs, have been shown to increase driver alertness[18, 19]. However, automatic detection and warning of driver fatigue and distraction level is considered to be of equal importance with respect to road accident prevention. Other than for reasons of road safety enhancement, there are also commercial reasons for fitting driver alertness monitoring systems, particularly with respect to truck and bus fleet managers.

Computer-based driver assistance systems is one of the most critical technology for intelligent vehicles, which can be traced to the earlier efforts dealing with autonomous mobile robots and autonomous driving [20–22]. Such efforts helped to demonstrate the power of camera-based systems to support real-time control of vehicles. In [23] and [24], Bertozzi et al. give a comprehensive survey of the use of computer vision in intelligent vehicles. Approaches for lane, pedestrian, and obstacle detection are described and analyzed. The trend started emerging in the late 1990s, where research in computer vision focused on enhancement of the safety of automobiles [25, 26]. Already, camera-based modules with safety-oriented features such as "back-up" (or reverse) viewing, lane-departure warning, and blind-spot detection are offered in commercial vehicles. It was realized that in addition to monitoring the surroundings, the monitoring of the driver state is also important for improving safety.

Consequently, this thesis aims to develop a vision system able to analysis the driver state including activities, postures, and fatigue. In addition, pedestrian gait identification, as a relevant application of intelligent vehicle, is also investigated. The structure of this thesis is summarised in the following Section 1.2.

1.2 Thesis structure

The thesis structure is summarized in terms of chapter relationships as shown in Fig. 1.1.

Chapter 2 describes the research scope. The readers can therefore understand the research fields and disciplines that our work can be traced from. Afterwards, three particular topics corresponding to our work are reviewed in subsequence including driver distraction analysis, driver distraction analysis and pedestrian gait analysis.

Chapter 3 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.

Chapter 4 presents a novel system for video-based driving behaviour recognition. The fundamental idea is to monitor driver hand movements and to use these as predictors for safe/unsafe driving behaviour. The proposed method utilises hierarchal classification and treats driving behaviour in terms of a spatial-temporal reference framework as opposed to a static image. The Approach was verified using the Southeast University Driving-Posture Dataset, a dataset comprised of video clips covering aspects of driving such as: normal

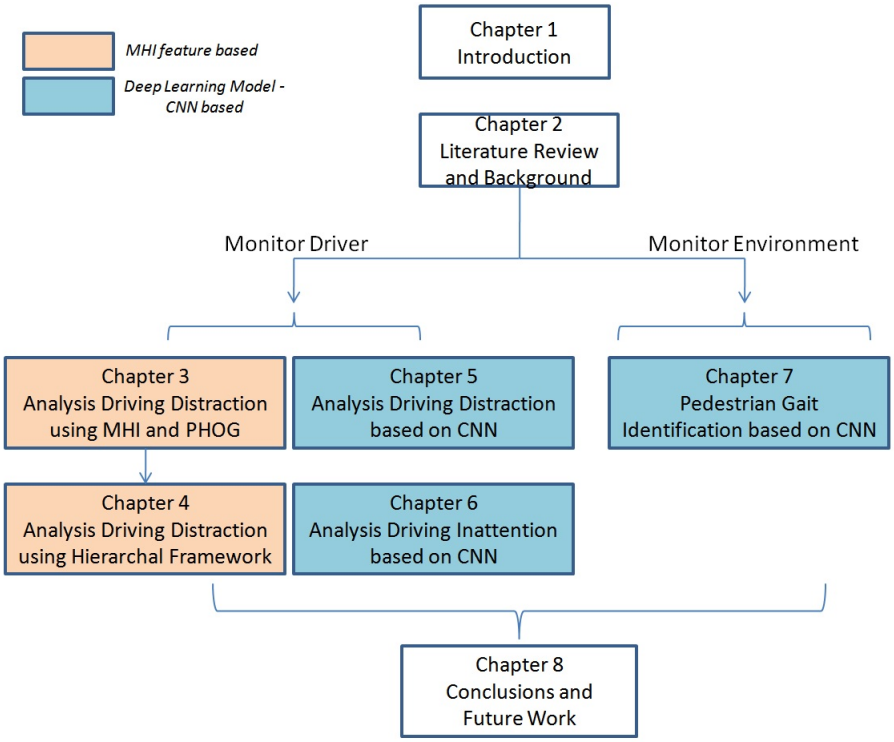


Fig. 1.1 Overview of the Thesis Structure

driving, responding to a cell phone call, eating and smoking. After pre-processing for illumination variations and motion sequence segmentation, eight classes of behaviour were identified.

Chapter 5 presents a novel system which applies convolutional neural network (CNN) to automatically learn and predict pre-defined driving postures. The main idea is to monitor driver hand position with discriminative information extracted to predict safe/unsafe driving posture. In comparison to previous approaches, convolutional neural networks can automatically learn discriminative features directly from raw images. In our works, a CNN model was first pre-trained by an unsupervised feature learning method called sparse filtering, and subsequently fine-tuned with classification.

Chapter 6 presents a novel system which applies convolutional neural network to automatically learn and predict the state of driver's eye, mouth and ear. The main idea is to predict driver fatigue and distraction by analysing the state of eye, mouth and ear. In order to robust propose the region of of eye, mouth and ear, Face++ Research Toolkit [27] are applied to localize the facial landmark [28] on the driver's face. Then the context of these regions are trained through a CNN model. The Approach was verified using self-specified Driving Dataset, which comprised of video clips covering behaviours, including normal driving, responding to a cell phone call, eating, falling asleep.

Chapter 7 proposes a robust and effective gait recognition approach using convolutional neural networks(CNNs) and multi-task learning model(MTL). Firstly, Gait Energy Image(GEI) was extracted from each walking period as the low level input for the CNNs. Multi-task CNN model is trained through back-propagation using a joint loss of each task. Then, the high-level features for multiple tasks could be extracted simultaneously with the given input.

Chapter 8 draws conclusions of this thesis and suggestions for further work.

1.3 Thesis contribution

The major contributions proposed in this thesis are as follows:

1. Many published works on drivers' posture based on static images from drivers' action sequence 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 [29] which is based on the combination of basic movements, we propose 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. (Chapter 3)

2. The proposal of a global grid-based representation for the driving actions, which is a combination of the motion history image (MHI) [30] and pyramid histogram of oriented gradients (POHG) [31], and the application of random forest classifier (RF) for the driving actions recognition. (Chapter 3)

3. A two stage intensity normalization preprocessing technique to minimize the influence from illumination variation. The first stage comprised a moving average method that smoothed the intensity variation caused by periodic lighting change. The second stage comprised application of the three frame difference method[32] to detect motion. For the task of motion detection and segmentation in video, it was found that the proposed two-stage pre-processing technique performed well in context of compensating for noise and illumination variation in video data. (Chapter 4)

4. A hierarchal classification system for driving behaviour recognition which considers different sets of features at different levels. Hierarchical classification is specifically intended for data where the features of interest can be arranged in a hierarchical

manner. As such it offers advantages in terms of learning and representation in comparison to attempts to use "flat" classification techniques for the purposed of classifying hierarchical data[33]. These efficiency gains are realised because only a subset of the complete set of available features is considered at each node in the hierarchy. Hierarchical classification schemes have been applied in many areas [34–36]. However, it should be noted here that, to the best knowledge of the authors, they have not been applied to driving behaviour recognition. (Chapter 4)

5. To recognise driving posture, eye state, mouth state, ear state and pedestrian subject, we proposed to build a deep convolutional neural network in which trainable filters and local neighborhood pooling operations are applied alternatively to automatically explore salient features. Using CNN to learn rich features from the training set is more generic and requires minimal domain knowledge of the problem compared to hand crafted feature in previous approaches. (Chapter 5, Chapter 6, and Chapter 7)

6. We proposed to applied Face++ Research Toolkit [27] to localize the facial landmark [28] on the driver's face, which is used to propose the region of eye, mouth and ear. It is much robust under the effect of illumination variation and occlusion in real driving condition than previous approaches. (Chapter 6)

7. Multi-task model is a machine learning approach that jointly trains one task together with other related tasks at the same time sharing the same lower feature layers, which uses the commonality among tasks and therefore learns shared feature representation benefits all tasks. Since the difficulties in human gait identification are mainly caused by the multi factor's effects, it is very natural to use multi-tasks learning to simultaneously identify the multiple attributes of the gait. Thus, the approach in this chapter aims to investigates a convolutional neural network model for identifying the human gait while simultaneously predicting other human attributes at same time. To the best of our knowledge, this is the first approach that using MTL to investigate how human gait can be identified together with other auxiliary tasks. (Chapter 7)

1.4 Publication

Parts of the contributions presented in the thesis and other research outputs have been published or are under review:

Journal Publications

1. **Yan, C.**, Coenen, F. and Zhang, B., "Driving Posture Recognition by Convolutional Neural Networks," *IET Computer Vision* (Accepted)

2. **Yan, C.**, Coenen, F. and Zhang, B., "Driving Posture Recognition by Joint Application of Motion History Image and Pyramid Histogram of Oriented Gradients," *International Journal of Vehicular Technology*, vol. 2014, Article ID 719413, 11 pages, 2014.

3. **Yan, C.**, Coenen, F., Yue, Y., Yang, X. and Zhang, B., "Video-Based Classification of Driver Behaviour using a Hierarchal Classification System with Multiple Features," *International Journal of Pattern Recognition and Artificial Intelligence* (Under review)

4. **Yan, C.**, Zhang, B., Coenen, F., Wang, Z. and Yang, X., "Multi-attributes Gait Identification by Convolutional Neural Network," *Journal of Computer Science and Technology* (Under review)

Conference Publications

1. **Yan, C.**, Coenen, F. and Zhang, B., "Driving Posture Recognition by Convolutional Neural Networks," *Proceedings of the 11th International Conference on Natural Computation* (ICNC'15) (To appear)

2. **Yan, C.**, Coenen, F. and Zhang, B., "Multi-attributes Gait Identification by Convolutional Neural Networks," *Proceedings of 8th International Congress on Image and Signal Processing* (CISP'15) (Accepted)

3. **Yan, C.**, Coenen, F. and Zhang, B., "Driving posture recognition by a hierarchal classification system with multiple features," *Proceedings of 7th International Congress on Image and Signal Processing* (CISP'14), pp.83-88, Dalian, October, 2014

4. **Yan, C.**, Coenen, F. and Zhang, B., "Driving posture recognition by joint application of motion history image and pyramid histogram of oriented gradients," *Advanced Materials Research*, vol.846-847, pp.1102-1105, Xi'an, 2014

5. **Yan, C.**, Coenen, F. and Zhang, B., "Recognizing Driver Inattention by Convolutional Neural Network," *Proceedings of 8th International Congress on Image and Signal Processing* (CISP'15) (Under review)

Chapter 2

Literature Review and Background

In this chapter, the research scope is firstly defined in Section 2.1. The readers can therefore understand the research fields and disciplines that our work can be traced from. Second, three particular topics corresponding to our work are reviewed in subsequence. Specifically, Section 2.4 reviews previously works on driver distraction analysis, Section 2.5 reviews previously works on driver distraction analysis and Section 2.6 reviews previously works on pedestrian gait analysis.

2.1 Scope of the Research

Due to the improving vehicle population [12] and road accidents, finding solutions to reduce road accidents and improve traffic safety has become a top-priority for many government agencies and automobile manufactures alike. The development of Advance Driver Assistance Systems(ADAS) ,that continuously monitor, not just the surrounding environment and vehicle state, but also driver behaviours, have attracted increasing worldwide attention[17].

Wikipedia gives the definition of ADAS as follows. Advanced Driver Assistance Systems, or ADAS, are systems to help the driver in the driving process. When designed with a safe Human-Machine Interface, they should increase car safety and more generally road safety [37]. The research presented in this thesis is under the scope of investigations into the roles of computer-vision technology in developing safer vehicles, and can be therefore considered belonging to a part of Advance Driver Assistance Systems(ADAS).

IEEE Intelligent Transportation Systems Society gives the definition of Intelligent Transportation Systems (ITS) as those utilizing synergistic technologies and systems engineering concepts to develop and improve transportation systems of all kinds [38]. Wikipedia defines ITS as advanced applications which, without embodying intelligence as such, aim to provide innovative services relating to different modes of transport and traffic management and enable

various users to be better informed and make safer, more coordinated, and 'smarter' use of transport networks [39]. As a result, ADAS can be further classified as a subset of Intelligent Transportation Systems(ITS).

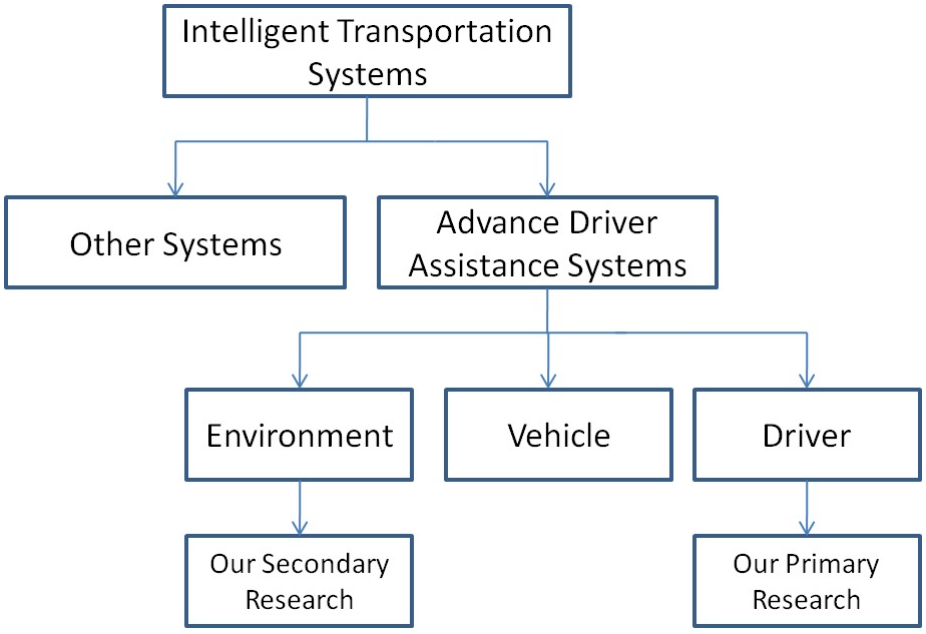


Fig. 2.1 Scope of the Research

As above, the scope of our research can be illustrated in Fig. 2.1. In the following, Intelligent Transportation Systems and Advance Driver Assistance Systems will be briefly introduced in Section 2.2 and Section 2.3, respectively.

2.2 Intelligent Transportation Systems

Intelligent Transportation Systems (ITS) are advanced technologies which aim to provide innovative services relating to different modes of transport and traffic management. It enables various users to be better informed and make safer, more coordinated, and 'smarter' use of transport networks. It is playing a critical role in virtually all facets of modern life. In the meanwhile, significant challenges remain to further improve the efficiency and safety of the current transportation systems of all kinds and develop value-added applications closely tied into such systems [40].

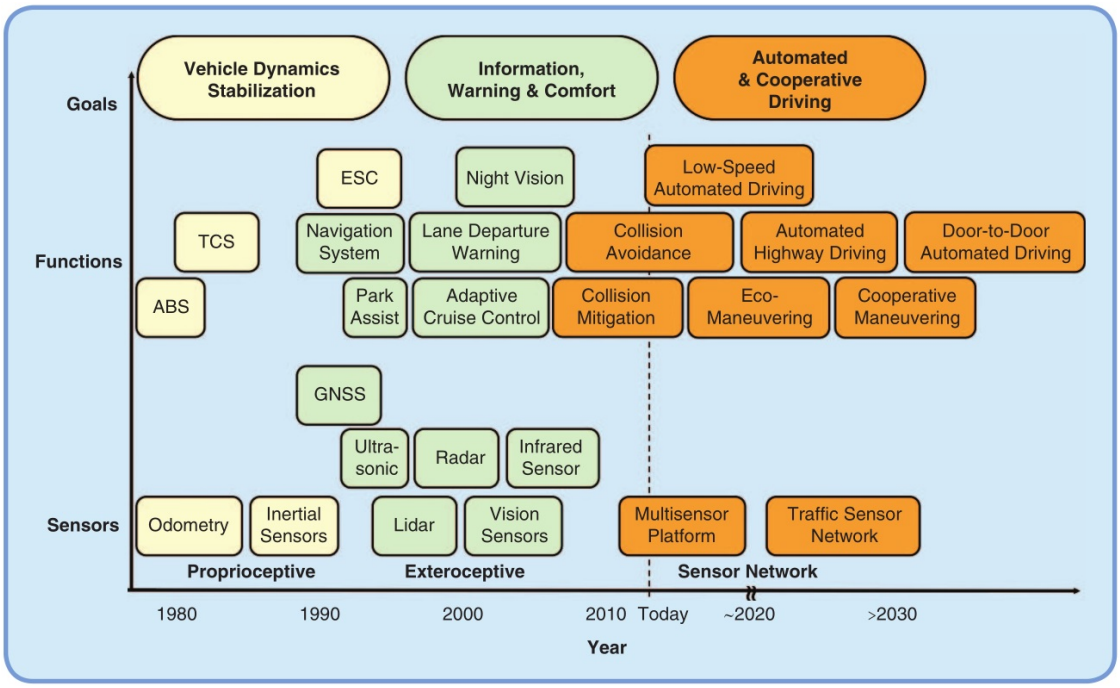


Fig. 2.2 Past current and potential future evolution of Driver Assistance Systems

2.3 Advance Driver Assistance Systems

Mobility is a fundamental desire of a vehicle. Virtually any society strives for safe and efficient mobility, although its technical implementation significantly differs and depending on different culture and degree of industrialization [41]. This sub-section introduces past, current and a potential future evolution of Driver Assistance Systems(DAS). It is sketched in Fig. 2.2 from a technological point of view [41]. From the figure, past, current, and future, the three generations of Driver Assistance Systems are highlighted in three different colors, each of which contains different sensors, functions and goals. A brief description of the three generations of DAS are as follows.

2.3.1 First/Past Generation DAS

Early DAS were based on proprioceptive sensors, i.e. sensors measuring the internal status of the vehicle, such as wheel velocity, acceleration, or rotational velocity. These enable the control of vehicle dynamics with the goal of following the trajectory requested by the driver in the best possible way.

With respect to the safe purpose. The vehicle-based safety systems are typically viewed as one of the two kinds [17]. The first one is termed as "Passive". Examples of these are seat

belts, airbags, collapsible steering columns, and shatter-resistant windshields. The purpose here is to passively minimize the severity of injuries sustained in case of accidents.

The second kind is "Active", which is supposed to prevent vehicular accidents actively rather than reduce the severity of injuries passively. One of the first active assistance systems based on proprioceptive sensors was the Anti-lock Braking System (ABS), with serial production from 1978 (Bosch). A Traction Control System (TCS) later augmented the system. Years later in 1995, the introduction of additional dynamic driving controls, such as Electronic Stability Control (ESC), marked a further milestone in assistance development [42].

In terms of road safety, studies have shown that dynamic driving controls are one of the most efficient safety system for passengers, outmatched only by passive DAS such as the seatbelt [43, 44]. With the public recognition of the safety potential of dynamic driving control systems, the frequency of implementation for such active DAS increased significantly, and they have, in consequence, saved several thousands of lives.

2.3.2 Second/Current Generation DAS

Exteroceptive sensors including ultrasonic, radar, lidar or video sensors and to some extent Global Navigation Satellite System (GNSS) as illustrated in Fig. 2.2, are able to provide information about the road ahead and the presence as well as the driving status of other traffic participants or the vehicle's position in the world.

With the development of exteroceptive sensors, the second generation of driver assistance functions first introduced around 1990 based on exteroceptive sensors focuses on providing information and warnings to the driver, and on enhancing driving comfort. Some selected DAS technologies that have been used in commercial application, are listed as follows.

1. **Navigation Technology** For a given GEI belonging to the class of *only shift gear related*, find its corresponding original frame sequence.

2. **Parking Assistance Systems** Transform the original sequence into a binary image sequence based on hand skin region segmentation proposed in previous subsection.

3. **Adaptive Cruise Control (ACC)** Calculate the frame differencing image sequence from the binary image sequence.

4. **Forward Collision Prevention Systems** For each frame in the sequence, project its binary frame differencing image onto the vertical-axis and get the projection vector.

Research towards latest class of DAS, aims to intelligently analysis the information from exteroceptive sensors and to enhancing safe and smooth operation of a vehicle in traffic.

Trivedi et al. [17] proposed an active-safely system (EVD) framework as shown in Fig. 2.3, which is able to continuously monitor, not just the surrounding environment (E) and vehicle state (V), but also driver behaviours (D).

The front end of an active-safety system is a sensing subsystem, which needs to provide an accurate description of the dynamic state of the EVD system. The second important subsystem is an analysis subsystem which needs to analyze the EVD dynamic state using a model-based approach to compute some sort of a measure of safety underlying that particular EVD state. If this measure falls under a predefined threshold of margin of safety, then the analysis module needs to direct the active- safety control unit to initiate a corrective course of action so that the vehicle can always operate within the margins of an accident-free safety zone. There are some very challenging problems involved in each of the above three subsystems of an active-safety system. One of the most challenging problems is the computer-vision-based analysis of the context captured from vision-sensor, which will be introduced as follows.

Computer-Vision-based Driver Assistance Systems

Recognition of computer vision as a critical technology for intelligent vehicles can be traced to the earlier efforts dealing with autonomous mobile robots and autonomous driving [20–22]. Such efforts helped to demonstrate the power of camera-based systems to support real-time control of vehicles. In [23] and [24], Bertozzi et al. give a comprehensive survey of the use of computer vision in intelligent vehicles. Approaches for lane, pedestrian, and obstacle detection are described and analyzed. The new trend started emerging in the late 1990s, where research in computer vision focused on enhancement of the safety of automobiles [25, 26]. Already, camera-based modules with safety-oriented features such as "back-up" (or reverse) viewing, lane-departure warning, and blind-spot detection are offered in commercial vehicles. It was realized that in addition to monitoring the surroundings, the monitoring of the driver state is also important for improving safety.

Fig. 2.4 illustrated selected computer-vision-based approaches, which provide the essential driver-assistance system. The basic objective is the development of novel vision systems and appropriate algorithms for enhancement of safety.

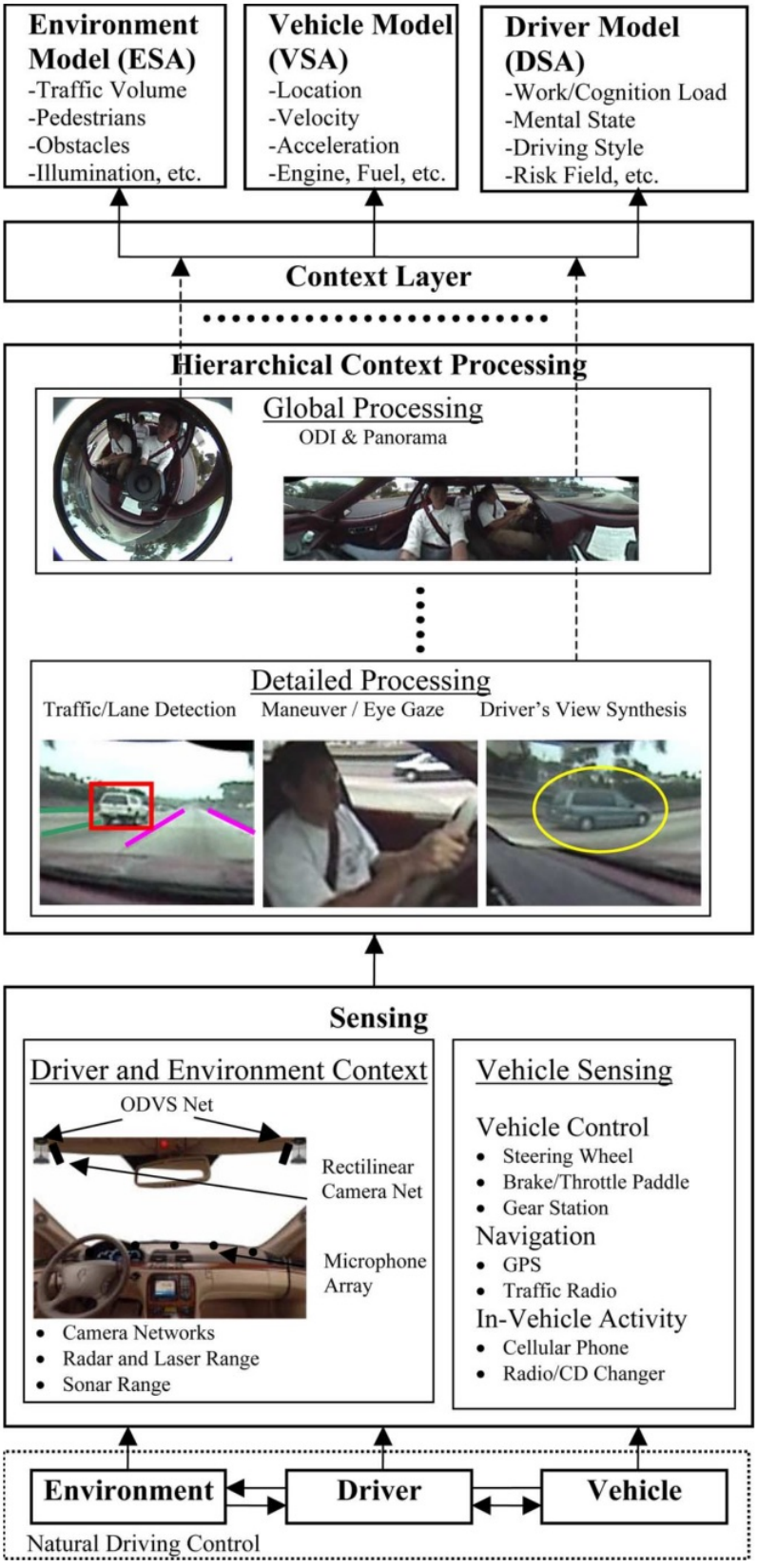


Fig. 2.3 Framework for multi-functional "active" dynamic context-capture system.

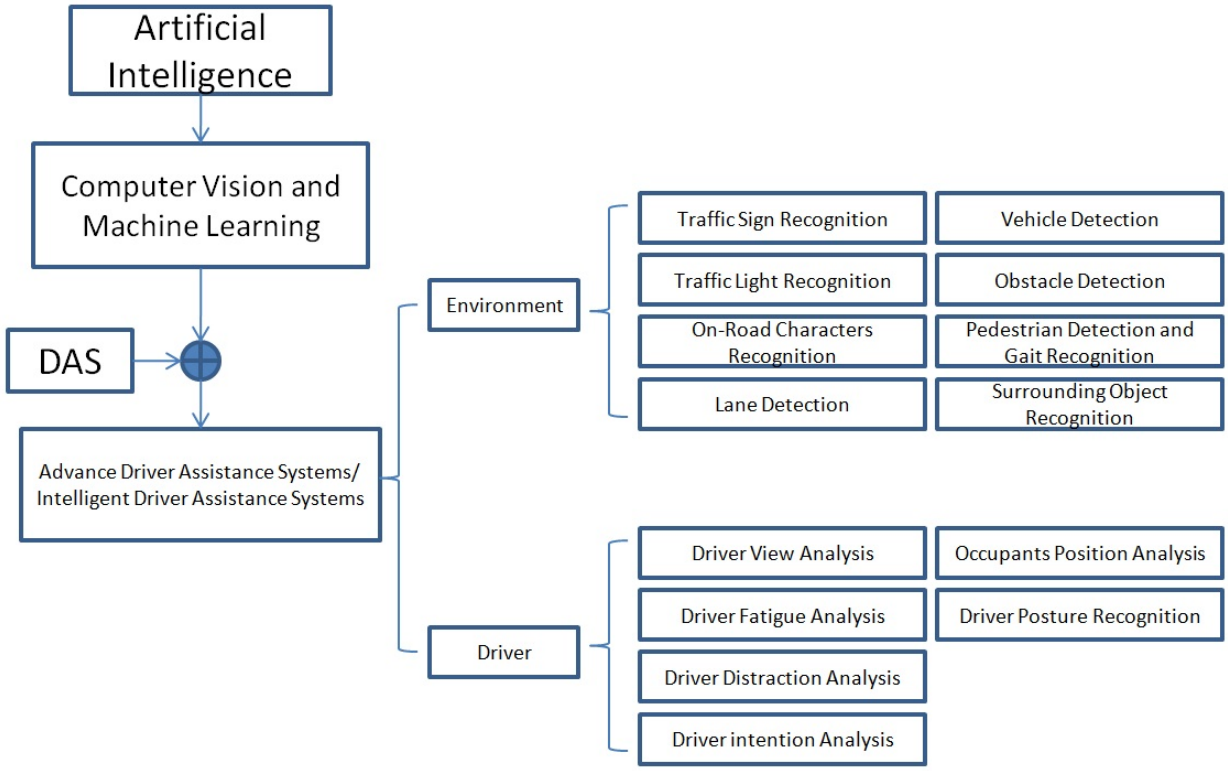


Fig. 2.4 Selected computer-vision-based approaches for ADAS.

2.3.3 Future Generation DAS

2.4 Review on Driver Distraction Analysis

Driver distractions are the leading cause of most vehicle crashes throughout the world. Distraction means anything that diverts the driver's attention from the primary tasks of navigating the vehicle and responding to critical events despite the presence of obstacles or other people. A technique report [45] from AAA Foundation for Traffic Safety, summarized thirteen types of potentially distraction activities including: eating or drinking, outside person, object or event, talking or listening on a cellular phone, dialing a cellular phone, using in-vehicle-technologies, and so on. Alternatively, in [46], driver distractions are classified into the following three categories:

1. visual distraction (e.g., looking away from the roadway);
2. cognitive distraction (e.g., your mind off the road);
3. manual distraction (e.g., your hands off the wheel, responding to a ringing cell phone, and manually adjusting the radio volume);

2.4.1 Visual Distraction

With respect to the first category of distraction (Visual distraction). It is reliable to monitor the driver head pose and gaze direction. Head pose and gaze direction of driver can be measured by applying computer vision techniques properly. Different articles and studies were examined and several key factors including accuracy, advantages, disadvantages,limitations, methods, classifiers and system implementation for this technique are given in Table II. Wahlstrom et al. [47] proposed a mechanism for locating the eyes and pupils in a facial image using skin colour area and then estimating the gaze direction from the relative positions of the eyes and pupils. Of course this approach will not succeed if the driver's head is turned away from the camera. In order to minimize the influence of various illumination and background interferences, infrared cameras were used in the work presented in [48] to estimate the driver face direction, again based on skin colour area analysis. To improve the performance of head pose estimation, in the presence of dramatic changes in illumination, the use of isophote features was introduced in [49]. In [50], video frames were represented using the Fisher face approach and then classified using the nearest neighbor and neural network models. However, the system is driver dependent, which makes it unrealistic in many situations. An integrated system for monitoring driver awareness, based on head pose estimation, was presented in [51], which include head detection and tracking. A comparative study of the influence that eye gaze and head movement dynamics have on (i)driver behaviour and (ii)intent prediction with respect of lane change manouvers was presented in [18].

2.4.2 Cognitive Distraction

The second category of distraction, as noted above, is cognitive distraction behaviour. Driver fatigue is a leading cause to cognitive distraction which will be reviewed in the following section. However, it is very difficult to monitore cognitive distraction behaviour which does not have any physical behavior if using computer vision techniques. For example, brainstorming a work-related affair may takes your mind off the road without any change on driver's facial expression or body movement.

2.4.3 Manual Distraction

The research on the third categories of distraction, directed at vision-based automatic driver manually behaviour prediction, centers on the characterization of driver body posture, including arms, hands and feet. For example, a variant of the Iterative Closest Point (ICP) registration algorithm was proposed in [52] to estimate the location and orientation of a

driver's limbs, with visual information provided by an infrared Time-of-Flight camera. Driver posture dynamics in 3D was investigated in [53] using a vision-based system. In [54] a camera array system was proposed to track important driver body parts and to analyze driver activities such as steering movements. In [55] an agglomerative clustering and Bayesian eigen-image approach were applied to represent and recognize predefined safe/unsafe driving activities, such as talking on a cellular phone and eating. A modified Histogram of Oriented Gradients(HOG) feature description mechanism coupled with a support vector machine classifier was applied in [56] to discriminate which of the front-row seat occupants was accessing "infotainment" controls. To investigate "pedal error phenomenon" Tran et al. [57] developed a vision based system for driver foot behaviour analysis which featured an optical flow based foot tracking and a Hidden Markov Model (HMM) based approach to characterize temporal foot behaviour.

2.5 Review on Driver Fatigue Analysis

Driver fatigue is another major factor that may cause traffic accident on the road. The term fatigue refers to a combination of symptoms such as impaired performance and a subjective feeling of drowsiness [4]. Even with the intensive research that has been performed, the term fatigue still does not have a universally accepted definition [5]. Thus, it is difficult to determine the level of fatigue-related accidents. However, studies show that 25%-30% of driving accidents are fatigue related [6].

When a driver is fatigued, certain physical and physiological phenomena can be observed, including changes in brain waves or EEG, eye activity, facial expressions, head nodding, body sagging posture, heart rate, pulse, skin electric potential, gripping force on the steering wheel, and other changes in body activities. Previous designs for the detection of fatigue are based on the analysis of these driver physical and physiological phenomena, and can be broadly divided into two categories: (i)visual features based and (ii)non-visual features based.

2.5.1 Non-visual Features Based

Techniques based on non-visual features are generally intrusive and can be divided into two categories: driver physiological analysis and vehicle parameter analysis.

For the first one, driver fatigue is analyzed through vehicle behaviour including movement of steering wheel, pressure on the acceleration pedal, speed, deviations from lane position, response time against an obstacle braking, and etc. The main limitations of these approaches

[58, 59] include their dependence on the shape of the road, the vehicle performance and the manner of driving.

For the second approach, driver fatigue monitoring is based on the analysis of physiological and biomedical signals such as heart rate, brain activity, temperature, vascular activity, muscular activity, electroencephalograph(EEG), electrocardiogram (ECG), electrooculography (EOG), and surface electromyogram (sEMG). These signals are collected through electrodes in contact with the skin of the human body. They are widely accepted as good indicators of the transition between wakefulness and sleep, as well as between the different sleep stages. It is often referred to as the gold standard. Svensson [60] proposed objective sleepiness scoring (OSS), which is derived from EEG signals, as the ground truth for validating other fatigue detection algorithms. However, these methods [61] rely on contactable sensors which decrease user experience and increase hardware cost.

2.5.2 Visual Features Based

Techniques using visual features take advantage of computer vision approaches for the detection of fatigue. Exploiting visual features focuses on extracting facial features like face, eyes and mouth. Analyzing the state of eyes and mouth can provide observable cues for the detection process. Mainly, techniques using visual features can be divided into four categories: (i)eye state analysis, (ii)eye blinking analysis, (iv)mouth and yawning analysis and (iv)facial expression analysis. In the following subsections, the mostly used techniques with these four categories based on visual features are explained in detail.

Eye state analysis

Eye state analysis is the most common and straightforward technique for detecting driver fatigue. The systems applying this technique focus on the states of eyes [62–64]. In such solutions, the system warns the driver by generating an alarm, if the driver closes his/her eye(s) for a particular time. Some available systems based on this technique use a database where both closed and open templates of eye are stored [65]. In addition to non-adaptive systems, there are some adaptive solutions where the open and close eye templates of a related driver are exploited. Customized template matching technique on a frame-by-frame basis is used to detect the state of the two eyes [66].

On the other hand, eye state analysis is computationally intensive. In addition to that, there are some limitations, such as lighting conditions and sunglasses that affect the accuracy of the template matching technique. Another disadvantage of the template matching technique

is that it would fail if the templates were distorted due to the image processing [66]. Because of the above reasons this technique is not sufficient enough to detect the driver's drowsiness.

Eye blinking analysis

As the eyes of drivers can provide observable information about fatigue level, many researches and studies exploit eye blinking frequency for drowsiness detection [67–69]. Those systems are based on monitoring the changes in the eye blinking duration. According to the study in [70],the eye blinking duration is the most reliable parameter for the detection of the drowsiness level. Since whenever a driver is tired or feels sleepy, his/her eye's blinking frequency changes and the eyelid closure duration starts involuntarily to prolong. To be more specific, when the driver is alert, his/her eye blinking frequency is low and his/her eyelid closure duration will be slower. However, when the driver is exhausted, his/her eye blinking frequency gets higher (more closed-eye images) and his/her eyelid closure duration will be shorter.

Mouth and yawning analysis

Most of the existing works, which exploit eye state features, suffer from the presence of sunglasses [71].With regard to the driver's mouth state it is also possible to determine driver's sleepy level with yawning measurement [72–77]. Yawning is an involuntary intake of breath through a wide-open mouth; usually triggered by fatigue or boredom. This technique is also one of the non-intrusive techniques for detecting driver fatigue by applying computer vision. In this approach, detecting drowsiness involves two main phases to analyze the changes in facial expressions properly that imply drowsiness. First, the driver's face is detected by using cascade classifiers and tracked in the series of frame shots taken by the camera. After locating the driver's face, the next step is to detect and track the location of the mouth.For mouth detection the researchers have used the face detection algorithm proposed by Paul Viola and Michael J. Jones [78]. Afterwards, yawning has been analyzed to determine the level of the drowsiness [76]. This is presumed to be modeled with a large vertical mouth opening and changes in the driver's mouth contour. Mouth opens wide and the distance between its counters gets larger.

Facial expression analysis

Contrary to exploit specific regions of face such as eye and mouth, this technique broadly analyzes more than one face region. ANN (Artificial Neural Network) is usually used for optimization of driver drowsiness detection [79]. In addition to that, it provides a different

way to approach such a control problem, this technology is not difficult to apply and the results are usually quite surprising and pleasing. However,there are limited researches applied ANN to detect driver fatigue [80, 81].

2.6 Review on Pedestrian Gait Analysis

Biometrics is a study on automatically recognising people using physiological or behavioral characteristic. Gait, as one of the most distinctive behavioral biometrics, exploits the uniqueness of walking to perform identification without interfering with the subject's activity[82, 83]. Human gait recognition algorithms can be roughly divided into two categories [84, 85]: (i)model-based, (ii)model-free approaches. The following two subsections are frow the paper [86].

2.6.1 Model-based Approaches

Model-based methods generally aim to model kinematics of human joints in order to measure physical gait parameters such as trajectories, limb lengths, and angular speeds. Gait signatures derived from these model parameters are employed for identification and recognition of an individual. It is evident that model-based approaches are view-invariant and scale-independent, which however suffer from accurately locating the joints' position due to the highly flexible structure of the nonrigid human body [87, 88]. Another disadvantage of the model-base approach is its large computation and relatively high time costs due to parameters calculations.

Primary model-based approaches employ static structure parameters of body as recognition features. BenAbdelkader et al. [89] present structural stride parameters consisting of stride and cadence. The cadence is estimated via the walking periodicity, and the stride length is calculated by the ration of travelled distance and walking steps. Bobick and Johnson [90] calculate four distances of human bodies, namely the distance between the head and foot, the distance between the head and pelvis, the distance between the foot and pelvis, and the distance between the left foot and right foot, as shown in Fig. 1. They use the four distances to form two groups of static body parameters and reveal that the second set of parameters are more view-invariant comparing to the first set of body parameters.

More recently, Yoo and Hwang [91] extract nine coordinates from the human body contours based on human anatomical knowledge to construct a 2D stick figure. Unlike some model-based approaches that utilize static structure parameters, Tanawongsuwan and Bobick [92] focus on the trajectories of joint angle from motion capture data. The joint angle

trajectories are computed by estimating the offsets between the 3D marker and joints. Yam et al. [93] construct a structure and motion model of legs to analyze walking as well as running using biomechanics of human and pendular motion. A comparative higher recognition currency of running demonstrates that running may be more reliable for human identification due to more different gait pattern. Additionally, based on comprehensively analyzing the characteristics and description of human gait, Cunado et al. [94] implemented Velocity Hough transform (VHT) [95] to extract the structure model of the thighs and the motion model of the thighs. It is reported that the VHT achieved good performance of median noise immunity.

Some other methods model human body parts separately. In Wang et al. [96]'s work, human body is modeled as fourteen rigid parts connected to one another at the joints. The whole model has forty-eight degrees of freedoms (DOFs). The tracking results, namely joint-angle trajectories signals, are considered as gait dynamics for identification and verification. They also obtain static information of body based on Procrustes shape analysis of the change of moving silhouettes, which can be independently or combinatively applied to improve the recognition. More recently, Boulgouris and Chi [97]separate human body into different components and combine the result obtained from different body parts to form a common distance metric. Based on the study of each part's contribution to the recognition performance, the recognition rate is improved by using the most contributing parts. In addition, Li et al. [98] divide the average silhouettes over a gait cycle into seven different parts and summarize the impact of each part on gait recognition.

2.6.2 Model-free Approaches

In the contrary to model-based approaches, without explicit modeling of human body structure, the mode-free methods typically analyze gait sequences for employing a compact representation to characterize the motion patterns of the human body. Model-free approaches are insensitive to the quality of silhouettes and have the advantage of low computational costs comparing to model-based approaches. However, they are usually not robust to viewpoints and scale.

The baseline algorithm proposed by Sarkar et al. [99] uses the silhouettes themselves as features, which are scaled and aligned before used. While the gait signature in the baseline algorithm is a sequence of gait silhouettes, Bobick and Davis [30] propose the motion-energy image (MEI) and motion-history image (MHI) to convert the temporal sequence of silhouettes to a 2D signal template. Han and Bhanu [100] employ the idea of MEI and propose the Gait Energy Image (GEI) for individual recognition. The left seven images in each row are silhouettes of walking sequences and the rightmost image is the corresponding gait

energy image. GEI converts the spatial-temporal information during one walking cycle into a single 2D gait template, which avoids matching features in temporal sequences. GEI is comparatively robust to noise by averaging images of a gait cycle. However, it loses the dynamical variation between successive frames. Liu and Zheng [101] develop the Gait History Image (GHI) to retain temporal information as well as spatial information. Chen et al. [102] propose the frame difference energy image (FDEI) based on GEI and GHI to address the problem of silhouette incompleteness. They calculate the positive portion of frame difference as positive values of the subtraction between the current frame and the previous frame. FDEI is defined as the summation of GEI and the positive portion. Liu et al. [103] assess the quality of silhouette sequences to determine the contribution of each GEI for classification according the quality of GEI. Xue et al. [104] apply the wavelet decomposition of GEI to infrared gait recognition. The infrared gait sequences are robust to the covariates of holding a ball and loading packages. Kale et al. [105] use the width of the outer contour of silhouette to encode the information of silhouettes. The width is defined as the horizontal distance between the leftmost pixel and the rightmost pixel of the contour. The width of the outer contour may be unreliable due to the poor quality of silhouettes. However, the silhouette itself as features may be more suitable for low quality and low resolution data.

Later, Kale et al. [106] combine the entire silhouette and the width of outer contour silhouette as gait features. Wang et al. [107] unwrap the 2D contour of silhouette to a 1D signal using the distance between pixels along the contour and the shape centroid. However, these 1D signals are easily affected by the quality of silhouettes. Dadashi et al. [108] apply wavelet transform to these 1D signals to extracted wavelet packets atoms coefficients as the gait signature. Instead of computing a distance between each pixel along the contour and the centroid, Boulgouris et al. [**?**] divide the silhouette into angular sectors and calculate the average distance between foreground pixels and the centroid in each angular sectors.

Some other algorithms pay attention to analyzing the whole shape of silhouettes. Wang et al. [7] apply the Procrustes shape analysis to silhouette shapes and extract a Procrustes mean shape from a sequence of silhouettes as gait signature. Boulgouris and Chi [109] perform Radon Transform on the binary silhouettes to get a template from gait sequences. Linear discriminate analysis (LDA) and subspace projection are used to extract Radon template coefficients to construct the feature vector.

2.6.3 Conclusion

Among the gait recognition approaches up to date, it is worth to note that (i)most approaches derived representation from human silhouette [98, 7, 110, 107, 99, 111–113, 100, 114–124] , (ii)One successful and representative stream of model-free approaches are by using Gait

Energy Image(GEI) [98, 113, 100, 114, 115, 117, 118, 121–123] or its variants [110, 116, 119, 120, 116, 124], have demonstrated powerful performance in representing human gaits. Moreover, the key task in gait identification is exploring robust feature representation that is able to tolerate the variation of challenging factors such as clothing, shoes, carrying objects, walking speed, age, gender, occlusions in the scene, variations in viewpoint. Among the approaches, the view invariant gait identification [118, 125, 121–124] was one of the most interested topic in recent years. Besides, researches on predicting factors such as gender[98, 126, 127], age[128, 129], or even simultaneously identify human[98, 124] based on gait are seldom but existed.

Chapter 3

Driver Distraction Activities Recognition by Joint Application of Motion History Image and Pyramid Histogram of Oriented Gradients

In the field of intelligent transportation system (ITS), automatic interpretation of a driver's behavior is an urgent and challenging topic. This chapter studies vision-based driver distraction 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 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.

3.1 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 [12]. The WHO also estimates that traffic accidents cost the Chinese economy over $21 billion each year. One of key contributing factors is reckless driving [12]. 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. [29] 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 chapter 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 [76, 130, 131, 18, 132]. In one study, the driver's face was tracked and yaw orientation angles were used to estimate the driver's face pose [133]. 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 [50]. 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 [134]. 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 [54]. Veeraraghavan et al. used the driver's skin-region information to group two actions; grasping the steering wheel and talking on a cell phone [55, 135]. 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 [136, 137]. Tran et al. studied driver's behaviors by foot gesture analysis [57]. Other works focused on capturing the driver's attention by combining different vision-based features and physical status of the vehicle [138–143].

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 [30], which extract silhouettes by using background subtraction and aggregate difference between subsequence in an action sequence. Other global description methods include the R transform [144], contour-based approach [145, 146], and optical flow [147–150]. 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 [151] 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 [151], 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 [152].

3.2 Contributions

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 [29] 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 chapter 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 [29], 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 [136], 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 [153].

The last contribution of this chapter is the proposal of a global grid-based representation for the driving actions, which is a combination of the motion history image (MHI) [30] and pyramid histogram of oriented gradients (POHG) [31], 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) [154] 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 Chapter is organized as follows. Section 3.3 gives a brief introduction on our driving posture dataset creation and the necessary pre-processing. Section 3.4 and Section 3.5 reviews the motion history image and the pyramid histogram of oriented gradients, with explanation of how they are applied in driving , respectively. Section 3.6 introduces the Random Forest classifier and other three commonly used classification methods for comparison. Section 3.7 reports the experiment results, followed by conclusion in Section 3.8.

3.3 Driving Action Dataset Creation and Pre-processing

A driving action dataset was prepared which contains 20 video clips in $640 \times 424@24fps$. 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, 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, then uses a lighter to light the cigarette and then put it back on the dashboard.

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

3.3.1 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 [155] was chosen, which minimize the intra-class variance of the black and white pixels. The existence of moving objects will be determined by evaluating whether there exists 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 Fig. 3.1.

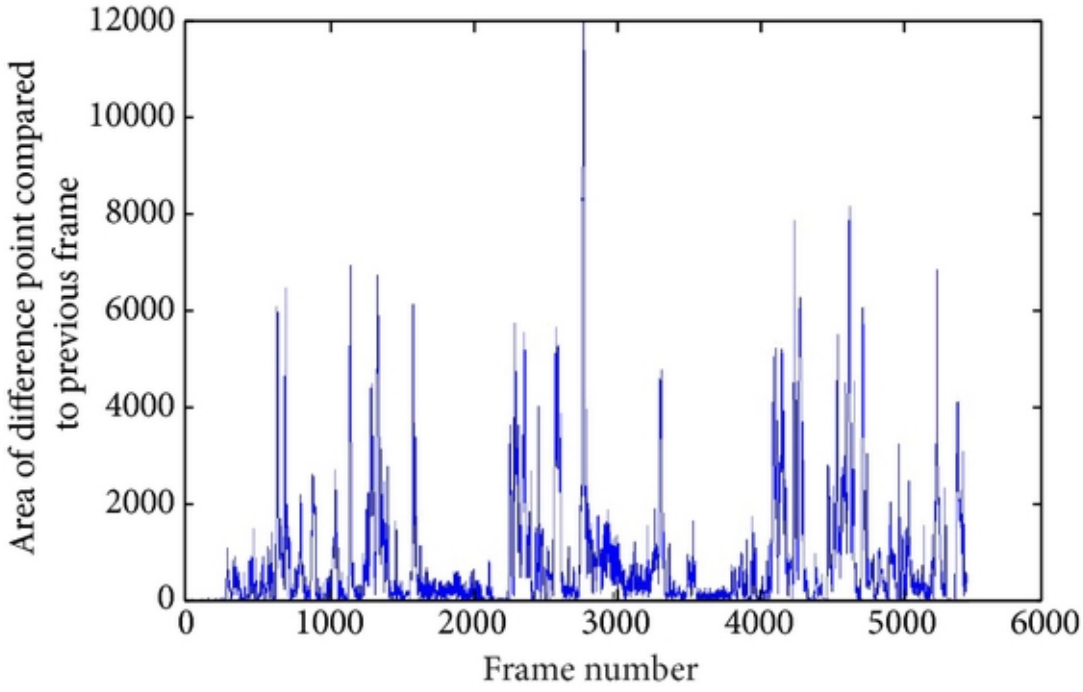


Fig. 3.1 The vertical axis stands for area of difference point compared to previous frame by applying Otsu's thresholding method, the horizontal axis stands for the frame number.

In the following section, the segmented actions images is further manually labelled into four different categories of action primitives based on the trajectory of driver's right hand as

shown in Fig. 3.2. 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 forth 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.

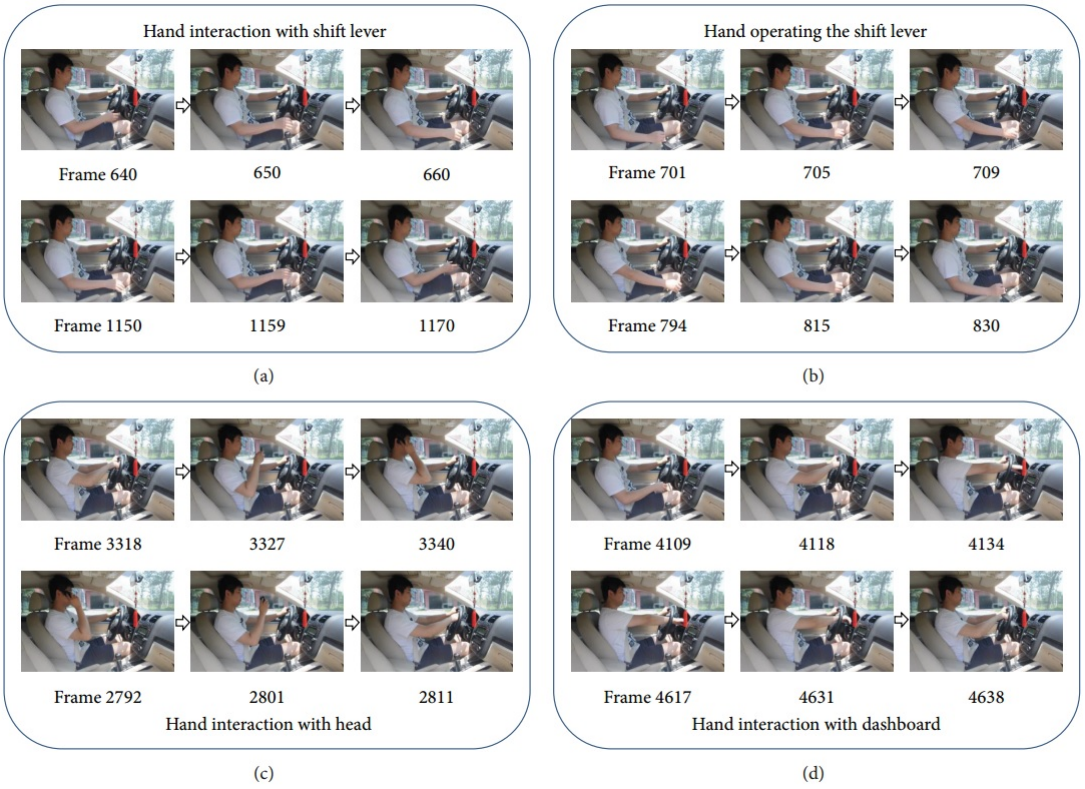


Fig. 3.2 Four manually decomposed action primitives.

3.4 Motion Energy Image (MHI)

Motion History Image (MHI) approach is a view-based temporal template approach, developed by Bobick and Davis [30], which is simple but robust in the representation of movements and is widely employed in action recognition, motion analysis and other related applications [156–158]. 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.

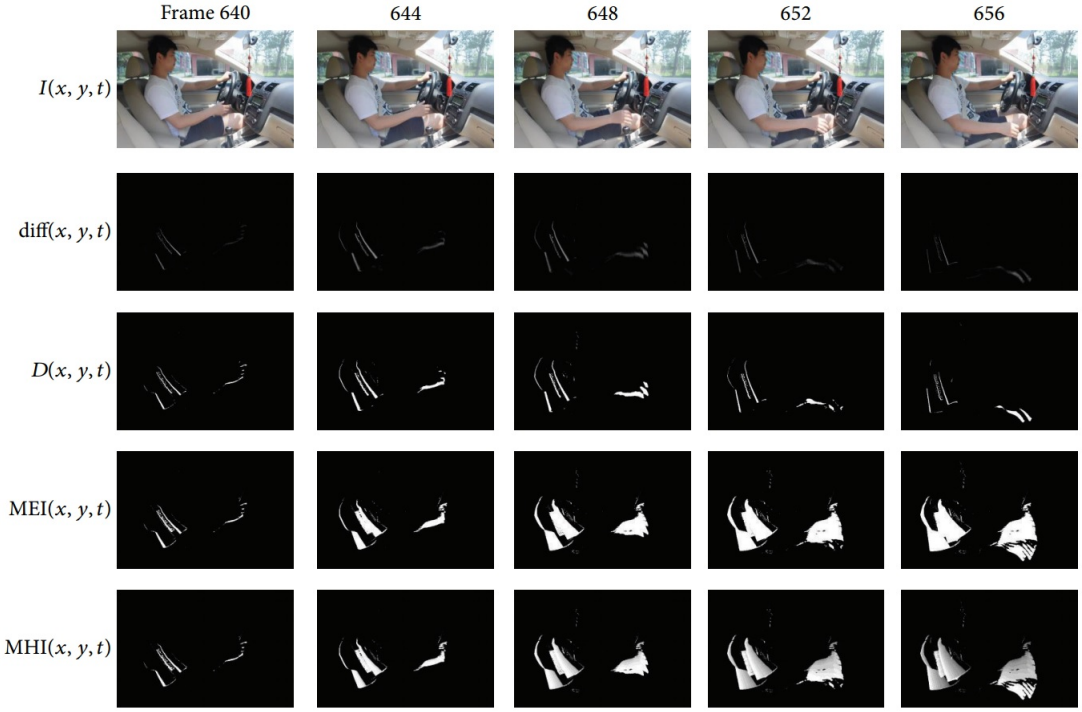


Fig. 3.3 Example of the driver's right hand moving to shift lever from steering wheel. The first row are some key frames in a driving action. The second row are the corresponding frame difference images. The third row are binary images resulted from thresholding. The forth row are cumulative motion energy images. The fifth row are cumulative motion history images.

Fig. 3.3 shows a movement in driving. The first row are 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.

$$diff(x,y,t) = \begin{cases} \text{zeros}(x,y,1), & \text{if t==1} \\ |I(x,y,t-1) - I(x,y,t)| & \text{otherwise} \end{cases} \tag{3.1}$$

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

$$D(x,y,t) = \begin{cases} 0, & \text{ifdiff(x,y,t)<threshold} \\ 1 & \text{otherwise} \end{cases} \tag{3.2}$$

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

$$MEI(x,y,t) = \cup_{t=action\_start\_frame}^{t=action\_end\_frame} D(x,y,t) \tag{3.3}$$

Both motion history images and motion energy images were introduced to capture motion information in images [17]. While MEI only indicates where the motion is, motion history image $MHI(x,y,t)$ represents the way the object moving, which can be defined as

$$\text{MHI} = \begin{cases} 255, & \text{if } D(x,y,t) == 1 \\ \max\{0, MHI(x,y,t-1) - 1\}, & \text{if } D(x,y,t) \neq 1 \text{ and } \frac{255}{pic\_seq\_length} \leq 1 \\ \max\{0, MHI(x,y,t-1) - \text{floor}(\frac{255}{pic\_seq\_length})\}, & \text{if } D(x,y,t) \neq 1 \text{ and } \frac{255}{pic\_seq\_length} > 1 \end{cases} \tag{3.4}$$

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. Fig. 3.4 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.

3.5 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 [30], 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 [159]. In this chapter, the histogram of oriented gradients feature is extracted from the MHI as the suitable features for classification.

Fig. 3.4 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

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) [154]. 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) [31] 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 [160]. 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. 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.

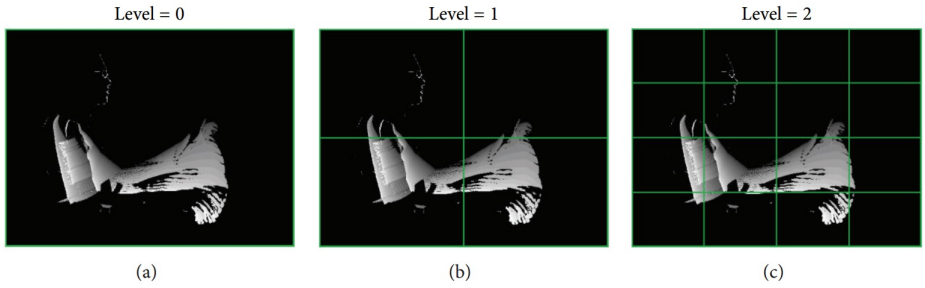


Fig. 3.5 A schematic illustration of PHOG. At each resolution level, PHOG consists of a histogram of orientation gradients over each image subregion.

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 [31] by limiting the number of levels to L = 3 to prevent over-fitting. Fig. 3.5 shows 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 caculated as follow:

$$m(x,y) = \sqrt{g_x(x,y)^2 + y_y(x,y)^2} \tag{3.5}$$

$$\theta(x,y) = \arctan \frac{g_x(x,y)}{g_y(x,y)} \tag{3.6}$$

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 $K \times 2^l \times 2^l$ 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) [161] due to its simplicity and effectiveness.

3.6 Random Forest (RF) And Other Classification Algorithms

Random forest (RF) [162] 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 creates 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 ($\approx 2/3$), 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 of trees T in the forest [162] . 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 Mreduces 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.

    - Construct a bootstrap sample of size n sampled from $S_n = (X_i, y_i)_(i = 1)^n$ with replacement and grows a tree from this bootstrap sample.
    - When growing a tree at each node select m variables at random and use them to find the best split.
    - 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 chapter 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.

3.6.1 Other classification methods

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 [29]. As a type of lazy learning, kNN classifier doesn't 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 to 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 computation expensive in practical.

Multilayer perceptron classifier

In neural network, mutilayer perceptron (MLP) is an extension of the single layer linear perceptron by adding hidden layers in between [161]. 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, i.e., the input layer, single or multiple hidden layer and an output layer. An MLP classifier is usually trained by the error backpropagation algorithm.

Support vector machine

Support vector machine (SVM) is one of the most commonly applied supervised learning algorithm. 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.

3.7 Experiments

3.7.1 Holdout experiment

We choose the two standard experimental procedures, namely holdout approach and the cross-validation approach, to verify the driving action recognition performance using RF classifier and the PHOG feature extracted from MHI. Other three classifiers, *k*NN, 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, clasification accuracy is one of the most common indicators used to evaluate the performance of the classification. Fig. 3.6 and Fig. 3.7 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 averages from 100 runs. The average classification accuracies of k-NN 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, i.e. 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 examples with the correct value as represented by the row. Thus, the diagonal elements show the number of correct

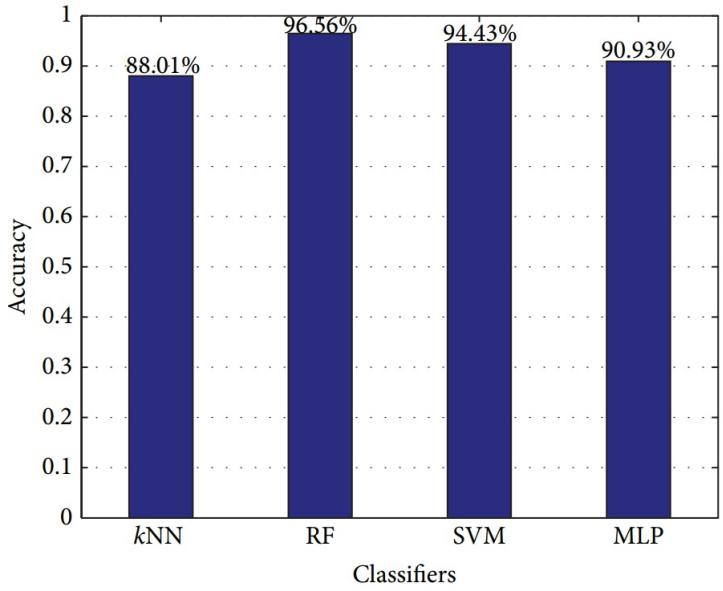


Fig. 3.6 Bar plots of classification rates from holdout experiment with 80% of data are used for training, and the remaining for testing.

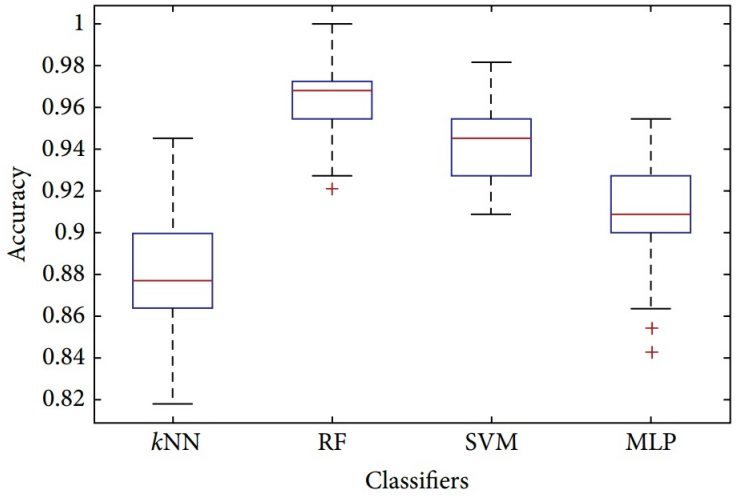


Fig. 3.7 Box plots of classification rates from holdout experiment with 80% of data are used for training, and the remaining for testing.

classifications made for each class, and the off-diagonal elements show the errors made. Fig. 3.6 shows the confusion matrix from the above experiment for the RF classifier.

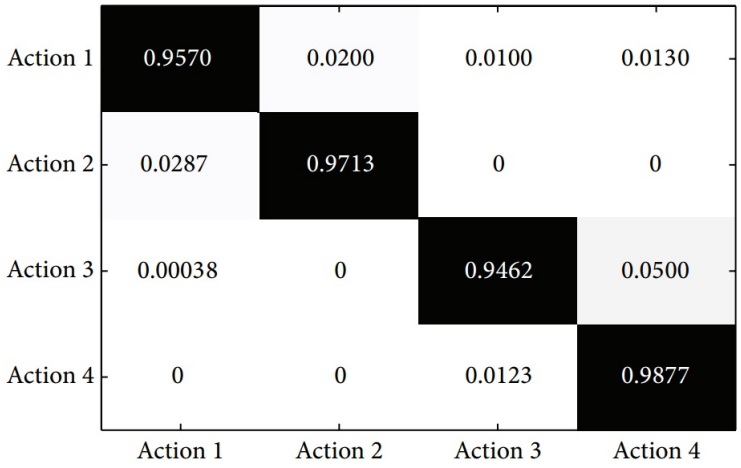


Fig. 3.8 Confusion matrix of RF classification result from the holdout experiment.

In the table, 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 class 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 confusions between class three and four lie between $1.2\% \sim 5\%$.

3.7.2 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 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 nine of the ten splitted sets are used for training and the remaining one reserved for testing.

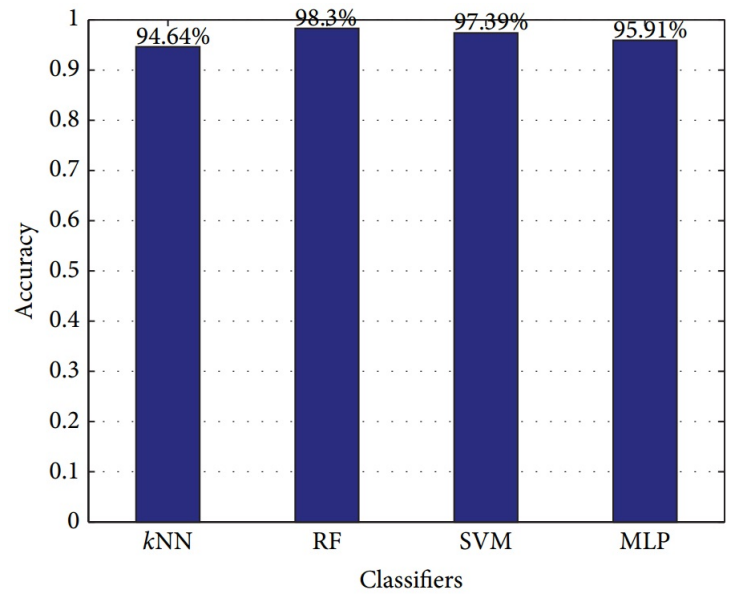


Fig. 3.9 Bar plots of classification rates from 10-fold cross-validation.

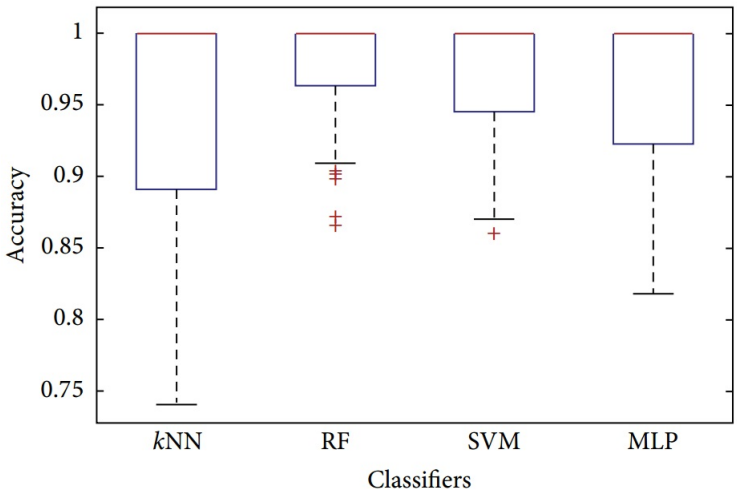


Fig. 3.10 Box plots of classification rates from 10-fold cross-validation.

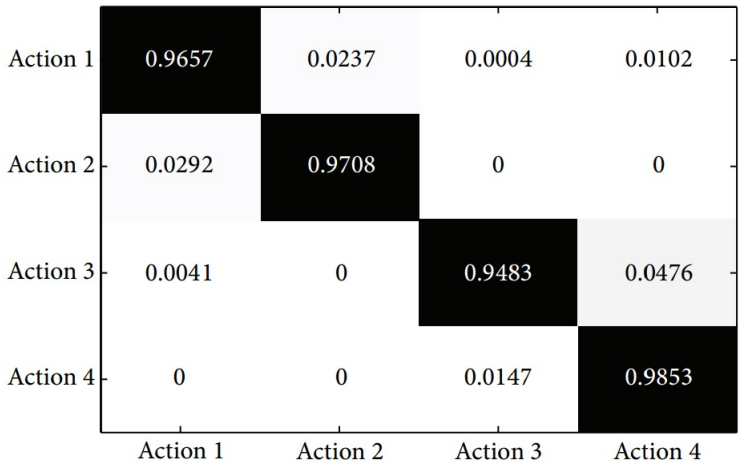


Fig. 3.11 Confusion matrix of RF classification from 10-fold cross validation experiment.

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 repetations are shown in the bar plots of Fig. 3.9 and box plots of Fig. 3.10. 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 and box plots in Fig. 3.9 and Fig. 3.10, the RF classifier clearly outperforms other three classifiers compared.

3.8 Conclusion

In this chapter, 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 leads 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.

Chapter 4

Video-Based Classification of Driver Distraction Behaviour using a Hierarchal Classification System with Multiple Features

Driver fatigue and distraction have long been recognized as the main contributing factors in traffic accidents. Development of intelligent driver assistance systems, which provide automatic monitoring of driver vigilance, is therefore an urgent and challenging task. This chapter presents a novel system for video-based driving behaviour recognition. The fundamental idea is to monitor driver hand movements and to use these as predictors for safe/unsafe driving behaviour. In comparison to previous work, the proposed method utilises hierarchal classification and treats driving behaviour in terms of a spatial-temporal reference framework as opposed to a static image. The Approach was verified using the Southeast University Driving-Posture Dataset, a dataset comprised of video clips covering aspects of driving such as: normal driving, responding to a cell phone call, eating and smoking. After pre-processing for illumination variations and motion sequence segmentation, eight classes of behaviour were identified. The overall prediction accuracy obtained using the proposed approach was 89.62% when using a hierarchical classification approach. The proposed approach was able to clearly identify two dangerous driving behaviours, *Responding to a cellphone call* and *Eating*, with an overall recognition rate of 91.87%.

4.1 Introduction

In this chapter, a video camera-based system to monitor driver behaviour and distinguish between safe and unsafe driving behaviours, is proposed that operates according to the analysis of hand movements and usage. This entails a number of challenges namely: (i)motion detection and segmentation, (ii)motion representation, and (iii)the classification of the hand gestures. For this purpose, unsafe hand movements and usage include: smoking ,eating, using a cell phone and adjusting the controls of the dashboard while driving. A further challenge is the nature of the required video data pre-processing to compensate for noise and illumination variation.

In the proposed video-based driving behaviour recognition system, raw video data was first pre-processed to compensate for illumination changes to improve the performance of motion detection. The pre-processing procedure uses a proposed two stage intensity normalization technique to minimize the influence from illumination variation. Next, the processed video data was segmented into video clips based on if motion exists. In this system, then the motion clips was represented using Gait Energy Image [100] and Pyramid histogram of gradient [163] to reduce data dimension. Finally, a hierarchal classification system is applied to improve the recognition performance. The proposed approach was test on the Southeast University Driving-Posture Dataset(SEU dataset). It includes activities of normal driving, responding to a cell phone call, eating and smoking.

Given the above, the contributions of this chapter are as follows:

1. A view-based temporal-spatial template approach to represent driving video sequences and that (as will be evidenced later in this chapter)archived competitive performance. Contrary to many perviously published work, this chapter argues that driver behaviour analysis is better treated as temporal-spatial problem as opposed to a static images analysis problem; driving behaviour analysis is a space-time human activity. It is argued that usage of static images is not sufficient to distinguish between classes of behaviour types and that this can only be done by considering a sequence of images(video frames).

2. A two stage intensity normalization preprocessing technique to minimize the influence from illumination variation. The first stage comprised a moving average method that smoothed the intensity variation caused by periodic lighting change. The second stage comprised application of the three frame difference method[32] to detect motion. For the task of motion detection and segmentation in video, it was found that the proposed two-stage pre-processing technique performed well in context of compensating for noise and illumination variation in video data.

3. A hierarchal classification system for driving behaviour recognition which considers different sets of features at different levels. Hierarchical classification is specifically intended for data where the features of interest can be arranged in a hierarchical manner. As such it offers advantages in terms of learning and representation in comparison to attempts to use "flat" classification techniques for the purposed of classifying hierarchical data[33]. These efficiency gains are realised because only a subset of the complete set of available features is considered at each node in the hierarchy. Hierarchical classification schemes have been applied in many areas [34–36]. However, it should be noted here that, to the best knowledge of the authors, they have not been applied to driving behaviour recognition.

The rest of the chapter is organized as follows. Section 6.2.1 gives a brief introduction to the SEU driving dataset followed by an overview of our proposed recognition system in Section 7.2. Section 7.2.1 explains the nature of the required preprocessing of the video data especially in the context of illumination variation. Section 4.5 introduces the driving motion segmentation algorithm and motion representation by Gait Energy Image(GEI) representation. Section 4.6 gives details of the hierarchal classification system adopted to predict driver behaviour. Section 7.4 reports the conducted evaluation and the experiment results obtained, this is followed by some conclusions presented in Section 7.5.

4.2 The SEU Driving Dataset

Fig. 4.1 SEU driving dataset

To test the proposed driving behaviour recognition approach, the Southeast University Driving-Posture Dataset(SEU dataset) was used. This data was first created by Zhao [164]. Some selected frames from this dataset are shown in Fig.5.2. Each video included in the dataset was obtained using a side-mounted Logitech C905 CCD camera under day lighting conditions with a resolution of 640x480. Ten male drivers and ten female drivers participated in the creation of the dataset. Each video was recorded under normal day light conditions, poor illuminated night time conditions were not considered.

4.3 System Overview

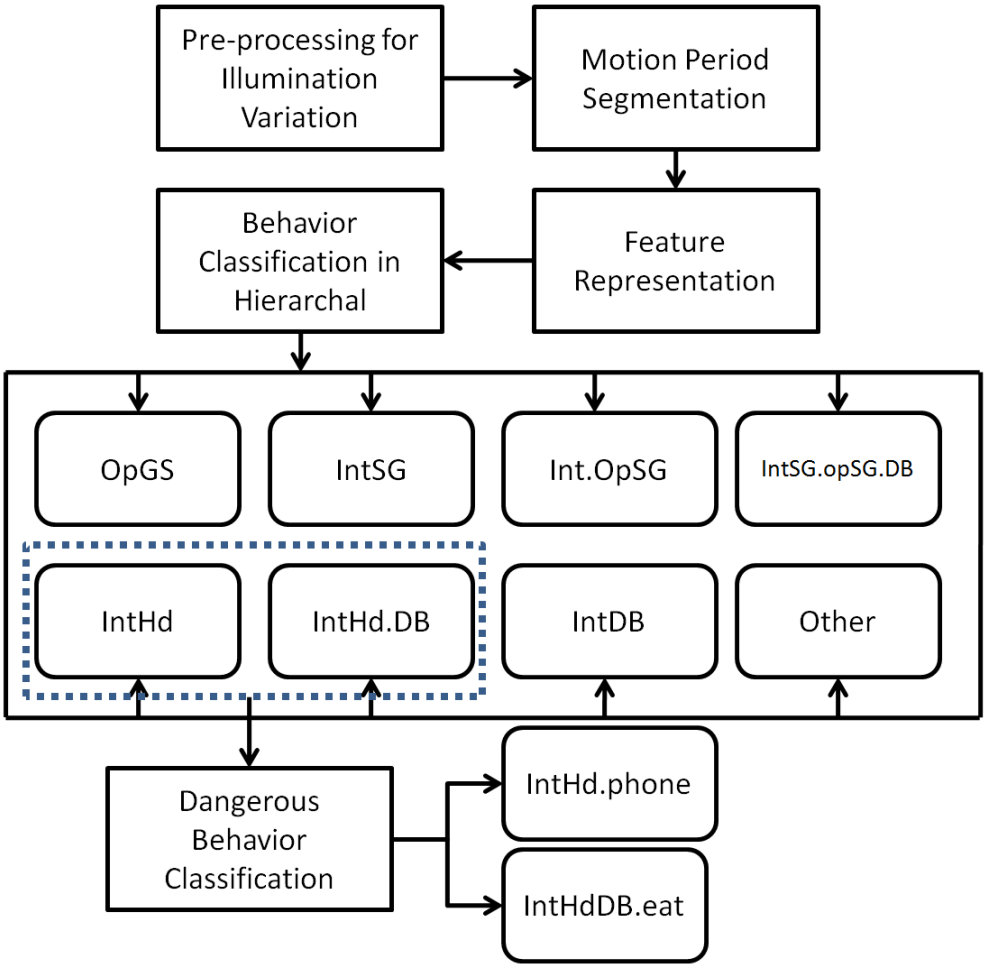


Fig. 4.2 System overview.

A schematic illustrating the operation of the proposed driving behaviour recognition system is shown in Fig.7.5. In the figure the directed arcs indicate the next step followed by

Table 4.1 Driver behaviour class definition

Class	Abbreviation	Description
1	OpGS	The normal operation of the gear shift.
2	IntSG	Interaction with the gear shift. Thus the movement of the right hand from the steering wheel to the gear shift, or the reverse procedure.
3	Int.OpSG	Interaction with the gear shift and then operation of the gear shift. It represents compositional behaviour comprising IntSG and OpSG
4	IntSG.opSG.DB	Interaction with and operation of the gear shift, followed by movement to Dashboard. The class describes the situation where right hand is first used to operate the gear shift, then moves back to the steering wheel and then reaches towards the dashboard.
5	IntHd	Describes situation where the driver moves his right hand towards or away from his/her head. For example moving food towards the mouth or taking a call by moving a cell phone towards the ear (we call this head interaction)
6	IntHd.DB	Interaction between head and dashboard, encompasses IntHd.DB and IntDB
7	IntDB	Describes situation where the driver moves his right to place something on the dashboard or take something away from the dash board. For example, taking a cigarette from a packet or replacing a cigarette lighter.
8	Other	Behaviour undefined in the previous seven classes, such as turning of the steering wheel.

previous one. The proposed system comprises five steps: (i)video data pre-processing so as to compensate for noise and illumination variation, (ii)motion segmentation, (iii) feature representation, (iv)hand movement classification using a hierarchical classification model and (v)dangerous behaviour classification. Eight kinds of driver behaviour class, each defined according to driver hand movement, were identified as shown in the Table 4.1

From the table it can been seen that the identified eight driver behaviour classes are defined in terms of the physical position and/or movement of a driver's right hand. As our emphasis is on dangerous driving behaviour, such as responding to a cellphone call and eating, in the case of classes 5 and 6 are further analysis is applied with the result that video sequences corresponding to these classes may be reclassified as belonging to class 5* or 6* as shown in table 4.2:

Table 4.2 Dangerous Driver behaviour class definition

Class	Abbreviation	Description
1	IntHd.phone	Driver takes a cellphone from somewhere, such as dashboard, and place it on the profile of head
2	IntHdDB.eat	Either eating or smoking a cigarette.

4.4 Motion Detection

The task of driver behaviour monitoring can be generally studied within the human action recognition framework [165], that is action detection, action segmentation, action representation and action classification. The emphasis of the framework is often on finding good feature representations tolerant of variations in viewpoint, human subject, background, illumination, and so on. One of the common strategies of representing human motion is global description, which regards the visual observation as a whole. Global representation can be derived from motion object silhouettes [166, 167] based on effective motion detection and segmentation.

Three commonly used approaches to motion detection or moving object detection are: (i)temporal differencing, (ii)background subtraction and (iii)optical flow. Temporal difference methods are simply based on the subtraction of two consecutive frames followed, a similarity threshold is then used to determine whether the frames are different or not [155]. In the background subtraction methods, what is known as a background image is modeled first. This is a benchmark image against which other images are measured. Motion is identified by calculating the difference between a current frame and the background image [168]. A similarity threshold is then again used. Both of these methods can work well if an appropriate threshold value is used; however, this is not a trivial task in practice. A further disadvantage of the temporal difference approach (and its variants) is that when using this approach it is generally not possible to extract the complete contours of moving object. In the case of the background subtraction approach a further disadvantage is that it critically relies on precise background modeling, which in turn has a series of open problems. The essence of optical flow is to estimate the motion field and merge the motion vectors with similarities. The optical flow approach has been found to work well in the presence of camera motion [169], but with the disadvantage of higher computing capability requirement and the side-effect of being sensitive to noise.

From the above, in action recognition research, temporal difference is often preferred due to its computational efficiency and its consequent potential for usage in real-time applications. However, as noted above choosing a threshold value is a challenging problem. One widely

used solution is to Otsu's method [155] for selecting a threshold. Otsu's method minimizes the intraclass variance of the black and white pixels while at the same time being tolerant to slight and slow variation of illumination. The temporal difference motion detection approach, coupled with Otsu's threshold selection technique, was thus adopted with respect to the work presented in this chapter. However, prior to its application, two kinds of illumination variation found in the SEU dataset had to be taken be addressed, namely: (i)periodic variation, and ii)sudden change. The proposed mechanism for addressing these illumination variation issues are presented in Sub-sections 4.4.1 and 4.4.2.

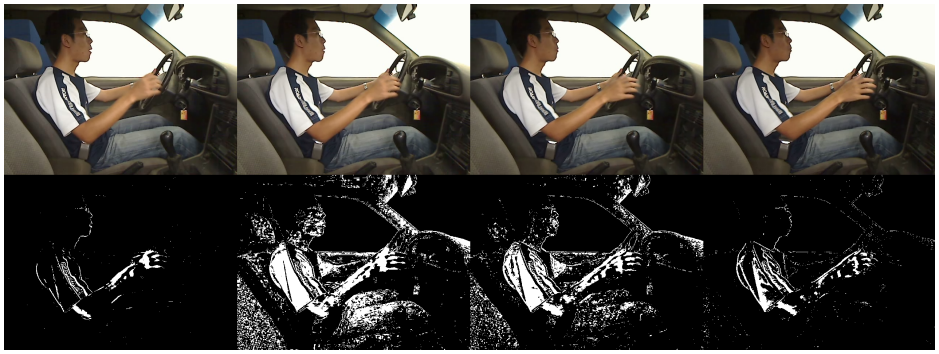
4.4.1 Periodic Variation

Periodic illumination variation occurs when a vehicle is passing a sequence of road side objects (such as lamp posts) where by the vehicle under illumination changes in a regular pattern. This type illumination variations thus quasi-periodic and as such is a negative influence on motion detection. This is particularly the case with respect to the temporal differencing approach used with respect to the work presented in this chapter because false foreground appears if illumination varies quasi-periodically.

Fig. 4.3a further explains the quasi-periodic illumination variations which arises from the simulated SEU driving dataset. In the figure, the first row comprises an image sequence representing a movement of the right hand reaching towards the gear shift. The second row is the corresponding sequence of frame differences generated by applying temporal difference motion detection (coupled with Otsu's threshold method). The white pixels indicate differences with respected to the previous and consequently are indicative of motion. Obviously, the direct frame differencing results are too noisy to be proceeded for moving object detection. Such a detriment is caused by the quasi-periodic lighting change, as demonstrated by Fig.4.4a, which shows the change of intensity value with time for a specific video sequence. From the figure peaks and throuhs in intensity value can be observed. As the video is recorded 30fps, the intensity value jumps roughly about every half a minute.

In order to reduce the influence from the above quasi-periodic lighting change, we proposed an intensity compensation method by smoothing the sharp peaks and valleys. For each frame in a given sequence, we first calculate the difference between the intensity values and the moving average intensity values with respected to no-motion area. Then we compensate each frame by adding the intensity difference to each pixel in the frame. The process is as follows:

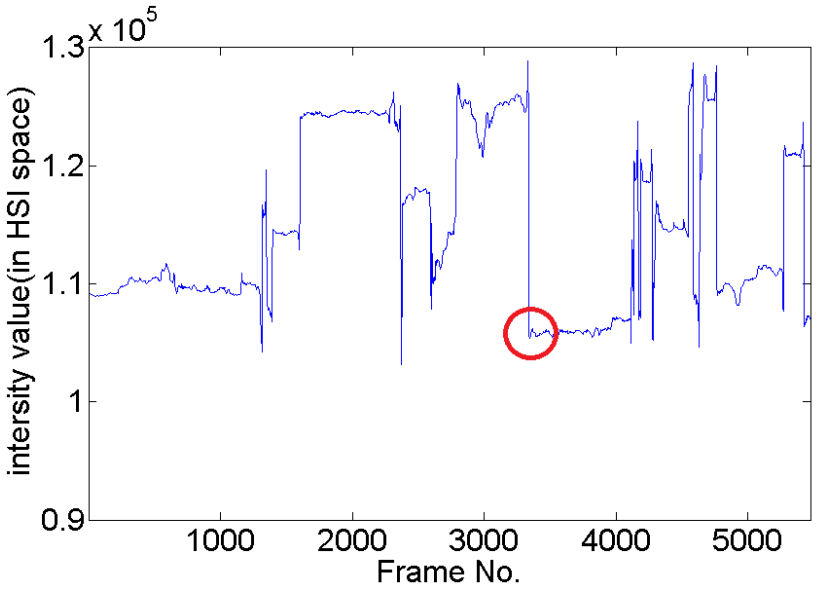
1. **Step 1:** For a given video sequence, we calculate the frame difference for each pair of consecutive frames and add these frame differences together. The final aggregated



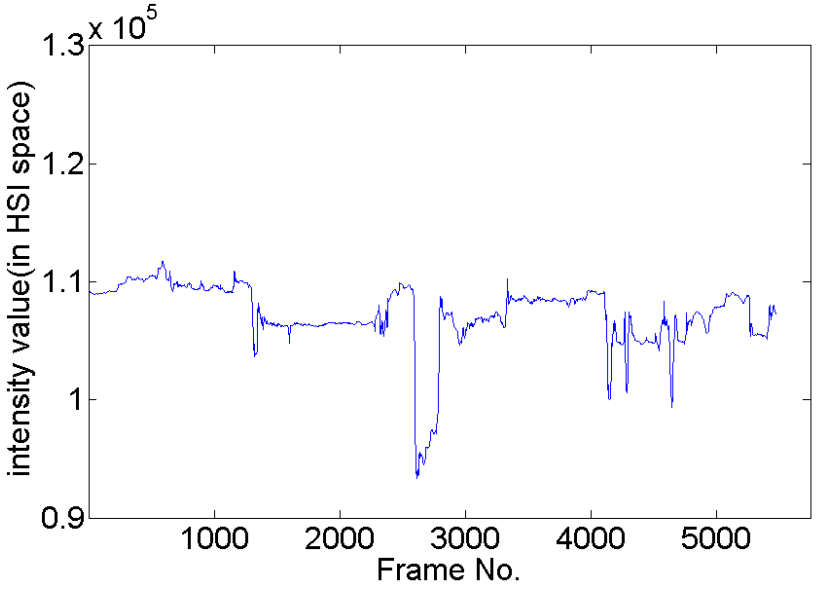
(a) Before compensation

(b) After compensation

Fig. 4.3 An example of Negative influence caused using periodic illumination variation and its compensation result



(a) Before processing



(b) After processing

Fig. 4.4 Intensity plot of video 25

frame difference is thresholded by Otsu's method [155], resulting in a mask for the static pixels. A set of 16 example masks are shown in Fig. 4.5, with black and white pixles representing motion and no motion, respectively.

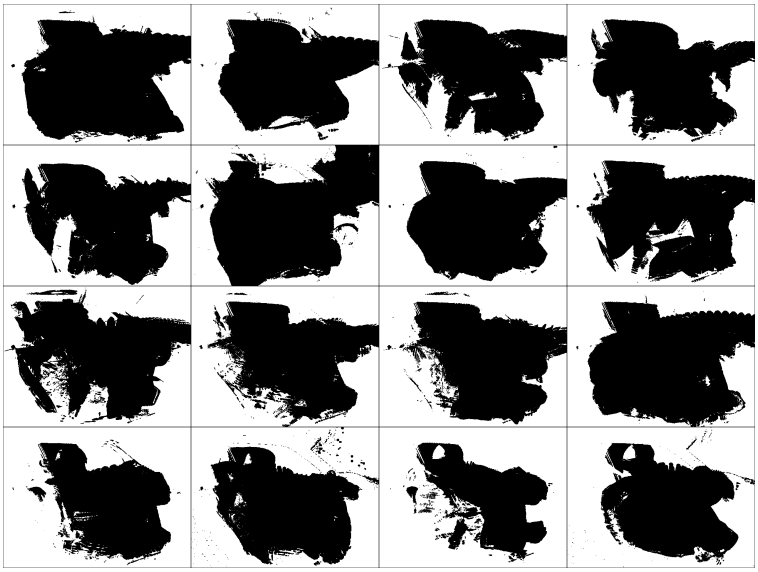


Fig. 4.5 Examples of no-motion masks.

2. **Step 2:** The mask from above step 1 is multiplied to its corresponding video frames I_n, with n for frame index, to yield the intensity sequences of no-motion area, denoted as $\bar{I}_n$.

3. **Step 3:** The moving average of $\bar{I}_n$ is defined as

$$BPI_n = \begin{cases} \bar{I}_n & \text{if } n = 1 \\ (1-a) \times BPI_{n-1} + a \times \bar{I}_n & \text{if } n > 1 \end{cases} \tag{4.1}$$

   where a is a coefficient representing the degree of weighting decrease.

4. **Step 4:** The difference between the BPI_n and $\bar{I}_n$ is generated, namely $\text{diff}_n = BPI_n - \bar{I}_n$.

5. **Step 5:** Finally, for nth frame Im_n in the original sequence, the intensity compensated result Im'_n is given by $\text{Im}_n + \text{diff}_n$

it should be noted that the compensation algorithm is directed specifically at the quasi-periodic illumination variation phenomena. The effect of the above compensation algorithm can be seen by comparing Fig. 4.4b with Fig.4.4a. Both figures feature the same video sequence, the first without compensation, and the second with compensation. Noise reduction can clearly be observed from Fig.4.3b.

4.4.2 Sudden Change Variation

While the influence from quasi-periodic illumination change can be compensated to a large extent by the proposed intensity compensation method, sudden light change remains a problem, which may bring false motion area when the simple temporal difference is applied. In recent years, there have been some exploratory works on the robust moving object detection against fast illumination changes [170–172], some of which are extended from temporal difference. For example, a three-frame difference method was proposed in [32], aiming to solve occluded objects detection while alleviating the negative effect from sudden illumination changes. A recent approach [173] uses several temporal reference images to detect moving objects and adapt to sudden illumination change, holes are reduced inside the foreground. However, the detected objects may drag ghost artifacts due to the use of several consecutive frames possibly involving moving objects.

In our works, the three frame difference approach [32] was applied to the intensity compensated sequence to robustly detect moving objects. The approach first applies frame difference to three consecutive frames, and then make an AND operations to the results. Specifically, denote three consecutive frames f_{k-1}, f_k and f_{k+1}, then two binary images D_1 and D_2 can be obtained:

$$D_1(x,y) = \begin{cases} 1, & |f_k(x,y) - f_{k-1}(x,y)| \geq T \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

$$D_2(x,y) = \begin{cases} 1, & |f_{k+1}(x,y) - f_k(x,y)| \geq T \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

Then the three difference image is given by as follows:

$$D(x,y) = \begin{cases} 1, & D_1(i,j) \cap D_2(i,j) = 1 \\ 0, & D_1(i,j) \cap D_2(i,j) = 0 \end{cases} \tag{4.4}$$

The performance is shown in Fig. 4.6, the first row is an original image sequence representing driver's hand moving back from the dashboard after intensity compensation. There exists an illumination sudden change between the third and forth frame of the first row. The second row is the corresponding two consecutive frame differencing image threshold by Otsu's method. The intensity sudden change caused false foreground in the third frame of the second row. By applying three difference method, the three frame differencing image was shown in the third row which proves that the the false foreground was reduced.

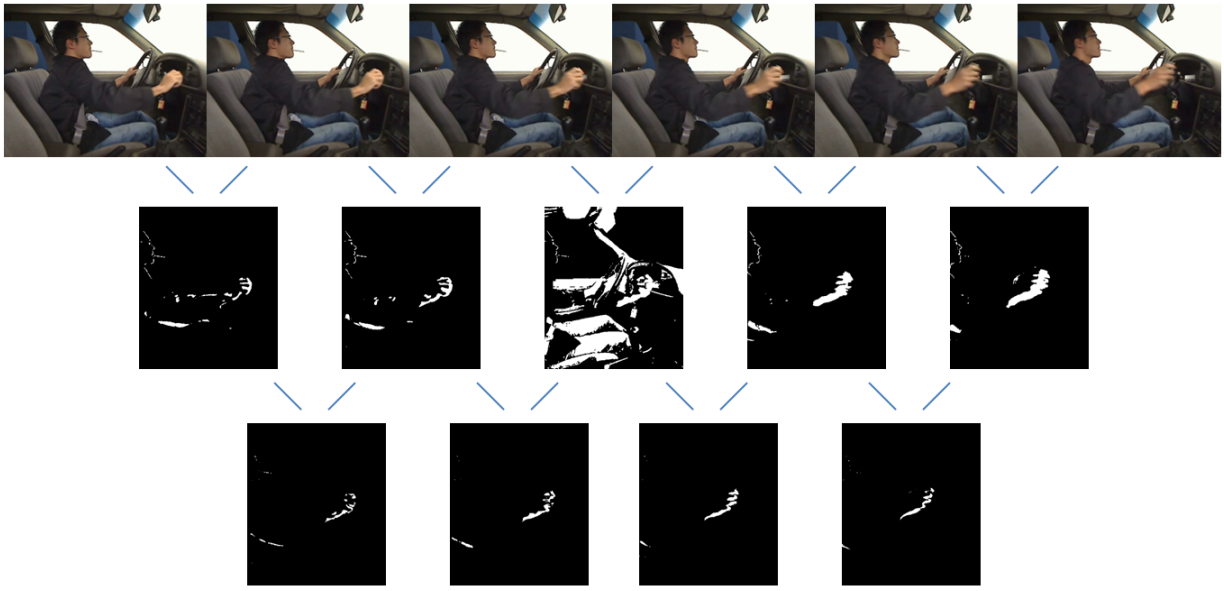


Fig. 4.6 The first row is the original image sequence after intensity compensation. The second row is the corresponding two consecutive frame differencing image threshold by Otsu's method. The third row is the three frame differencing image corresponded to the second row

.

4.5 Driving Motion Segmentation and Representation

There has been a large body of work that addresses the topic of automatic human action recognition, which focus on the video analysis based on durations and changes of spatial features over time, for example, flow-based iterations [174], motion history image [30], and local interesting points [152]. An implicit assumption on these features, namely, the availability of consecutive frames on a small group of predetermined pixels from which the features are calculated, can not be made in practice. It remains a challenge to find a generic vocabulary of parts of actions, and the corresponding methods for breaking video streams into the corresponding segments.

Currently, there exists several different kind of methods to temporally segment video streams into fragments or clips [165], including boundary detection [175, 176], sliding windows [177, 178]and grammar concatenation [179, 180]. Among the methods proposed, the boundary detection is relative easy and efficient for the driving behaviour video analysis. Specifically in our approach, motion clips are segmented if there exists a continuity of at least 15 frames with which motion area is over 950 pixels. The two values, i.e., 15 frames and 950 pixels, are from empirical analysis of the SEU datasets. This can be further explained by Fig. 4.7, which plots the detected motion area in pixels over the frames for the video No.25

of the SEU dataset, showing that six motion clips can be segmented between frames 2000 to 3000. With the simple boundary detection method for video segmentation, 527 motion clips are obtained from 20 raw videos sequence.

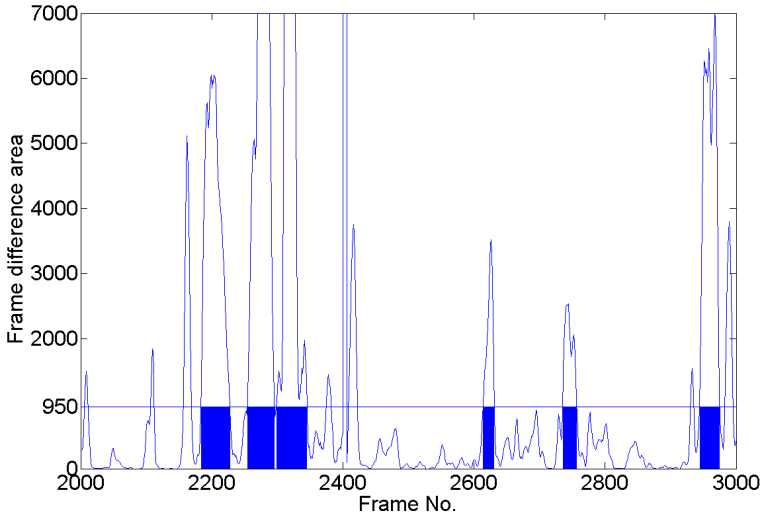


Fig. 4.7 Motion period segmentation

.

Motion clips segmented from the original video is a sequence of high-dimensional images, which cannot be directly applied for classification. We have obtained satisfactory recognition performance in our earlier works by representing motion clips with motion history image (MHI) [30] and pyramid histogram of oriented gradients (PHOG) [163]. Motion history image (MHI) is a view-based temporal approach, which is simple yet robust in the representation of movements and is widely employed in action recognition, motion analysis, and other related applications [156–158]. The essence of MHI is to describe motion in the image sequence by representing a pixel intensity as a function of the recency of motion in a sequence, where brighter values correspond to more recent motion. Inspired by MHI, a special motion feature expression approach, termed Gait Energy Image (GEI), was proposed for individual gait recognition [100] and later applied in repetitive human activity classification [181] due to a number of attractive attributes. Recently, some extensions or variants of GEI have been proposed [182, 183].

GEI is a simple yet competitive appearance based method that exploits average (i.e., energy) cues as motion features of the whole sequence. With period of gait or other action estimated, GEI can be used to represent the motion with both spatial and temporal information included, and their robustness to specific noises have been proved [184]. Suppose $B_t(x,y)$ is the binary silhouette images at time t in a sequence, GEI is defined as follows:

$$GEI(x,y) = \frac{1}{N}\sum_{t=1}^{N} B_t(x,y) \tag{4.5}$$

where N is the number of frames, t is the frame number in the sequence, and x and y are values in the 2D image coordinate.

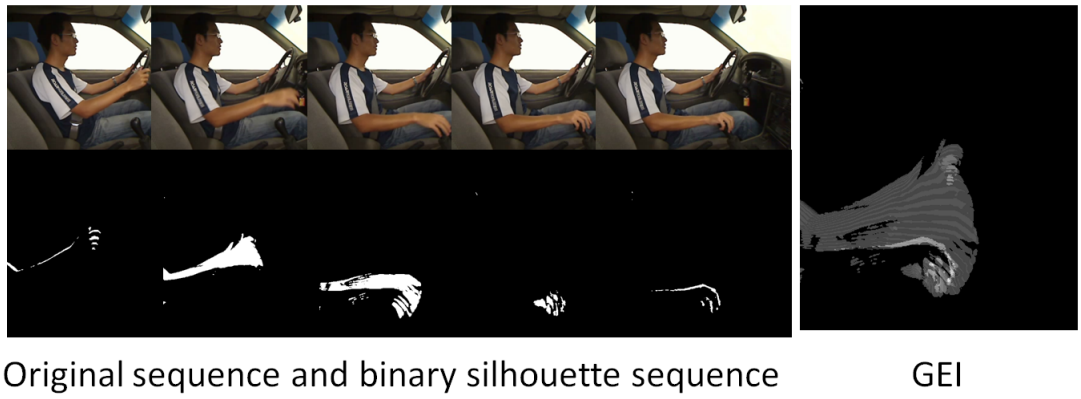


Fig. 4.8 Example procedure in extracting gait energy image.

An example procedure of extracting GEI form driving behaviour is illustrated in Fig.4.8. The first row in left part of the Fig. 4.8 is an original sequence while the second row in left part of the Fig. 4.8 is the corresponded silhouette sequence generated from original sequence by the approach described in pre-processing section. The right part of the Fig. 4.8 is the GEI by averaging the silhouette sequence. From the example gait energy image, it is obvious that higher intensity pixels indicate static areas, while lower intensity pixels highlight dynamic portions of the performed actions.

4.6 Hierarchal Classification of the Driving Behaviour

To alleviate the problems from applying flat classification on overlapping classes, which is obvious for some subclasses defined in Section 7.2, a commonly applied methodology of hierarchal classification is adopted [34–36].

With the aid of explanation of Fig. 4.9, a segmented video clip is first classified into *shift gear related* and *shift gear not-related* classes, each of which will be further classified in next level of the hierarchy. Different regions of interest (ROI) and features can then be exploited for the different subclasses.

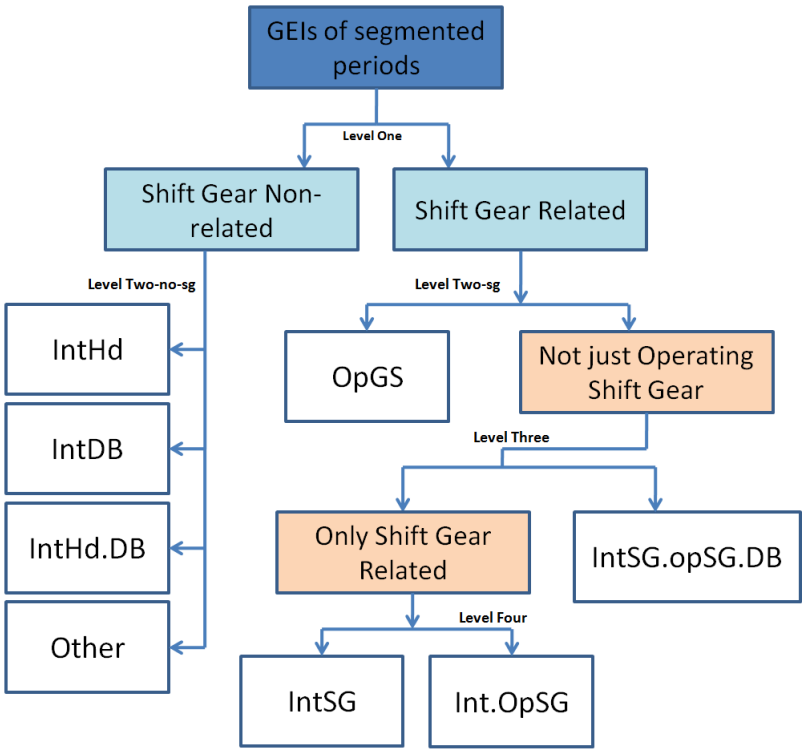


Fig. 4.9 Hierarchal classification system

.

4.6.1 Level One Classification

We applied SVM classification [185, 186] for the first level classes to make a distinction between the *shift gear related* and *shift gear not-related* behaviours. When a driver conducts behaviours including *OpGS* or *IntSG*, the hand will appear in the right bottom in the field of viewing, as indicated by red circle in Fig. 4.10. The shift gear related area can then be represented by the motion energy images (MEI) for the two classes, as illustrated by Fig. 4.11.

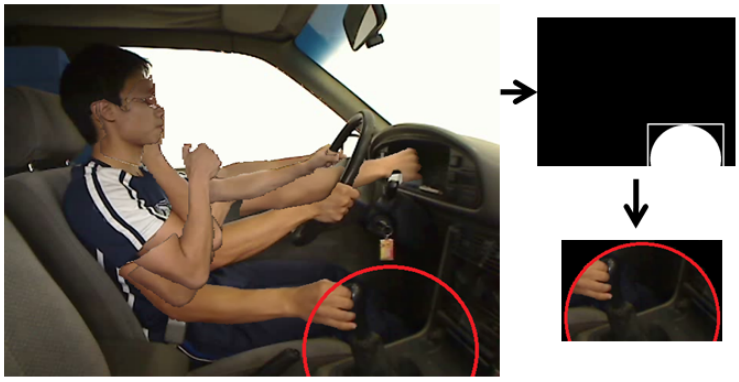


Fig. 4.10 ROI based on skin region time lapse image

.

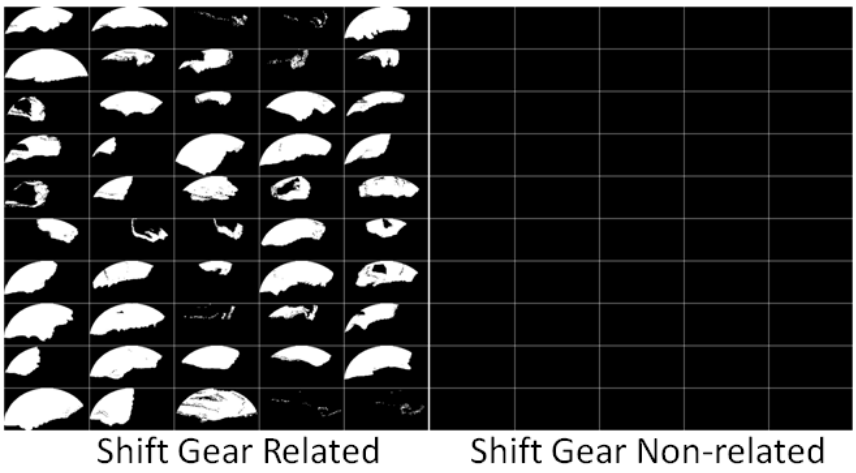


Fig. 4.11 Two classes in level one of the hierarchal classification system

.

4.6.2 Level Two Classification

There are two branches in the 2nd level of class hierarchy. The first branch (abbreviated as level two-sg in the figure) categorizes two situations, namely, *OpGS* and *not only operating*

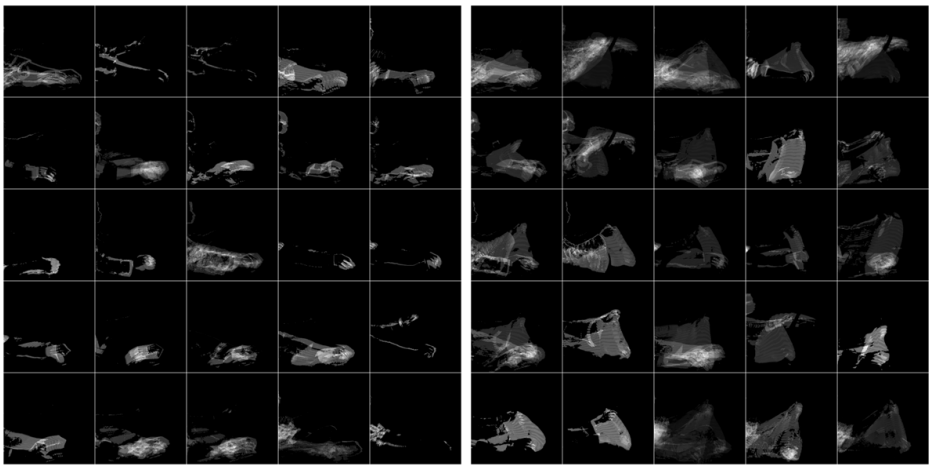
shift gear. A random forest classifier [187] is trained to classify the two groups of pattern as shown in Fig. 4.12a. The second branch (abbreviated as level two-no-sg in the figure) covers the following four cases: *IntHd*, *IntHd.DB*, *IntHd.DB*, and *Other*, as shown in Fig. 4.12b. Similar to the previous discussion, random forest classifier is trained to classify the four groups of GEI.

4.6.3 Level Three Classification

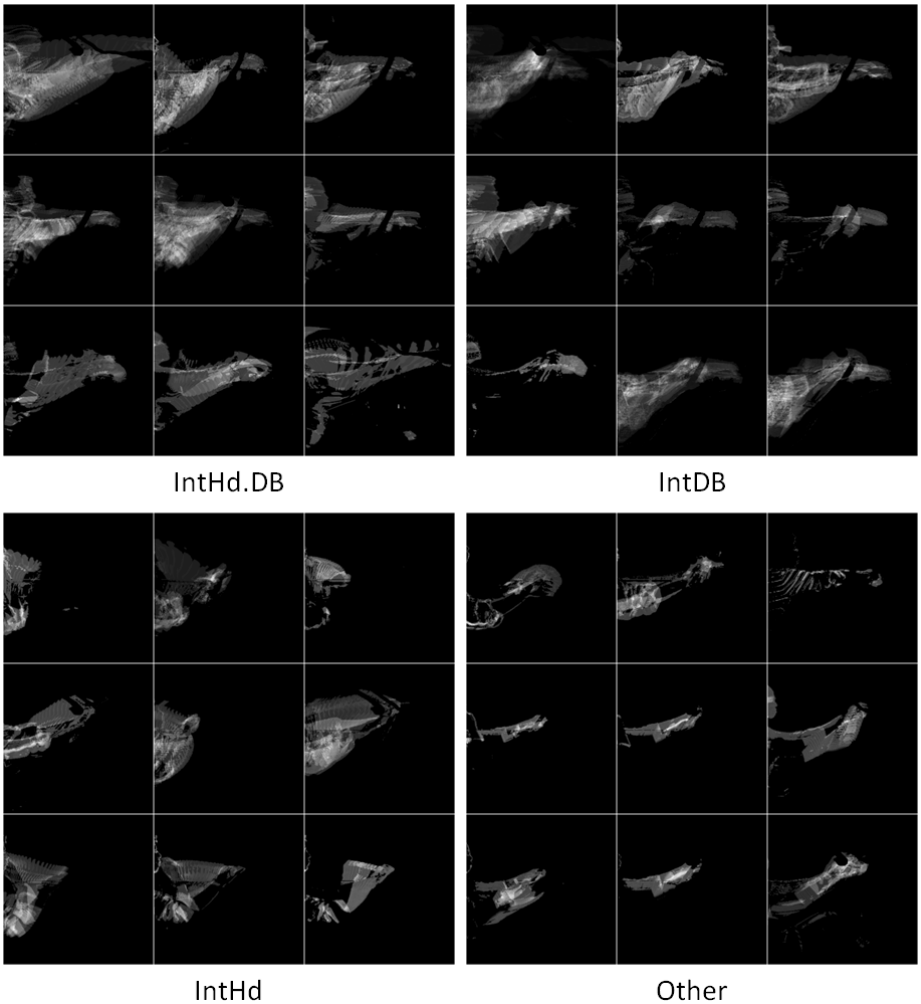
In the third level of classification hierarchy, two subclasses of the *not only operating shift gear* class are defined, that is *Only shift Gear Realted* and *IntSG.opSG.DB*, as shown in Fig. 4.13a. There exists much overlapping if in the GEI feature space, which makes classification difficult. As the two behaviours are performed by the right hand with motions mainly consisting of moving among shift gear and steering wheel and dashboard, the trajectories of the right hand are easier to distinguish. One possible approach to locate the right hand is by skin-region analysis in a well-defined region of interest (ROI). Specifically, we further extract the right hand skin-region in a ROI for each image of the action sequence, and combine them to form a right hand skin-region GEI. There exist many methods for skin region segmentation, for example, difference color space thresholding [188], Gaussian and mixture of Gaussian distributions thresholding method [189]. In this experiment, we simply segment the region of skin based on the following decision rules for the pixel value in YCbCr color space:

$$\begin{cases} 80 \leq Cb \leq 120 \\ 140 \leq Cr \leq 170 \end{cases} \tag{4.6}$$

Fig. 4.14 demonstrates the above procedure of locating the right hand skin region in ROI. The first row is four selected frames from the original sequence. The second row is the skin region after applying the above rule corresponding to the first row. As the two classes of behaviours are related to the shift gear region and the dashboard region, the region of interest(ROI) is located at a right trapezoid region of the lower right corner of the frame, which covers the shift gear region and the dashboard region. We only estimate the right hand region in ROI. The third row shows the hand region in ROI after connected component analysis and further analysis of the hand area. After locating the right hand skin region in ROI for each frame in the sequence, the right hand region sequence is combined to form another group of GEI, as shown in Fig. 4.13b, which is much easier to classify compared to the pattern in Fig. 4.13a.

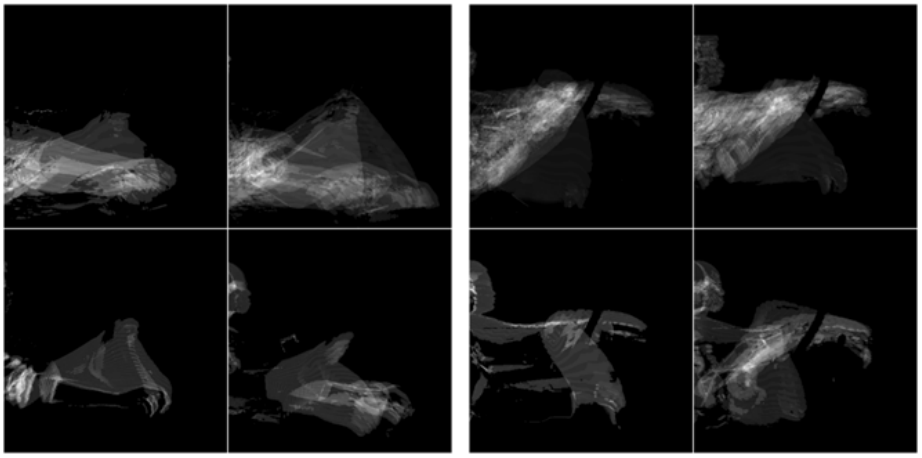


(a) Two classes of shift gear related

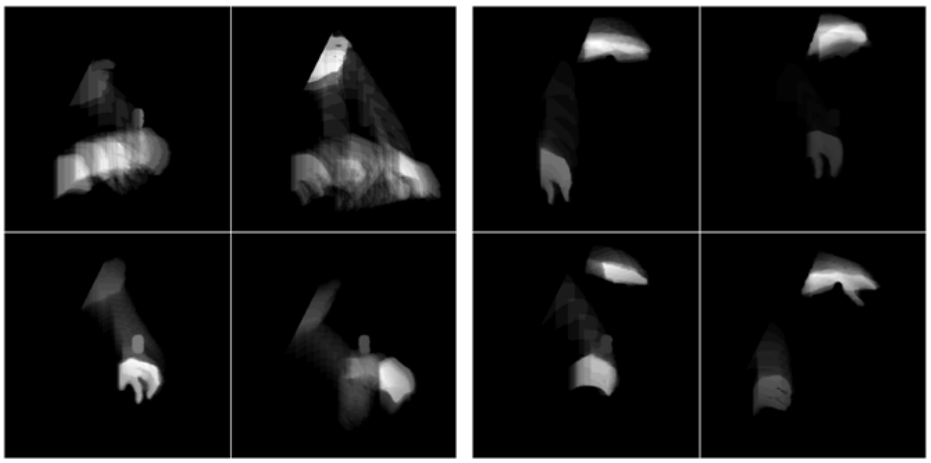


(b) Four classes of no shift gear related

Fig. 4.12 GEI patterns in level two



(a) Original GEI patterns



(b) Four classes of no shift gear related

Fig. 4.13 GEI patterns in level three

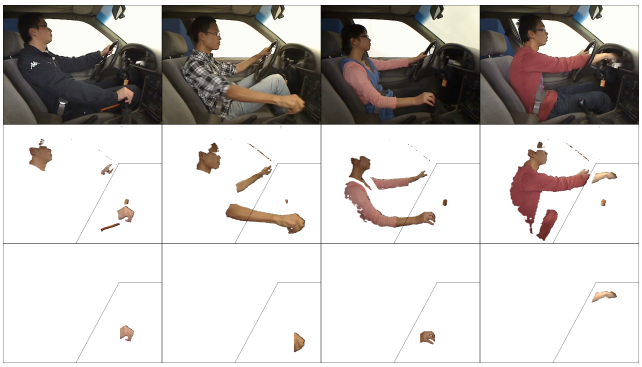


Fig. 4.14 Locating the right hand skin region in ROI

.

4.6.4 Level Four Classification

In the forth level of classification, the class of *only shift gear related* from level three can be further divided into two subclasses, namely *IntSG* and *Int.OpSG*, respectively. However, neither original GEI nor right hand skin region-GEI feature could give a satisfactory separation between these two subclasses. To solve the problem, we propose to exploit features that are more discriminative for hand motions. More specifically, if we summate the vertical projection values on a frame differencing image sequence , a behaviour containing *OpGS* will cause more movement around shift gear which makes larger projection value on the period of vertical axis corresponded to the shift gear area.

Therefore, we calculate the skin region frame differencing sequence and to summate the vertical projection to form a cumulative vertical projection histogram for classification. The detailed steps are as follows:

1. **Step 1:** For a given GEI belonging to the class of *only shift gear related*, find its corresponding original frame sequence.
2. **Step 2:** Transform the original sequence into a binary image sequence based on hand skin region segmentation proposed in previous subsection.
3. **Step 3:** Calculate the frame differencing image sequence from the binary image sequence.
4. **Step 4:** For each frame in the sequence, project its binary frame differencing image onto the vertical-axis and get the projection vector.
5. **Step 5:** Summate the projection vectors corresponded to each frame to form a vertical projection histogram.
6. **Step 6:** Use the histogram to represent a sequence after size normalization.

Fig. 4.15 Right hand skin sequence of video 7(frame 645–648)and their corresponding horizontal projection image

.

Fig. 4.15 shows the procedure to generate a horizontal projection histogram. The first row is four consecutive binary frames after right hand skin region segmentation in video 7. The second row corresponds to frame differencing image sequence. The third row shows

the horizontal projection histogram corresponding to the frame differencing image in the second row. The forth row is the cumulative horizontal projection histogram. The image of histogram in fourth row and fourth column of Fig. 4.15 is an example of a cumulative horizontal projection histogram which can be used to represent the motion among the four frames. However, the size of the histogram could be different, we normalize all the histogram to a fixed size.

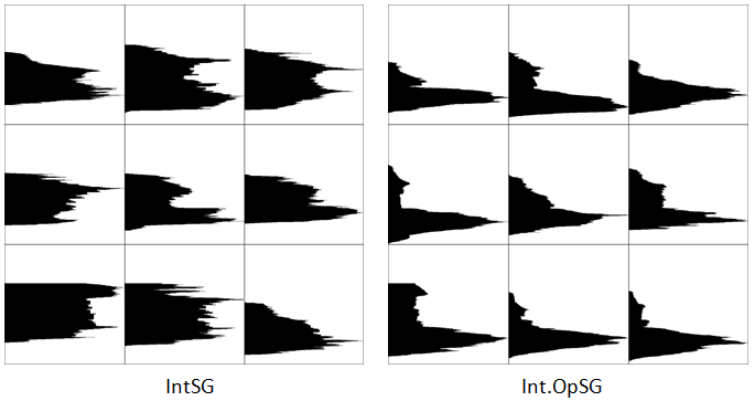


Fig. 4.16 Normalized horizontal projection histogram of the two classes in level four

.

Fig. 4.16 shows the normalized horizontal projection histogram of two classes. The lower side sharp peak in the histogram of *Int.OpSG* class represents operating the shift gear in the steering room which is the most distinguish feature by this method.

4.6.5 Additional Stage Classification on dangerous behaviour

The segmented driving motion clips are classified into eight classes based on their contents in previous four level hierarchal classifications. Dangerous driving behaviours, including eating, smoking and responding to a cell phone call, can all be described as the relative motion with reference to head. Therefore, we perform an additional stage of classification. Specifically, each frame in motion clips from the spatial oriented classes of *IntHd* and *IntHd.DB* will be re-examined and further be classified into two human perception oriented classes, that is *IntHd.phone* and *IntHdDB.eat*. In this additional stage, all the frames belonging to classes of *IntHd* and *IntHd.DB* will be further classified into another two classes as shown in following Fig. 4.17.

The first two rows are belongs to the class of *no hand in profile* while the bottom two rows are belongs to the class of *hand in profile*. The PHOG feature is extracted from every frame in every sequence in the *IntHd* class and *IntHd.DB* class. The PHOG feature is used to train and test a k-nearest neighbor(KNN) classifier with good performance. If any frame

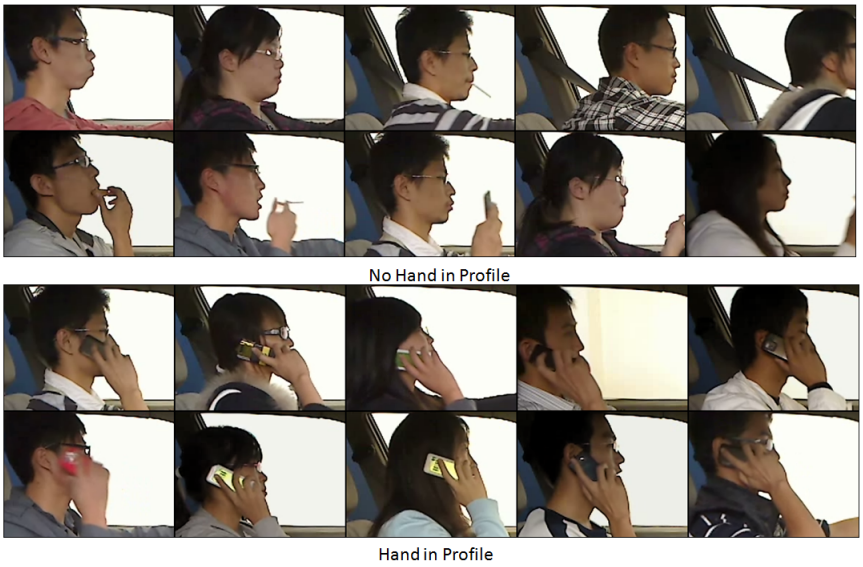


Fig. 4.17 Selected frames from the two classes: *no hand in profile* and *hand in profile*

.

from the two class of *IntHd* and *IntHd.DB* is labeled to be hand in profile, the behaviour sequence contains that frame is *IntDd.phone*, otherwise it is *IntHdDB.eat*.

4.7 Experiment

Experiments are carried out to verify the effectiveness of the proposed algorithm on the SEU driving database. This database consists of 20 sequences from 20 drivers conducting eight driving behaviours which have been introduced in section 7.2. The experiment was conducted on a Dell M6700 workstation with CPU i7 3740QM 2.7GHZ and the proposed algorithm are programmed using MATLAB. In the experiment, 20 videos from the original SEU dataset are first pre-processed to reduce the influence of illumination variation. After that, 527 motion clips are segmented from the original video by the algorithm discussed in section IV. Then eight different classes of motion clips are sent to the hierarchal classification system for training and classification. In order to evaluate the significance of hierarchal system, we also sent the data to a traditional non-hierarchal one-versus-eight classifier for comparison. Finally, we conduct an experiment on additional stage classification for exploring dangerous driving behaviour, one behaviour is *IntHdDB.eat*, the other is *IntDd.phone*. Meanwhile, in each level of the hierarchal system, the non-hierarchal system and the additional stage classification, we compare the classification performance by four commonly used classifiers, that is k-nearest neighour classifier(KNN), random forest classifier(RF), support vector machine classifier(SVM) and multi-layer perceptron classifier(MLP).

Table 4.3 Classification Accuracy

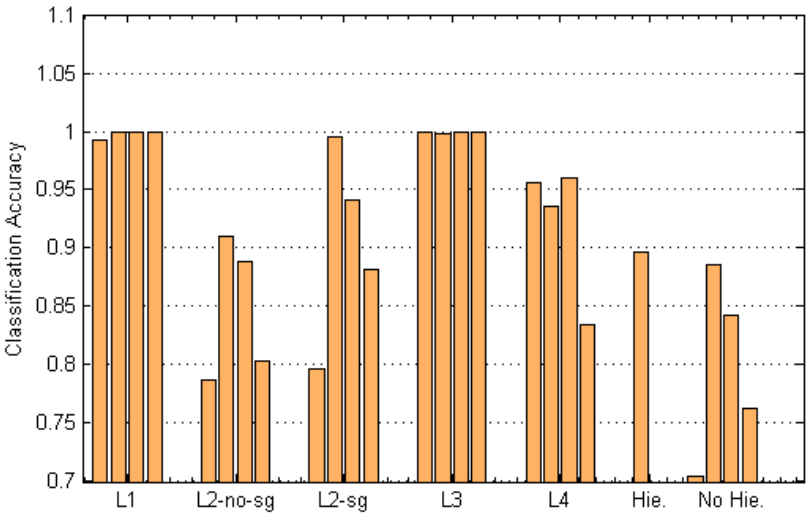
	Classification Accuracy(%)			
	KNN	**RF**	**SVM**	**MLP**
Level one	99.27	**99.87**	99.87	99.87
Level two-no-sg	78.68	**91.02**	88.86	80.32
Level two-sg	79.63	**99.48**	94.18	88.20
Level three	100	99.83	**100**	100
Level four	95.60	93.60	**96.00**	83.40
Hierarchal	**89.62**			
No hierarchal	70.47	**88.57**	84.31	76.22

4.7.1 hirachal and non-hirachal classification performance

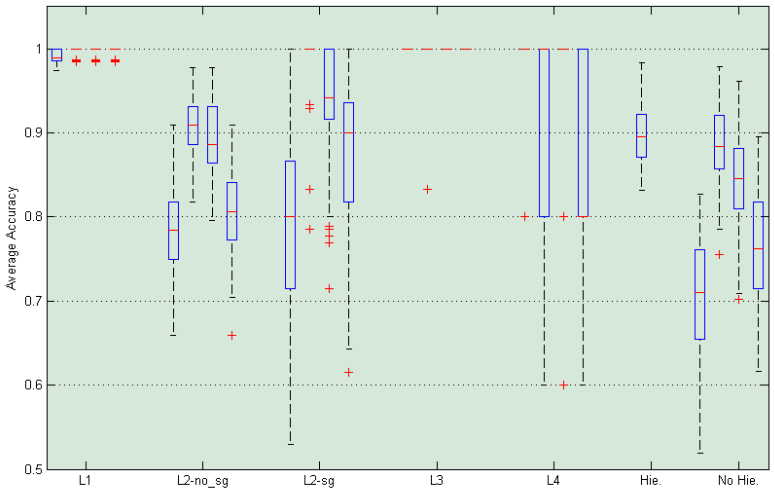
We chose a standard experimental procedure called the holdout approach to verify the driving behaviour recognition system. In the holdout experiment, 10% of the 20 videos, that is 2 videos, are randomly selected as the testing dataset, while the remaining 18 videos are used as the training dataset. The bar plot and box plot of average accuracy results from 100 runs are shown in Fig. 4.18a and Fig. 4.18b, respectively.

The ticks in the vertical axis represent level one classification (abbreviated as L1), level two no-shift gear related classification (abbreviated as L2-no-sg), level two shift gear related classification (abbreviated as L2-sg), level three classification (abbreviated as L3), level four classification (abbreviated as L4), hierarchal classification (abbreviated as Hie.), and non-hierarchal classification (abbreviated as No Hie.), respectively. Each tick except Hie. corresponds four classifier performances(from left to right), that is, KNN, RF, SVM and MLP, respectively. The Table 4.3 is the numerical results of the bar plot in Fig. 4.18a. Based on the performance shown in Table I, we choose RF in previous two levels and SVM in last two levels to form the hierarchal classification system, and the final classification accuracy is 89.62%. It is 1.05% improved compared to the non-hierarchal classification result of 88.57% which only applies GEI and PHOG in a one-versus-eight RF classifier. The improvement performance yields the significance of applying hierarchal system.

Moreover, to further evaluate the classification performance, confusion matrix is used to visualize the discrepancy between the actual class labels and predicted results from the



(a) Bar plot



(b) Box plot

Fig. 4.18 Plot of experiment result in the hierarchal system

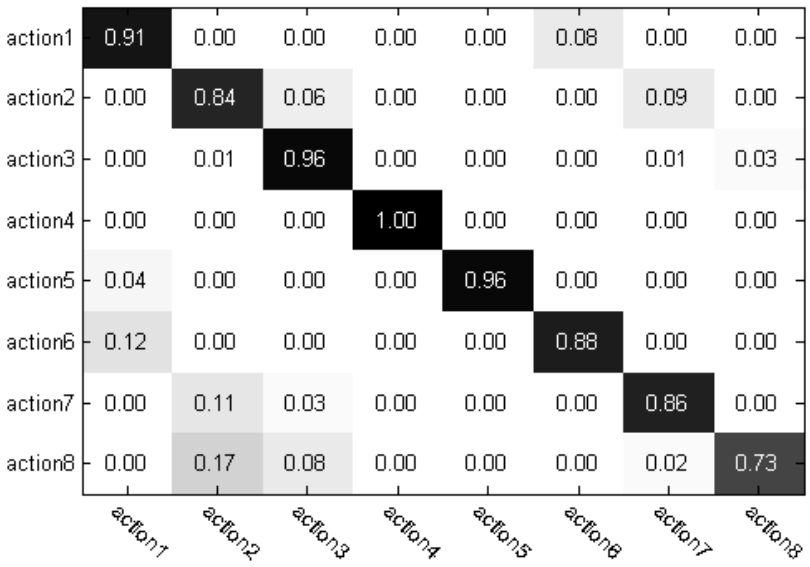
classification. Confusion matrix gives the full picture at the errors made by a classification model. The confusion matrix shows how the predictions are made by the model. The rows correspond to the known class of the data, that is, the labels in the data. The columns correspond to the predictions made 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.. The confusion matrices of the hierarchal system and non-hierarchal system are shown in Fig. 4.19a and Fig. 4.19b, respectively. In Fig. 4.19b, the accuracy of action 5 is only 19% and the action 5 is confused into action 2 with a rate of 59% and action 7 with a rate of 23%. However,as shown in Fig. 4.19a, the accuracy of action 5 is increased to 96% which means that 77% subsets of action 5 is closer to the others class in a non-hierarchal system by the feature of GEI.

4.7.2 Dangerous Behaviour Classification Performance

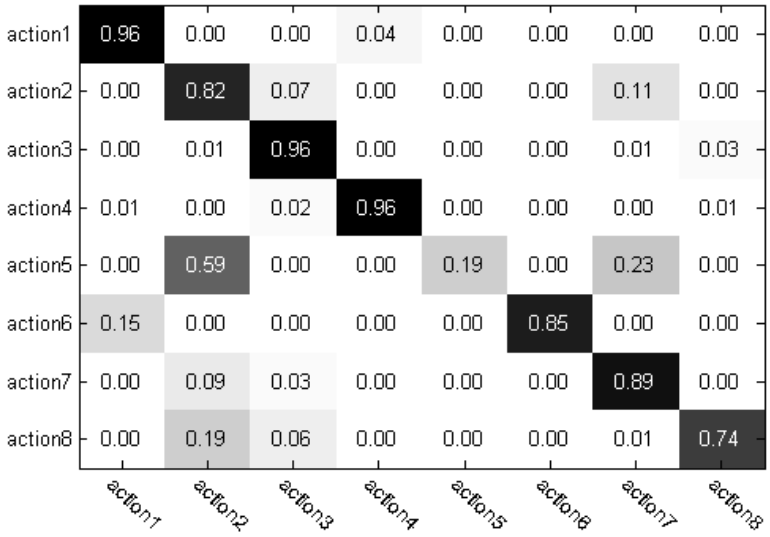
From the motion clips belong to the classes of *IntHd* and *IntHd.DB*, we extracted about 10 thousand frames. We manually labeled these 10 thousand frames into two classes, one is *No Hand in Profile* and the other is *Hand in Profile* which has been illustrated in Fig.21. We setup a holdout experiment based on randomly dividing the 10 thousand frames into a training dataset (90% of the 10 thousand feature vectors extracted from the 10 thousand frames) and a test dataset (10% of the 10 thousand feature vectors extracted from the 10 thousand frames). Using the holdout experiment approach, only the test dataset is used to estimate the generalization error. We repeat the holdout experiment 100 times by randomly splitting the 10 thousand features and recorded the classification results. The bar plot and box plot shows the classification performance among four commonly used classifiers in Fig. 4.20a and Fig. 4.20b. The result of classification rate of KNN, RF, SVM and MLP are 99.86%, 99.14%, 99.27% and 97.76%, respectively. The box plot in Fig. 4.20b further verifies that KNN classifier offers the best classification performance rate of the four classifiers. The confusion matrix of KNN shown in table 4.4 indicates that only 0.1% class I samples are mis-classified into class II while all class II samples are correctly classified.

4.7.3 Discussion

We treat the driving behaviour as temporal-spatial action instead of static images in other state-of-art approaches [55, 56, 164]. We first hierarchically recognise periods of motion under the framework of the action recognition. Then based on the prior knowledge of categories

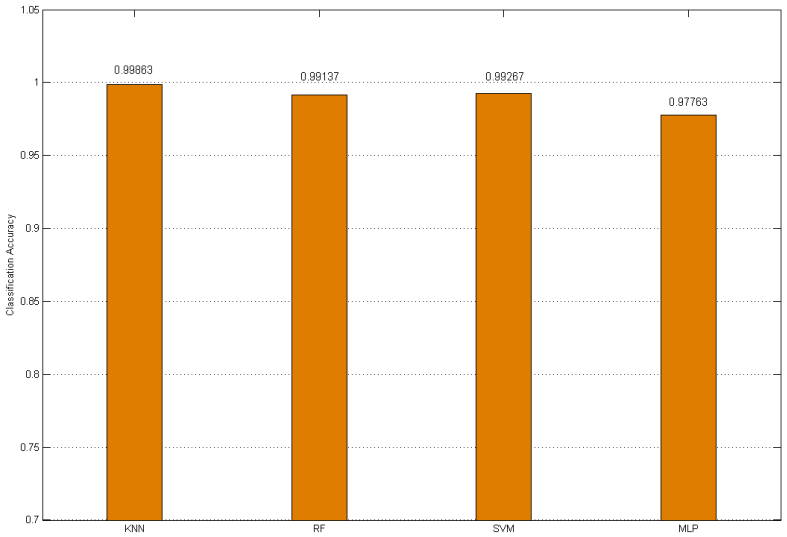


(a) Using hierarchal system

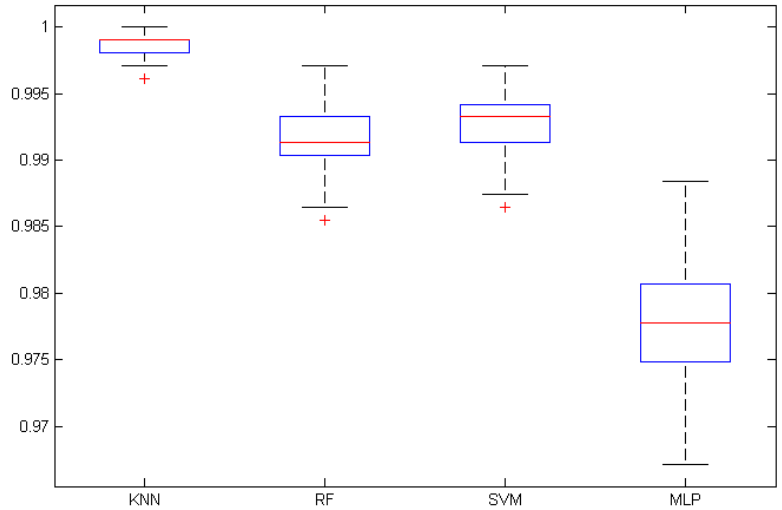


(b) Without hierarchal system

Fig. 4.19 Confusion matrix



(a) Bar plot



(b) Box plot

Fig. 4.20 Experiment result in the dangerous behaviour classification

Table 4.4 Confusion matrix for the result from KNN classifier.
(I)No Hand in Profile,(II)Hand in Profile

class	**I**	**II**
I	99.9%	0.1%
II	0	100%

of dangerous behaviour(eg. eating, smoking and responding a cellphone call), we further classify the head-related motion by the combination of PHOG and KNN with a high accuracy of 99.86%. In our hierarchal classification system, we achieve 96% accuracy rate for class 5(*IntHd*) and 88% accuracy rate for class 6(*IntHd.DB*). We roughly estimate the overall dangerous driving behaviour recognition rate by $(96\% * 99.86\% + 88\% * 99.86\%)/2 = 91.87\%$. This is a competitive performance which is closest to real application when compared to other state-of-art approaches.

We apply gait energy image representation combined with shifting of ROI, skin region analysis and projection histogram in different levels of our hierarchal classification system which proves: 1)improved overall performance(89.62%) compared to traditional flat classification(88.57%); 2)classification accuracy for each class increases to no less than 73%. The hierarchy of the system and the representation feature used in each hierarchy can be further improved in later extension of our work. In addition, we combined PHOG and KNN in the classification of dangerous behaviour which resulted in a high recognition rate of 99.86%. But eating and smoking are very similar behaviours and they are difficult to distinguish. They are labeled as the same class in our work. Further extension work is suggested to explore a better solution to distinguish eating and smoking.

4.8 Conclusion

This chapter addresses the importance of automatic understanding and characterization of driver behaviours in preventing motor vehicle accidents and presents a novel system for vision-based driving behaviour recognition. We verify our approach on the SEU driving dataset which includes activities of normal driving, responding to a cell phone call, eating and smoking. After pre-processing for illumination variations and motion clip segmentation, eight classes of behaviour are extracted for classification. By joint application of gait energy image, pyramid histogram of oriented gradients, hand skin-region segmentation

and the hierarchal classification, our overall accuracy is over 89.62%. While the overall accuracy increases 1.05% compared to non-hierarchal classification system, the individual classification accuracy for each class increases to no less than 73%. We also estimate two dangerous driving behaviour, that is *IntHd.phone* and *IntHdDB.eat*, with an overall recognition rate of 91.87%.

Chapter 5

Driving Posture Recognition by Convolutional Neural Networks

Driver fatigue and inattention have long been recognized as the main contributing factors in traffic accidents. Development of intelligent driver assistance systems with embedded functionality of driver vigilance monitoring is therefore an urgent and challenging task. This chapter presents a novel system which applies convolutional neural network (CNN) to automatically learn and predict pre-defined driving postures. The main idea is to monitor driver hand position with discriminative information extracted to predict safe/unsafe driving posture. In comparison to previous approaches, convolutional neural networks can automatically learn discriminative features directly from raw images. In our works, a CNN model was first pre-trained by an unsupervised feature learning method called sparse filtering, and subsequently fine-tuned with classification. The approach was verified using the Southeast University Driving-Posture Dataset, which comprised of video clips covering four driving postures, including normal driving, responding to a cell phone call, eating and smoking. Compared to other popular approaches with different image descriptors and classification methods, our scheme achieves the best performance with an overall accuracy of 99.78%. To evaluate the effectiveness and generalization performance in more realistic conditions, the method was further tested using other two specially designed datasets which takes into account of the poor illuminations and different road conditions, achieving an overall accuracy of 99.3% and 95.77%, respectively.

5.1 Introduction

With the ever-growing traffic density, the number of road accidents is anticipated to further increase. Unsafe and dangerous driving accounts for the death of more than one million lives and over 50 million serious injuries worldwide each year [12]. Finding solutions to reduce road accidents and to improve traffic safety has become a top-priority for many government agencies and automobile manufactures alike. It has become imperative to the development of Intelligent Driver Assistance Systems (IDAS) which is able to continuously monitor, not just the surrounding environment and vehicle state, but also driver behaviours.

Previous works using IDAS to prevent traffic accident can be categorized into two main streams of activities, which are: the vehicle-oriented approaches [58, 59] and the driver-oriented approaches [61, 50, 15, 54]. For the first one, driver vigilance is analyzed through vehicle behaviour including movement of steering wheel, pressure on the acceleration pedal, speed, deviations from lane position, response time against an obstacle braking, and etc. The main limitations of these approaches [58, 59] include their dependence on the shape of the road, the vehicle performance and the manner of driving. For the second approach, driver behaviour monitoring is based on the analysis of physiological and biomedical signals such as heart rate, brain activity, temperature, vascular activity and muscular activity. Such methods [61] rely on wearable sensors which decrease user experience and increase hardware cost. An alternative way to analyse driver behaviour is using a camera. There are three categories of vision-based approaches to automatically monitor the unsafe driver behaviour: (i) gaze and head pose analysis for the prediction of driver behaviour and intention [50, 51, 18, 190, 191], (ii) extraction of fatigue cues from driver facial image [15, 16, 192, 193], and (iii) characterization (in the context of safe versus unsafe driving behaviour) of driver body postures, including the positioning of arms, hands and feet [54–57]. Despite the encouraging performances under appropriate conditions, the proposed approaches share a common disadvantage of being ad hoc. Most of the vision-based methods follow a two-step framework: (i)extraction of hand-crafted features from raw data, usually with certain assumptions about the circumstances under which the data was taken, (ii)learning classifiers based on the obtained feature. Methods under such a framework cannot reach an optimal balance between the discriminability of the extracted features and the robustness of the chosen classifier. The reason is the uncertainty of what features are important for the task at hand since the choice of features is highly problem-dependent in real-world scenarios.

Recently, there has been growing interest in the development of deep learning models for various vision tasks [194–196]. Deep learning models generally features of learning multiple layers of feature hierarchies, with increasingly abstract representations extracted at each stage. Such learning machines can be trained using either supervised or unsupervised

approaches, and the resulting systems have been shown to yield competitive performance in speech recognition [197, 198], natural language processing [199], image classification [200–203], visual object detection [204], and other visual tasks [205–209].

One of the most successful deep learning models is the Convolutional Neural Network (CNN) [196, 200], a hierarchical multi-layered neural network able to learn visual patterns directly from the image pixels. In CNNs, small patches of the image (dubbed as a local receptive field) are inputted to the first layer of the hierarchical structure. Information generally passes on the different layers of the network, and at each layer trainable filters and local neighborhood pooling operations are exploited in order to produce salient features for the data observed. In addition, the method provides a level of invariance to shift, scale and rotation as the local receptive field allows the processing unit access to elementary features such as oriented edges or corners. It has been repeatedly proved that CNN is powerful to learn rich features from the training set automatically.

In this chapter, we apply Convolutional Neural Network architecture to represent and recognise driving postures, which aims at building high-level feature representation from low-level input automatically with minimal domain knowledge of the problem. Our work focus on the characterization of driving posture, with high-level features extracted hierarchically from raw input image. Each convolutional layer generates feature maps using sliding filters on a local receptive field in the maps of the preceding layer (input or max-pooling layer). The map sizes decrease layer by layer such that the extracted feature becomes more complex and global. Then, the output is inputted to a fully connected multilayer perceptron (MLP) classifier. The proposed approach was evaluated on the Southeast University Driving-Posture Dataset [192], demonstrating competitive performance.

The key contributions of this work can be summarized as follows:

1. To recognise driving posture, this chapter proposed to build a deep convolutional neural network in which trainable filters and local neighborhood pooling operations are applied alternatively to automatically explore salient features. Using CNN to learn rich features from the training set is more generic and requires minimal domain knowledge of the problem compared to hand crafted feature in previous approaches.

2. We using sparse filter [210] to pre-train the filters in our networks, with advantages including (i)acceleration of training for faster convergence, (ii) a better generalization performance. In addition, we setup experiment to evaluate the CNN architecture selection, with max-pooling and ReLU identified as better options for pooling operation and activation function, respectively.

3. The proposed approach was evaluated on the Southeast University Driving-Posture Dataset [192]. To account for the poor illuminations and different road conditions, we created two sets of video data, namely, the Driving Posture atNight Dataset and the Driving Posture inReal Dataset. The performances achieved on these three datasets are characterized by accuracies of 99.47%, 99.3%, and 95.77%, respectively.

The rest of the chapter is organized as follows. Section 7.2 presents an overview of our proposed method and the SEU driving posture dataset, while Section 7.2.2 gives a detailed introduction to the convolutional neural network followed by the training details in Section 5.4. Section 7.4 reports experiment results for the performance evaluation, followed by some conclusions presented in Section 7.5.

5.2 System Overview

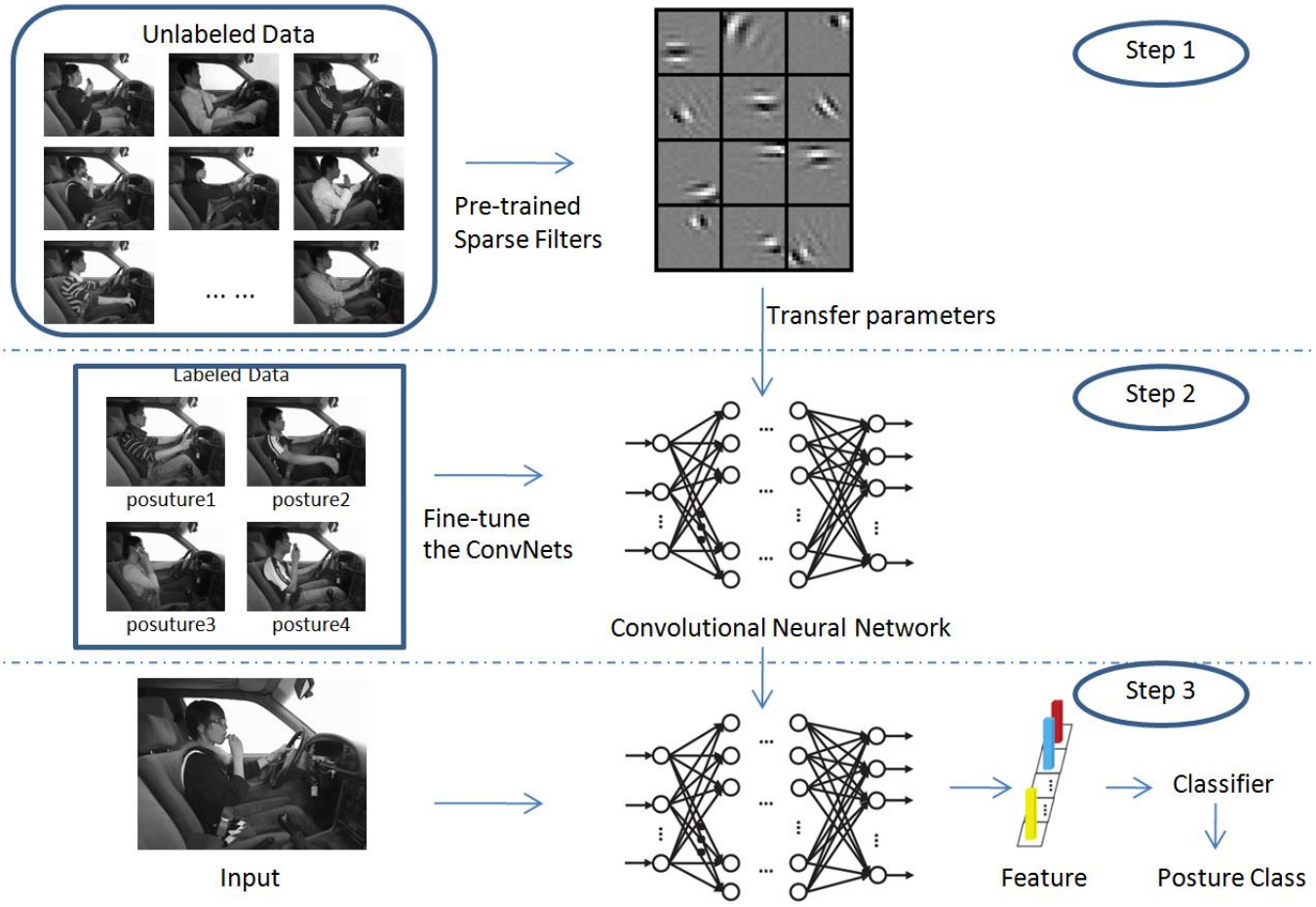


Fig. 5.1 The frameworks of our method.

Following the approach proposed by Zhao [192], our work also focus on the characterization of driving postures with reference to driver hand position. However, hand region is difficult to be accurately estimated due to the limitations of skin region segmentation algorithms and the variation of light condition. In our approach, we use the whole frame extracted from raw video instead of hand region as input. This is achieved by using convolutional neural network which can automatically learn discriminative feature representation directly from raw data.

To have an overview of the proposed approach, Fig.6.1 gives an illustration of the driving posture recognition system. The system comprises three steps: (i)unsupervised pre-training of the network with unlabeled data (ii)fine-tuning the network with four classes of the labeled data, (iii) feature extraction using the network from input for classification .

5.2.1 Southeast University Driving-Posture Dataset

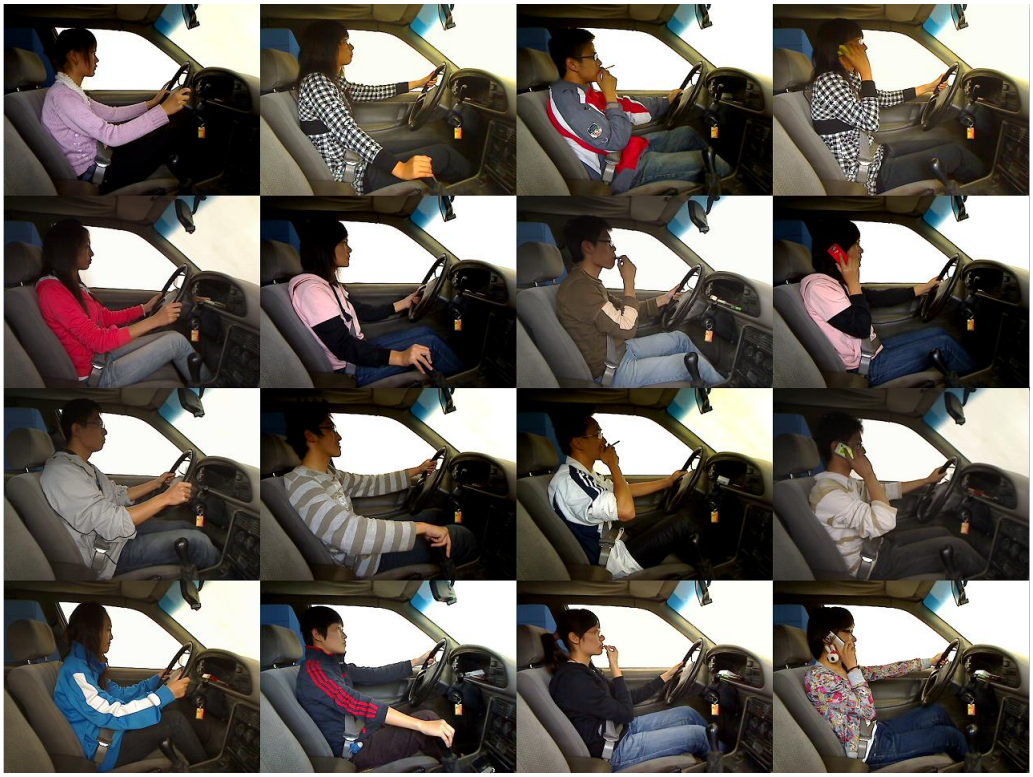


Fig. 5.2 Example images of from the SEU driving dataset. The first column is normal driving posture; The second column is the posture of operating the shift gear; The third column is the posture of eating or smoking; The forth column is the posture of responding a cell phone

To test the proposed driving posture recognition approach, the Southeast University Driving-Posture Dataset(SEU dataset) was used. This data was first created by Zhao [192].

Each video included in the dataset was obtained using a side-mounted Logitech C905 CCD camera under day lighting conditions with a resolution of 640×480. Ten male drivers and ten female drivers participated in the creation of the dataset. Each video was recorded under normal day light conditions. We extracted all frames in these videos and manually labeled four pre-defined postures including :

1. normal driving (posture1)
2. operating the shift gear (posture2)
3. eating and smoking (posture3)
4. responding a cell phone (posture4)

Some selected samples are shown in Fig.5.2. Each posture from (1) to (4) contains 46081, 12000, 18181, 16211 samples, respectively.

5.2.2 New Driving-Posture Dataset

To further evaluate the effectiveness and generalization performance of our approach, we built two different datasets, namely, Driving-Posture-atNight and Driving-Posture-inReal.

The first dataset, Driving-Posture-atNight, was recorded using a UWISH UC-H7225 infrared camera at night under low illumination conditions. Similar to SEU dataset, ten male and ten female participants performed the same four pre-defined postures, namely normal driving, operating the shift gear, eating/smoking, and responding a cell phone. All frames were extracted and manually labeled. Some examples are shown in Fig.5.4. They are divided into three subsets, that is, training set, validation set, and test set, with 24210, 1000, and 4200 samples, respectively.

To further evaluate the system performance in more realistic condition, the second dataset, Driving-Posture-inReal, was recorded using a Philips CVR300 car driving recorder. It was side-mounted in front window of a family car. In the dataset creation, five experienced drivers drove the car in turn on city road. The drivers were required to conduct two activities while driving, that is, eating cookies and responding cellphone calls. All frames were extracted and manually labeled into four classes as previous. Some examples are shown in Fig.5.4. They are divided into three subsets, that is, training set, validation set, and test set, with 14230, 1000, and 2500 samples, respectively.

Fig. 5.3 Example images of from the Driving-Posture-atNight dataset. Column 1: normal driving; Column 2: operating the shift gear; Column 3: eating or smoking; Column 4: responding a cell phone

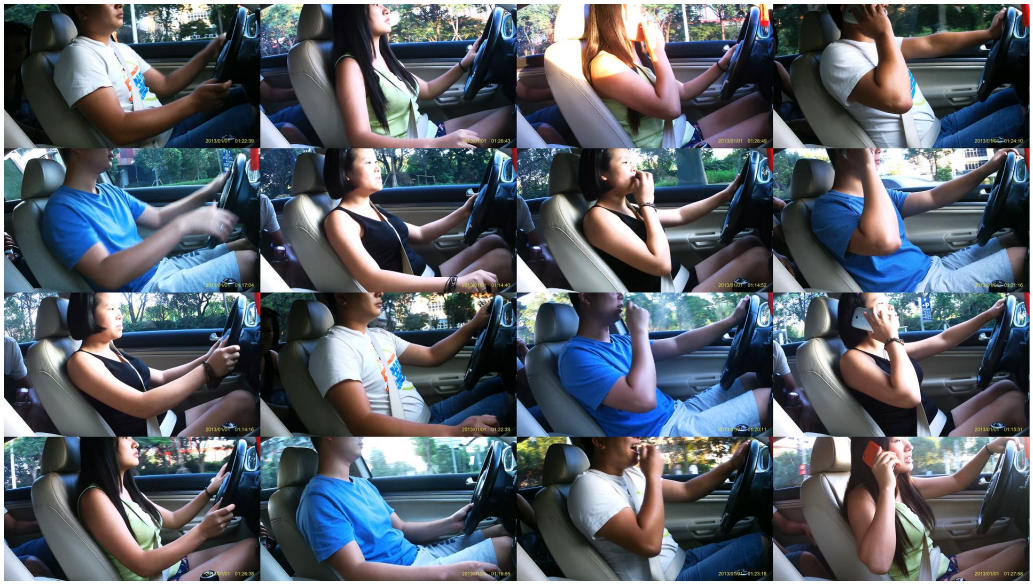


Fig. 5.4 Example images of from the Driving-Posture-inReal dataset. Column 1: normal driving; Column 2: operating the shift gear; Column 3: eating or smoking; Column 4: responding a cell phone

5.3 Deep Convolutional Neural Network Architecture

In this section , we will briefly overview the convolutional neural network architecture, with appropriate explanation of each building block.

5.3.1 Overall Network Architecture

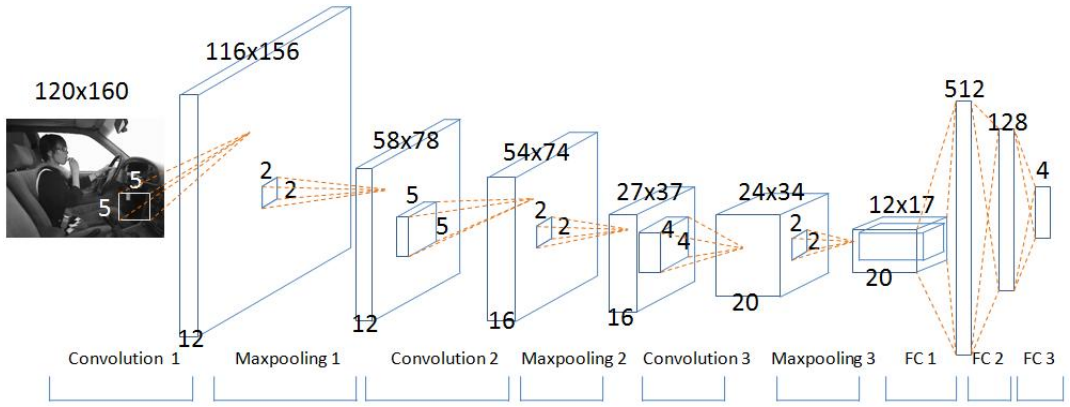


Fig. 5.5 The architecture of our unsupervised convolutional neural network. The network contains three stages, each of which is consisted of convolution layer, non-linear activation layer, local response normalization layer, and max-pooling layer. Only convolution and max-pooling layers which change the data size during operating, are illustrated here.

The overall CNN architecture is shown in Fig.7.5. The network consists of three convolution stages followed by three fully connected layers. Each convolution stage includes convolutional layer, non-linear activation layer, local response normalization layers and max pooling layer. The non-linear activation layer and local response normalization layers were not illustrated as data size was not changed in these two layers. Using shorthand notation, the full architecture can be donated as C(12,5,1)-$\tilde{A}$-N-P-C(16,5,1)-$\tilde{A}$-N-P-C(20,4,1)-$\tilde{A}$-N-P-FC(512)-$\tilde{A}$-FC(128)-$\tilde{A}$-FC(4)-$\tilde{A}$, where C(d,f,s) indicates a convolutional layer with d filters of spatial size $f \times f$, applied to the input with stride s. $\tilde{A}$ is the non-linear activation function, which uses ReLU activation function [211]. $FC(n)$ is a fully connected layer with n output nodes. All pooling layers P use max-pooling in non-overlapping 2×2 regions and all normalization layers N are defined as described in Krizhevsky et al. [200] and use the same parameters: $k = 2$, $n = 5$, $\alpha = 10^{-4}$, $\beta = 0.5$. The final layer is connected to a softmax layer with dense connections. The structure of the networks and the hyper-parameters were empirically initialised based on the previous work [212] using ConvNets. A cross-validation experiment was conducted to optimize the selection of network architecture, which will be elaborated in section 7.2.2.

5.3.2 Convolution Layer

The CNN is a biologically-inspired variant of the multi-layer perceptron (MLP), also known as "shared weight" neural networks introduced by LeCun [196]. From Hubel and Wiesel's early works on the cat's visual cortex [213], it is known that the visual cortex contains a complex arrangement of cells. These cells are sensitive to small sub-regions of the visual field, called a receptive field. The sub-regions are tiled to cover the entire visual field. These cells act as local filters over the input space and are well-suited to exploit the strong spatially local correlation present in natural images.

In the CNN architecture, the local receptive field (kernel or filter) is replicated across the entire visual field to form a feature map, which is known as convolution operation. The convolution operations share the same parameterization (weight vector and bias). Such a sharing of weights reduces the number of free variables, while increasing the generalization performance of the network. Weights (kernels or filters) are initialized as random and will be learned to be edge, color, or specific patterns' detectors.

The convolution operation is expressed as

$$y^j = f_{acti}\left(b^j + \sum_i w^{ij} * x^i\right) \tag{5.1}$$

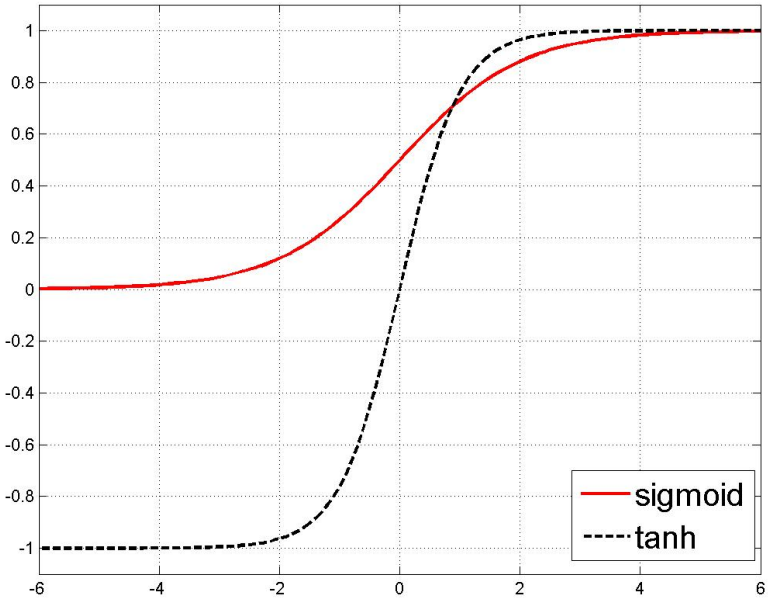
where x^i and y^j are the i-th input feature map and the j-th output feature map, respectively. w^{ij} is the weights of the convolution filter. $*$ denotes the convolution operation. b^j and $f_{acti}(\cdot)$ is the bias and activation function of the j-th output feature map, respectively. The non-linear activation function $f_{acti}(\bullet)$ is regarded as a single layer as opposite to a function intergraded in convolution layer, and will be introduced in more details in the following.

5.3.3 Nonlinear Activation Layer

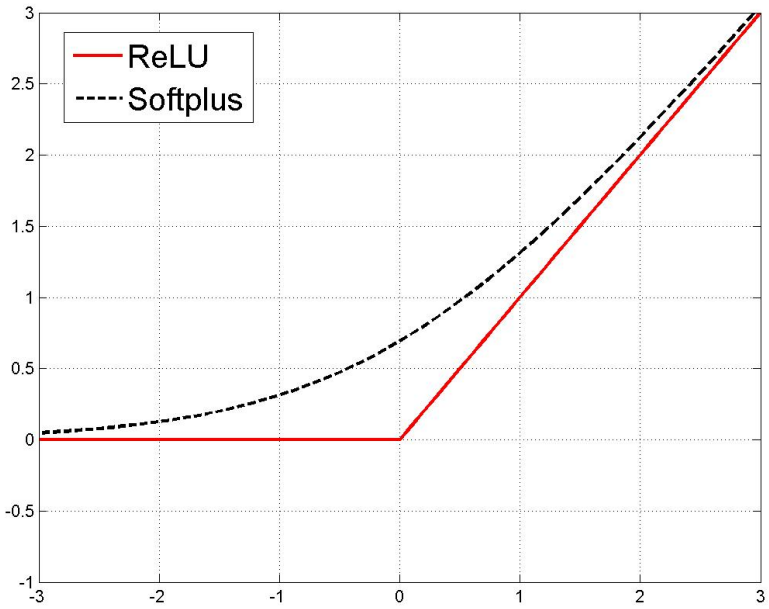
The convolution layer performs as a linear filter. To form a nonlinear complex model, nonlinear activation functions are needed to be passed, which transforms the input value nonlinearly to the output value of the neuron. Originally, sigmoidal activation functions were often used as the activation function in neural networks, for example, the sigmoid and hyperbolic tangent functions:

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{5.2}$$

$$tanh(x) = \frac{e^{2x}-1}{e^{2x}+1} \tag{5.3}$$



(a) Sigmoid vs Tanh



(b) ReLU vs Softplus

Fig. 5.6 Plots of four activation functions

Sigmoidal functions are bounded by minimum and maximum values as illustrated in Fig.5.6a. The sigmoidal function has the so-called saturation problem. In a saturated neuron, the gradient of the activation function approximates zero, which will diminish the gradient flow [214] to the lower layers in the neural network. This causes either the error rate decays or explodes exponentially to the lower layers, which making the training very slow.

Glorot et al. [211] proposed an alternative activation function, called rectifier linear unit(ReLU), as in Eq5.4, and compared it with a softplus activation function [215], a smooth version of the ReLU, as in Eq5.5.

$$ReLU(x) = max(x, 0) \tag{5.4}$$

$$softplus(x) = log(1 + e^x) \tag{5.5}$$

These two less saturated activation functions can be illustrated in Fig.5.6b. Recent CNN-based approaches [200–203, 208, 209] applied the ReLU as the nonlinear activation function for both convolution layer and full connection layer, generally with faster training speed as reported by [200].

5.3.4 Pooling Layer

An output map with local feature is extracted by previous linear convolution, adding bias and nonlinear activation. Due to the replication of weights in a convolutional layer, the detected features may still sensitive to the precise positions of the input pattern, which is harmful to the performance if it is followed by subsequent classification. To solve the problem, a reasonable approach is to pool the features, which has three major advantage: (i)Pooling in an efficient form of dimensionality reduction, which decreases feature maps' resolution and reduces computation for upper layers in CNN. It throws away unnecessary information and only preserves the most critical information [216] , (ii)Pooling makes activations in a neural network less sensitive to the specific structure of the neural network [217]. And it makes a network less sensitive to the exact location of the pixels, which results a form of translation invariance, (iii) Pooling summarizes the output of multiple neurons from convolutional layers with the essence of taking nearby feature detectors and forming local or global 'bag of features' [216].

Typical pooling functions are average and maximum, generally with the name of average-pooling (subsampling, downsampling, meanpooling) and max-pooling layers, as in Eq.5.6 and Eq.5.7, respectively.

$$y^{i}{}_{m,n} = \frac{1}{s^2} \sum_{0 \le \lambda,\mu < s} x^{i}{}_{m \cdot s+\lambda, n \cdot s+\mu} \tag{5.6}$$

$$y^{i}{}_{m,n} = \max_{0 \le \lambda,\mu < s} \left\{ x^{i}{}_{m \cdot s+\lambda, n \cdot s+\mu} \right\} \tag{5.7}$$

where each neuron in the i-th output map y^i pools over an $s \times s$ non-overlapping local region in the i-th input map x^i. Average pooling as the name suggest basically takes the arithmetic mean of the elements in each pooling region while max-pooling selects the largest element form the input.

5.3.5 Local Normalization Layer

There are two streams of normalization techniques being used in CNN architectures, including local contrast normalization [218, 219] and local response normalization [200].

The local contrast normalization is to eliminate the higher-order statistical dependencies in photographic images [220, 221], which is especially important in un-constrained environment machine vision tasks. Inspired by computational neuroscience models [222, 223], the local contrast normalization method was proposed as a layer in CNN architecture [218, 219]. Alternatively, Krizhevsky et al. [200] founds that their normalization method, local response normalization, increases generalization and decreases error rates in the ImageNet classification experiment. Denoting by $x^{i}{}_{m,n}$ the i-th input activity of a neuron after applying nonlinearity and pooling, the response-normalized activity $y^{i}{}_{m,n}$ is given by the following Eq.5.8 :

$$y^{i}{}_{m,n} = x^{i}{}_{m,n} \Big/ \left[k + \alpha \sum_{j=max(0,i-l/2)}^{min(L-1,i+l/2)} \left(x^{j}{}_{m,n}\right)^2 \right]^{\beta} \tag{5.8}$$

where the sum $\sum$ runs over l "adjacent" kernel maps at the same spatial position, and L is the total number of kernels in the layer. The constants k, l, α, and β are hyper-parameters whose values are determined using a validation set. The ordering of the kernel maps is arbitrary and determined before training begins. This sort of response normalization implements a form of lateral inhibition inspired by the type found in real neurons, creating competition for big activities amongst neuron outputs computed using different kernels. This scheme bears some resemblance to the local contrast normalization scheme, but would be more appropriately termed "brightness normalization", since no subtraction of the mean activity involved. As the impressive performance of ImageNet benchmark classification [200] and for fair comparison, the local response normalization is frequently used in recent

CNN architechture [200–203, 208, 209]. Hence, we adopted a same normalization as in Equ.5.8,with the same parameters as in [200].

5.3.6 Full Connection Layer

The four layers, convolution layer, nonlinear layer, pooling layer and normalization layer, are combined hierarchically to form a convolution stage (block). Generally, the raw input image will be passed through several convolution stages for extracting complex descriptive features in conventional CNN architecture. In the output of topmost convolution stage, all small-size feature maps are concatenated into a long vector. Such a vector plays the same role as hand-coded features and it is fed to a full connection layer. A standard full connection operation follows the conventional multiple layer perceptron (MLP), which can be expressed as

$$y^j = f_{acti}(\sum_{i=1}^{m} w^{ij} x^i + b) \tag{5.9}$$

where x^i is the i-th neuron of a m-dimension input vector, y^j is the j-th neuron of a n-dimension output vector, w is a $m \times n$ weight matrix, b is called bias units which correspond to the intercept term, and $f_{acti}(\cdot)$ is the activation function as introduced in section 5.3.3.

5.3.7 Output Layer

As the fully connected layers receive feature vector from the topmost convolution stage, the output layer can generate a probability distribution over the output classes. Toward the purpose, the output of the last fully-connected layer is fed to a K-way softmax (where K is the number of classes) layer, which is same as a multi-class logistic regression. If we denote by x^i the i-th input to the output layer, then the probability of the i-th class, p^i, is calculated by the following softmax function in Eq.5.10.

$$p^i = exp(x^i) / \sum_{j=1}^{K} exp(x^j) \tag{5.10}$$

5.4 Training Procedure

In this section, we first briefly outline the back-propagation algorithm for the CNN training. Then the practically important issue, i.e., the pre-training, will be subsequently discussed.

5.4.1 Learning through Back-propagation

In CNN training, the famous Back-propagation algorithm [224] is often used to propagate errors in the network architecture. Training is composed of two steps: (i)feed forward the training data through the network till the final output layer, and finally calculate the error or a loss function value (ii)back propagate the error/loss layer by layer from top to bottom, and update the weights in respective layers based on the back propagated errors.

An appropriate loss estimation for the output from the final layer is the cross-entropy loss [225], which has faster training process than the conventional mean square error. In the output layer, the cross-entropy loss function is given by :

$$L = -\sum_{j=1}^{K} [t^j log(p^j) + (1-t^j) log(1-p^j)] \tag{5.11}$$

where x^j and p^j are the j-th input and output respectively as defined in section 5.3.7, K is the number of output neurons (class) and t^j is the j-th class's one-per-class encoded target label.

When performing back propagation, the first step is to calculate the loss gradient by partial derivative as follows:

$$\begin{aligned} \delta^j = \frac{\partial L^j}{\partial p^j} \cdot \frac{\partial p^j}{\partial x^j} &= (-\frac{t^j}{p^j} + \frac{1-t^j}{1-p^j}) \cdot p^j(1-p^j) \\ &= \frac{p^j - t^j}{p^j(1-p^j)} \cdot p^j(1-p^j) \\ &= p^j - t^j \end{aligned} \tag{5.12}$$

where δ^j is the loss gradient in the output layer, which will be back propagated to the topmost full connection layer as an error.

In an l-th full connection layer, an error ${\delta^j}_{l+1}$, is back propagated from an upper $(l+1)$-th layer, then the error ${\delta^i}_l$ and the weight gradient $\Delta {w^{ij}}_l$ in this layer is given by Eq.5.13 and Eq.5.14, respectively.

$${\delta^i}_l = \sum_{j=1}^{n} {w^{ij}}_l \cdot \Delta {f_{acti}(\cdot)}_l \cdot {\delta^j}_{l+1} \tag{5.13}$$

$$\Delta {w^{ij}}_l = {x^i}_l \cdot \Delta {f_{acti}(\cdot)}_l \cdot {\delta^j}_{l+1} \tag{5.14}$$

where w^{ij} is the $m \times n$ weight matrix, n and ${x^i}_l$, is the total number of the output neuron and input neuron when feed-forwarding,respectively. $\Delta {f_{acti}(\cdot)}_l$ is the gradients of the nonlinearity

activation function, the error $\delta^i{}_l$ will be back propagated to the lower $(l-1)$-th layer. More details about the CNN training procedure can be referred to [226].

5.4.2 Pre-training

CNN architecture strongly depends on large amounts of training data for good generalization. When the amount of labeled training data is limited, directly training a high capacity CNN from only a few thousand training images is problematic. Researches [227] have shown an alternative solution to alleviate the CNN requirement, that is, choosing an optimised starting point which can be pre-trained by transferring parameters replacement of random initialization.

The first layer of many deep neural networks trained on natural images learns features similar to Gabor filters and color blobs. It seems that the first-layer features appear not to be dependant on a particular dataset or task, but general in that they are applicable to common visual datasets and tasks. Therefore, an intuitive hypothesis was proposed in [228] that features must eventually be transformed from general to specific layers by layers in a conventional deep neural network, which may provides the theoretical support for the pre-training by transferring parameters.

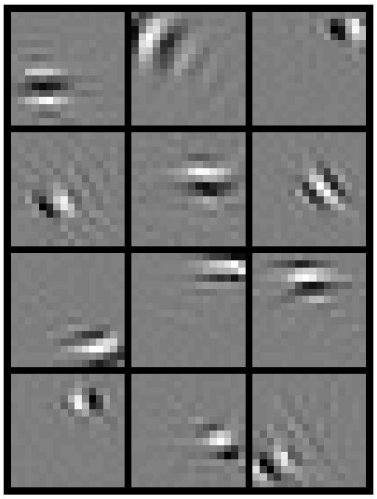


Fig. 5.7 The unsupervised pre-trained sparse filters of the first convolution layer

Following the guideline of pre-training methodology as discussed above, we introduce the Sparse Filtering [210] to learn the filter in each convolution layer of our network. Sparse filtering is an unsupervised feature learning algorithm which optimizes for the sparsity in the feature distribution and avoids explicit modeling of the data distribution. Compared to other unsupervised techniques such as Autoencoder [229], sparse coding [230] and Sparse RBMs

[231, 194], sparse filtering is easy to implement and hyperparameter free. In experiment, we randomly extracted patches with size 5×5 from the video dataset and the sparse filtering method is utilized to learn the filters. The learnt filters of the first layer are shown in Fig.5.7. As we expected, the filters mainly preserve the edge, point and junction information of the driver. Fig.5.8 shows a testing error versus epoch, illustrating the effect of using pre-train. From the figure, it is clear that pre-training improves convergence speed about 20 epoches.

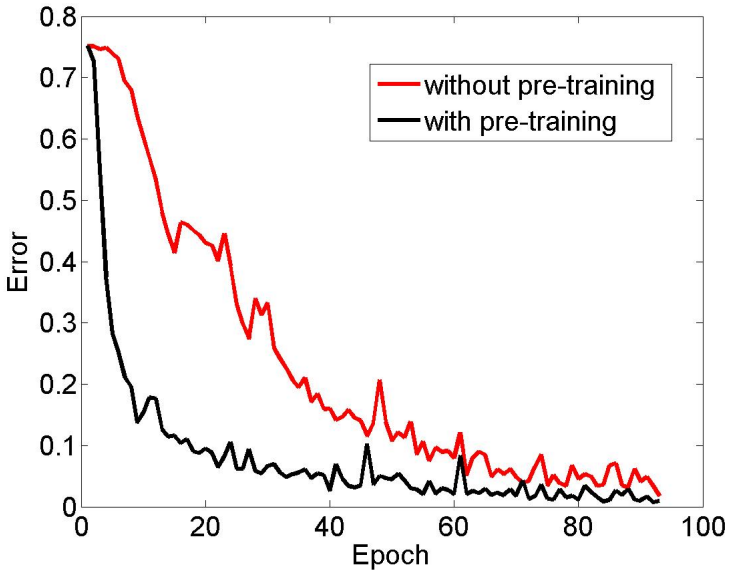


Fig. 5.8 Improvement of using pre-train.

5.5 Experiment

In this section, we first introduce some implementation details in section 6.4.1. Then three evaluation experiments to select the CNN structure, activation function, pooling method and other hyper-parameters are presented in section7.2.2. The CNN model is applied to verify the effectiveness of the proposed algorithm on the SEU driving posture database and our own driving posture database as described in section 5.5.3 and section 5.5.4. Finally, some published approaches with hand-coded features are compared in section 5.5.6.

The SEU driving posture dataset contains twenty video clips. All experiment are conducted using five-fold cross-validation. The original twenty videos are randomly divided into five folds, each containing four video clips. In five-fold cross-validation , one of the folds is retained for testing while the remaining four folds for training. The cross-validation process will then be repeated five times, which means that each of the fold will be used exactly once

as the testing data. In each training, 5% of the training data are randomly selected from each driver as validation data. The selection of validation data is consistent with dataset distribution.

5.5.1 Implementation Detail

We train our models using stochastic gradient decent with a batch size of 128 examples, momentum of 0.6, and weight decay of 0.0005. The learning rate is initialized as 0.01 for all trainable layers and adapted during training. How to adaptively control the learning rate within a reasonable range is an important issue in CNN learning. A too small learning rate makes the convergence rather slow, while a too big learning rate would make the network parameters vibrated. We proposed an adaptive learning rate by monitoring the loss function value and the validation error. We divide the learning rate by 2 if the loss value and validation error stops decreasing. To further prevent possible overfitting, we apply dropout and data augmentation as performed in [200].

The experiments were implemented on our GPU CNN package written in C++ language based on NVIDIA CUDA and cuDNNv2. Our experiments are conducted on a NVIDIA GTX Titan GPU and a 4-core Intel(R) Core i7-3770 3.40-GHz computer.

5.5.2 Architecture Selection

We conducted three experiments to select the network architectures, including convolution layer capacity, nonlinear activation function, and pooling method. Initially, a default architecture was chosen, denoted as C(9,5,1)-$\tilde{A}$-N-P-C(12,5,1)-$\tilde{A}$-N-P-C(15,4,1)-$\tilde{A}$-N-P-FC(512)-$\tilde{A}$-FC(128)-$\tilde{A}$-FC(4)-$\tilde{A}$, with the notation meaning explained in section 6.3. Then cross-validation was applied to optimize all the hyperparameters one by one.

Architecture Capacity

According to learning theory, if the architecture has too much capacity, it tends to overfit the training data with poor generalization. If the model has too little capacity, it underfits the training data, and both the training error and the test error are high. The capacity here means (i)the depth of the deep convolutional neural networks, and generally stands for the number of the repeated convolution stages, and (ii)the width of the network, which means the filter numbers in each convolutional layer.

The depth mainly depends on the size of raw input image and the complexity of the task. Compared to those 10,000 face image classification task [209], the complexity of our work is

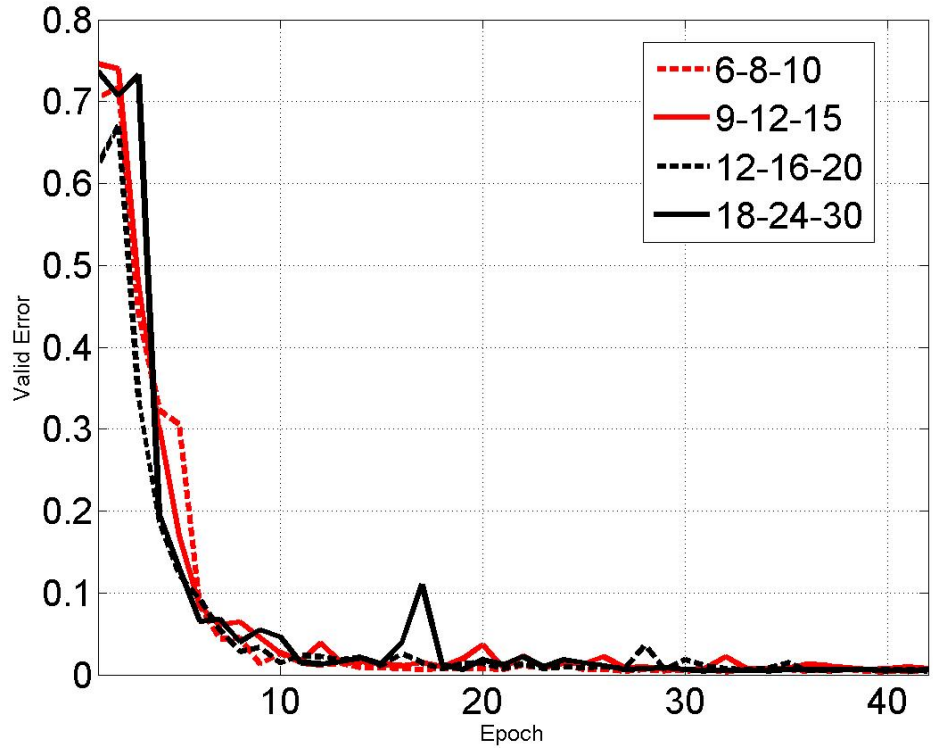
relatively lower. The original image is with size 480×640. To save computation resources, we downsized the images to 120×160, without losing discriminative information to classify the postures. We selected the filter size of 5×5 for first and second convolution layers ,and the filter size of 4×4 for the third convolution layer. Each subsequent pooling layer uses a non-overlapping 2×2 kernels which is same as most CNN-based approaches. Therefore, the size of the output feature map is 12×17 after three stages of convolution as shown in Fig.7.5. The selection is based on our empirical finding that the resolution of feature map is small enough to terminate convolution.

With regard to the width of the network, we empirically set the filter size as 9, 12 and 15 in the three convolution layers, respectively. The subsequent pooling layers will decrease resolution of the feature maps. To prevent the information from being lost too quickly, the filter size will generally increased. Here, we setup experiment to compare with three other groups of filter number, that is 6-8-10, 12-16-20 and 18-24-30, where the notation of x-y-z means x number filters, y number filters and z number filters in the first, second and third convolution layers, respectively. The validation error and testing error with epoch are shown in Fig.5.9a and Fig.5.9b,respectively. From the figure, the network comparing four combinations with different number of filters, converges since 10 epoches and is stable after 20 epoches. However, the black dash line which stands for the filter number group of 12-16-20, exhibits a minor early convergence compared to other three groups. Hence, we choose this group as our filter numbers in each convolution layer.

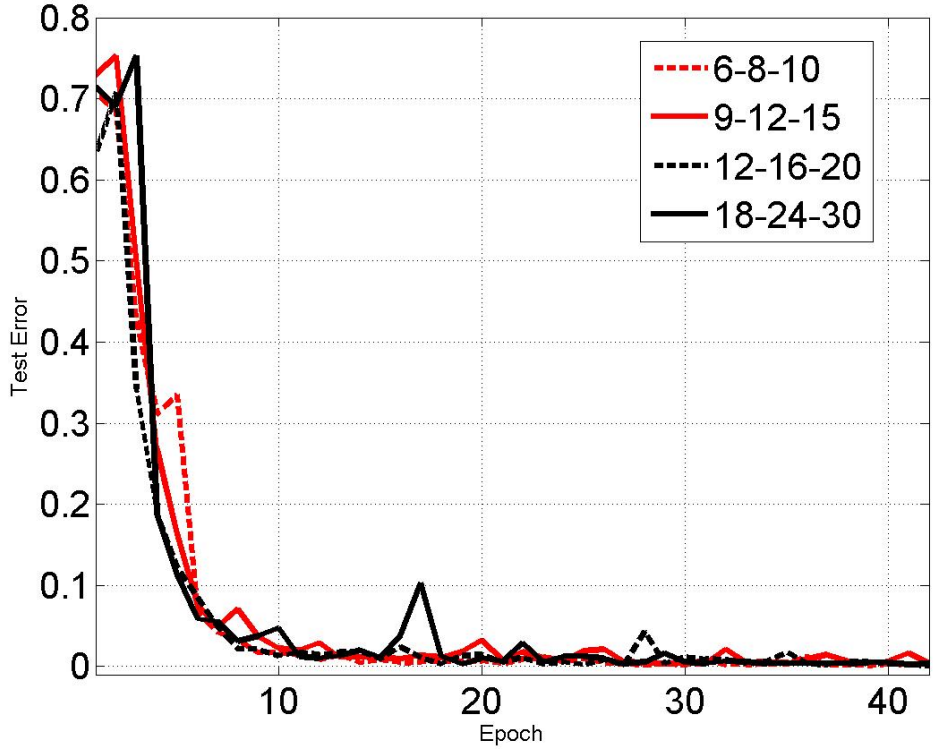
Nonlinear activation function and pooling method

In this section , we evaluate the nonlinear activation function and pooling method in the CNN network. The ReLU has been reported faster convergence than conventional sigmoidal functions, we re-implemented these activation functions, with the validation error with epoch illustrated in Fig.7.6c. In the figure, we plot the first ten epoches, which demonstrates a higher training speed of ReLU and softplus. ReLU has been reported with lower test error than softplus in [211], which, however, has not been observed in our experiments.

Typical pooling methods includes average-pooling and max-pooling. Average pooling method pools out the average value within a given reception field, while max-pooling method pools out the max value. The validation error with epoch for these two pooling methods are plotted in Fig.5.10b. It is obvious that max-pooling yields a better performance.

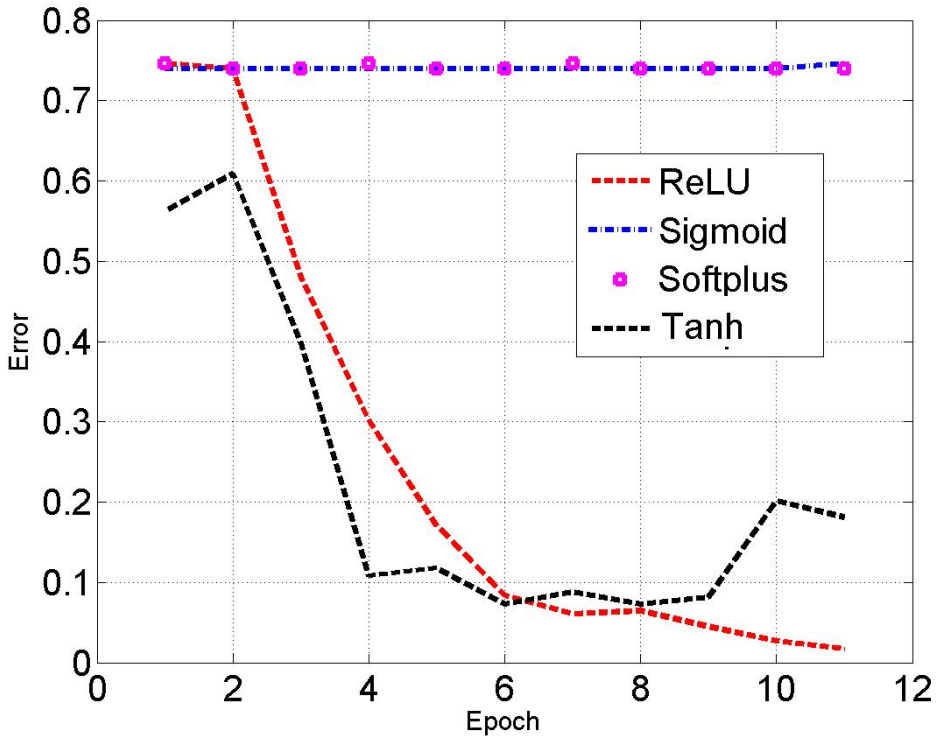


(a) Validation Error

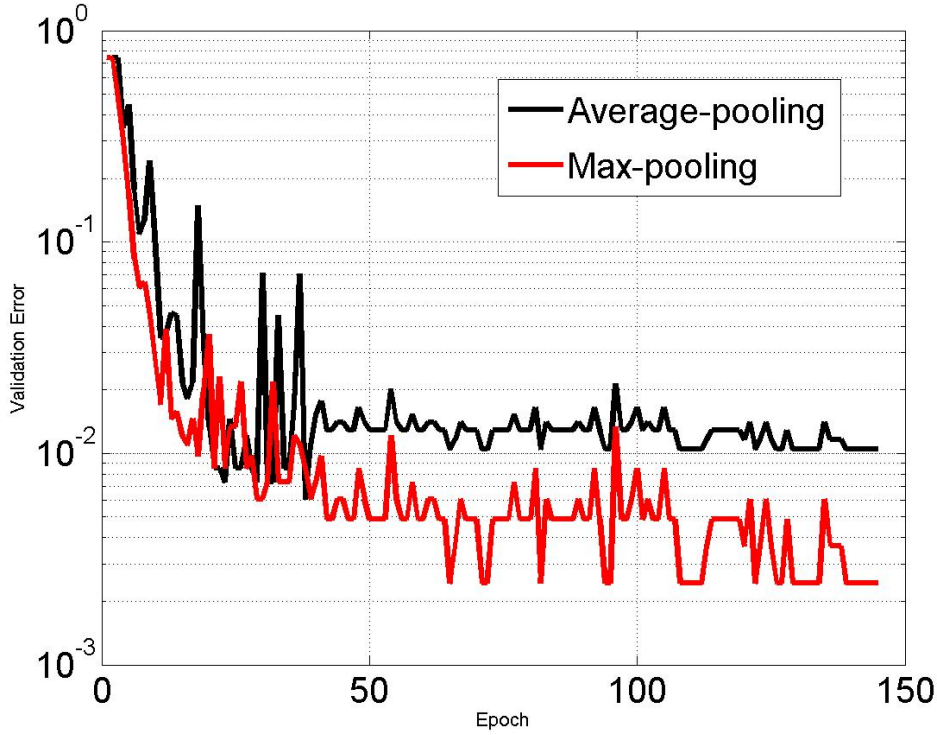


(b) Testing Error

Fig. 5.9 Selecting filter numbers in each convolution layer



(a) Validation error of four activation functions



(b) Validation error of two pooling method

Fig. 5.10 Selecting activation function and pooling method

5.5.3 Evaluation with SEU Driving-Posture Database

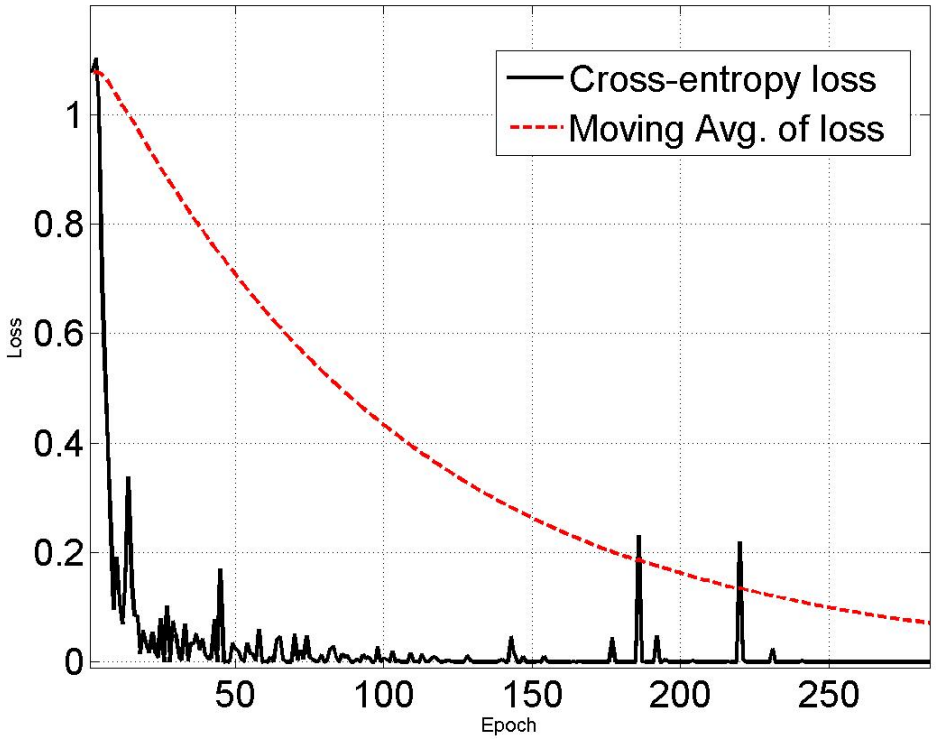
With the selected architecture described above, we carried out experiments with convolutional neural network for training and testing using the SEU dataset. The experiment procedure is five-fold cross-validation. The first time result is illustrated in Fig.5.11. In Fig.5.11a, the black line shows the global loss, the red line is the moving average of the black line ($y_n = x_{n-1} + \alpha \times x_n$, where x_{n-1} is the global loss in $(n-1)$-th epoch, y_n is the moving average result of global loss in (n)-th epoch, and α is the learning rate). The moving average result of the global loss shows a clearer decreasing trend. In Fig.5.11b, the training error, validation error and testing error are plotted in black, blue and red respectively. The training error are evaluated every five epoches. The network is convergent after 250 epoches.

Table 5.1 Confusion matrix for the cross validation result

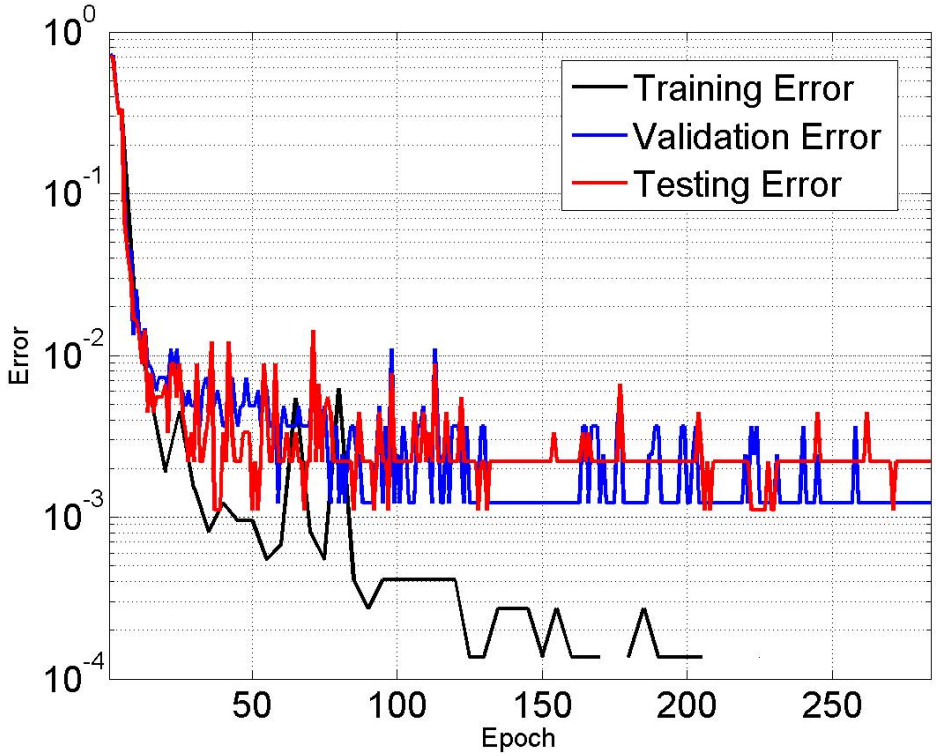
class	pose1	pose2	pose3	pose4
pose1	99.47	0	0.53	0
pose2	0	100	0	0
pose3	0	0	100	0
pose4	0	0	0.45	99.55

After the five cross-validation experiments, to further evaluate the classification performance, 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 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. The confusion matrices of the our results are shown in Table6.1. The accuracy for normal driving , operating shift gear, eating/smoking, and responding a cellphone are 99.47%, 100%, 100% and 99.55%, respectively.

From the confusion matrix, pose2 and post3 gain 100% classification accuracy, while around 0.5% of pose1 and pose4 are misclassified to pose3. Fig 5.12 shows some misclassification samples. In Fig5.12a, the hand position in misclassified pose1 is very close to mouth.



(a) Loss value, $\alpha = 0.01$



(b) Errors

Fig. 5.11 Plots of four activation functions

(a) pose 1 misclassified to pose 3

(b) pose 4 misclassified to pose 3

Fig. 5.12 Misclassification Analysis

This is the reason that they are misclassified to pose2(eating or smoking). In Fig5.12b, the hand position is in the middle of head and string wheel. These samples do not belong to any of our defined posture class. The error sample in pose 4 cause the misclassification.

5.5.4 Evaluation on the New Driving Posture Database

The SEU dataset was recorded under normal lighting conditions in a still vehicle. Low illumination and other realistic driving conditions were not considered. We further evaluated the effectiveness and generalization performance of our own driving posture database. It consists of two separated datasets, namely, Driving-Posture-atNight and Driving-Posture-inReal. The first one was recorded at night, while the second one was recorded in real driving conditions. Experiments were conducted in the similar procedure as with the SEU dataset. The confusion matrices of evaluation experiment on the Driving-Posture-atNight and the Driving-Posture-inReal are shown in Table5.2 and Table5.3, respectively.

Table 5.2 Confusion matrix for experiment using Driving-Posture-atNight

class	pose1	pose2	pose3	pose4
pose1	99.75	0	0.25	0
pose2	0	100	0	0
pose3	0	0	99.30	0.70
pose4	0	0	0.50	99.50

Table 5.3 Confusion matrix for experiment using Driving-Posture-inReal

class	pose1	pose2	pose3	pose4
pose1	95.77	0	2.70	1.53
pose2	0.22	99.15	0.63	0
pose3	1.35	0.82	96.23	1.60
pose4	0.55	0	1.90	97.55

5.5.5 Comparison with Other Methods

To provide a comprehensive performance evaluation, we conducted further experiments to compare the proposed CNN approach with several conventional methods using hand-crafted features. Specifically, the compared approaches include: (i) the method proposed in [192], which represents the posture pattern by contourlet transform on skin region; (ii) the method proposed in [232], which extracts feature using mutiwavelet transform method from skin region; (iii) an classifier ensemble method [233]; (iv) Bayesian classifier approach [234]; (v) PHOG descriptor [31] followed by Support Vector Machine, and (vi) SIFT descriptor [235] followed by Support Vector Machine.

Table 5.4 Classification Accuracy compared with other six approaches

	pose1	pose2	pose3	pose4	Avg.
Baseline[192]	97.70	87.55	85.95	89.30	90.63
WT[232]	97.52	92.77	88.99	83.02	89.23
RSE[233]	99.95	91.20	99.20	87.42	94.20
Bayes[234]	94.82	95.20	98.26	92.77	95.11
PHOG+SVM	**99.83**	88.71	89.12	73.20	91.56
SIFT+SVM	99.40	93.52	94.55	91.21	96.12
Proposed	99.47	**100**	**100**	**99.55**	**99.78**

For fair comparison, we re-implemented the methods proposed in [192, 232–234] and carried out experiment on other two popular vision descriptor approaches [31, 235]. The approach proposed in [192] and other three approaches [232–234] recognise driving postures using hand and head position. These methods are sensitive to illumination conditions and different race skin colors, which are located by skin color segmentation. In other two approaches [31, 235], feature are extracted from the whole frame without any pre-processing. The cross-validation experiment procedure was repreated in the same way as we did in section 5.5.3. The results are shown in Table 5.4. It clearly demonstrates that our approach outperforms all of the methods compared.

5.5.6 Discussion

The proposed CNN approach has been tested on the SEU database and our own database, demonstrating the effectiveness in day time/night conditions and the generalization performance in real driving environments. In the comparison experiment, the CNN model offers better performance than other approaches with hand-engineered features. The end-to-end

learning model is able to learn the temporal cues and discriminative representation from raw images. However, there exists some limitations with the proposed method. First, training takes days even it runs on GPUs. The algorithm takes high computation resource which makes it difficult to be applied in some conditions with common hardware architecture, e.g., off-line embedded system. Second, training a CNN needs a large amount of unconstraint data which is also difficult in some situations.

5.6 Conclusion

This chapter addresses the importance of automatic understanding and characterization of driver behaviours in the scenario of reducing motor vehicle accidents, and presents a novel system for vision-based driving behaviour recognition. We verified our approach on the SEU driving dataset which includes postures of normal driving, operating the shift gear, eating or smoking, and responding a cell phone. The proposed approach applied deep convolutional neural network, which learns feature from raw image automatically. We have described the details of each layer in our network and evaluated the selection of networks architecture and hyper-parameters. The final results demonstrate better performance than conventional approaches with hand-coded features, achieving an overall accuracy of 99.47% on the SEU dataset, 99.3% on the Driving-Posture-atNight dataset, and 95.77% on the Driving-Posture-inReal dataset.

Chapter 6

Recognizing Driver Inattention by Convolutional Neural Network

Driver inattention have long been recognized as the main contributing factors in traffic accidents. Development of intelligent driver assistance systems with embedded functionality of driver vigilance monitoring is therefore an urgent and challenging task. This chapter presents a novel system which applies convolutional neural network to automatically learn and predict the state of driver's eye, mouth and ear. The main idea is to predict driver fatigue and distraction by analysing these states. In our works, a CNN model was trained with six classes of labeled data. The Approach was verified using self-specified Driving Dataset, which comprised of video clips covering six classes, including normal driving, responding to a cell phone call, eating and smoking. Experiment shows our method achieves the promising performance with a overall accuracy of 95.56%.

6.1 Introduction

In this chapter, Face++ Research Toolkit [27] are first applied to localize the facial landmark [28] on the driver's face. Then, image patches including eye, mouth and ear are separately segmented from the driver's face based on landmark points. The convolutional neural network architecture which aims at building high-level feature representation from low-level input automatically with minimal domain knowledge of the problem, are followed to represent and to recognise different statuses of eye, mouth and ear on image patches. These statues can be therefore used to indicate driver physiological signals such as PERCLOS, eye closure duration(ECD), blink frequency and yawing. Our work focus on the characterization of statuses of eye, mouth and ear on image patches, with high-level features extracted

hierarchically from raw input image. Each convolutional layer generates feature maps using sliding filters on a local receptive field in the maps of the preceding layer (input or max-pooling layer). The map sizes decrease layer by layer such that the extracted feature becomes more complex and global. Then, the output is input to a fully connected multilayer perceptron (MLP) classifier. The proposed approach was evaluated on a self-designed real driving dataset, demonstrating promising performance.

The key contributions of this work can be summarized as follows:

1. To recognise statuses of eye, mouth and ear, this chapter proposed to build a deep convolutional neural network in which trainable filters and local neighborhood pooling operations are applied alternatingly for automatically exploring salient features. Using CNN to learn rich features from the training set is more generic and requires minimal domain knowledge of the problem compared to hand crafted feature.

2. We proposed to applied Face++ Research Toolkit [27] to localize the facial landmark [28] on the driver's face, which is used to propose the region of eye, mouth and ear. It is much robust under the effect of illumination variation and occlusion in real driving condition than previous approaches.

3. The proposed approach was evaluated on on a self-designed driving dataset, with best performance achieved with an overall accuracy of 95.56%.

The rest of the chapter is organized as follows. Section 7.2 presents an overview of our proposed method and the driving dataset creation, while Section 7.2.2 gives a detailed introduction to the convolutional neural network. Section 7.4 reports the conducted evaluation and the experiment results, followed by some conclusions presented in Section 7.5.

6.2 System Overview

The proposed driving inttention recognition system comprises three steps: (i)driver face landmark localization and region segmentation (ii)train the network with six class of labeled data, (iii) use the network to extract feature from input for classification . A schematic illustrating the operation of the proposed driving inattention recognition system is shown in Fig.6.1.

6.2.1 Driving Dataset Creation

To evaluate the proposed driving inattention recognition approach, a driving dataset was recorded using a Philips CVR300 car driving recorder. It was mounted in front window of a

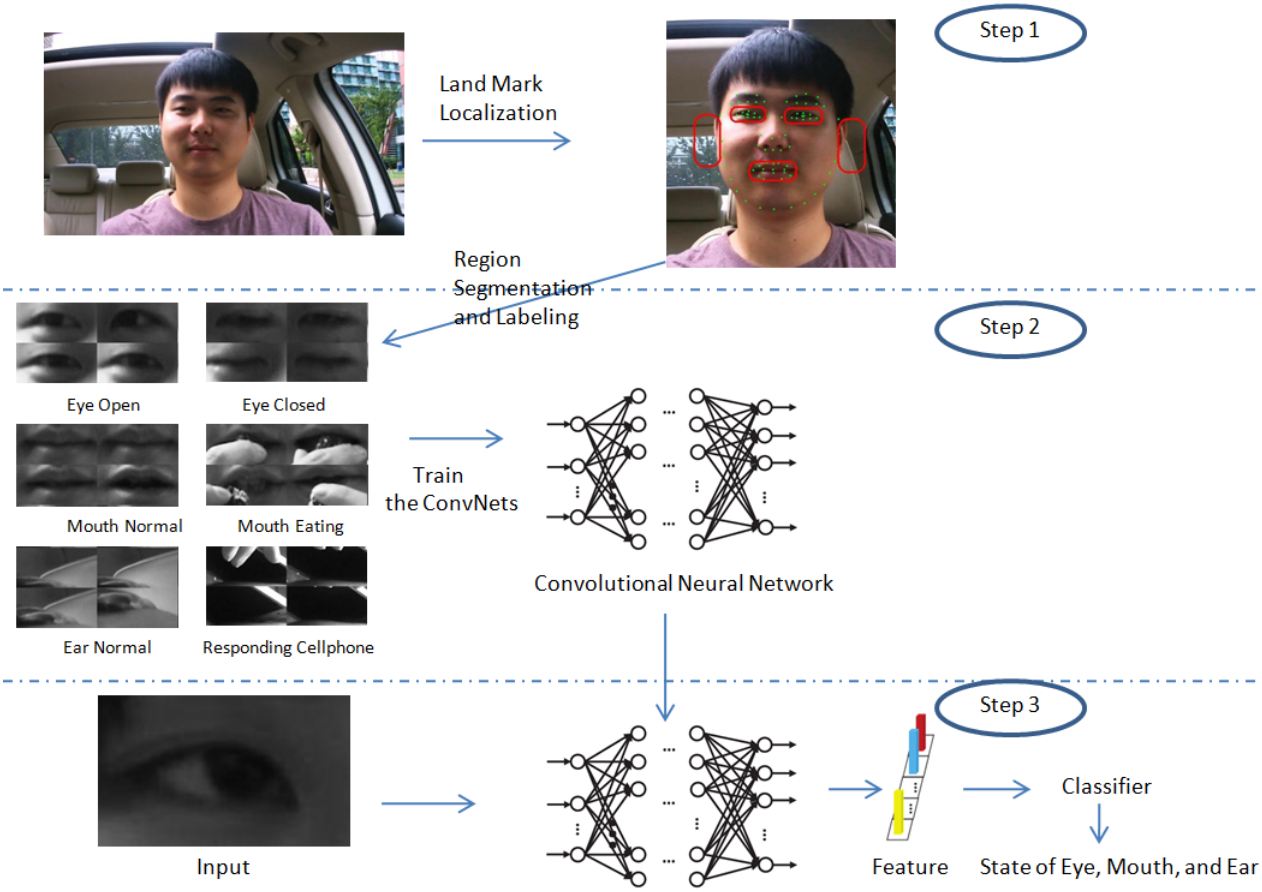


Fig. 6.1 The frameworks of our method.

Fig. 6.2 Example images of from the driving dataset. Up left: Normal driving. Up right: Sleepy. Bottom left: Eating. Bottom right: Responding a cell phone.

family car. In the dataset creation, six participants pretend to drive the car. The drivers were required to conduct three activities while driving, that is, falling asleep, eating cookies and responding cellphone calls. Some frames extracted from the video are shown in Fig.6.2, they are divided into four conditions including:

1. normal driving (up left)
2. falling asleep (up right)
3. eating (bottom left)
4. responding a cell phone (bottom right)

The driving activities can be recognise through analysing the state of eye, mouth and ear. In order to robust propose the region of of eye, mouth and ear, Face++ Research Toolkit [27] are applied to localize the facial landmark [28] on the driver's face. Some examples are shown in Fig.6.3. Then, image patches including eye, mouth and ear are separately segmented from the driver's face based on landmark points. The eye patches are manually labeled into two classes, that is, eye open and eye close. Some examples are show in Fig.6.4. The mouth patches are manually labeled into two classes, that is, mouth normal and mouth eating. Some examples are show in Fig.6.5. The ear patches are manually labeled into two classes, that is, ear normal and responding cellphone. Some examples are show in Fig.6.6.

6.3 Deep Convolutional Neural Network Architecture

The overall convolutional net architecture is shown in Fig.7.5. The network consists of three convolution stages followed by three fully connected layers. Each convolution stage includes convolutional layer, non-linear activation layer, local response normalization layers and max pooling layer. The non-linear activation layer and local response normalization layers were not illustrated in Fig.7.5 as data size was not changed in these two layers. Using shorthand notation, the full architecture is C(12,5,1)-$\tilde{A}$-N-P-C(16,5,1)-$\tilde{A}$-N-P-C(20,4,1)-$\tilde{A}$-N-P-FC(512)-$\tilde{A}$-FC(128)-$\tilde{A}$-FC(4)-$\tilde{A}$, where C(d,f,s) indicates a convolutional layer with d filters of spatial size $f \times f$, applied to the input with stride s. $\tilde{A}$ is the non-linear activation function, which uses ReLU [211] activation function. $FC(n)$ is a fully connected layer with n output nodes. All pooling layers P use max-pooling in non-overlapping 2×2 regions and all normalization layers N are defined as described in Krizhevsky et al. [200] and use the same parameters: $k = 2$, $n = 5$, $\alpha = 10^{-4}$, $\beta = 0.5$. The final layer is connected to a softmax layer with dense connections. The structure of the networks and the hyper-parameters were empirically initialised based on previous works using ConvNets, then

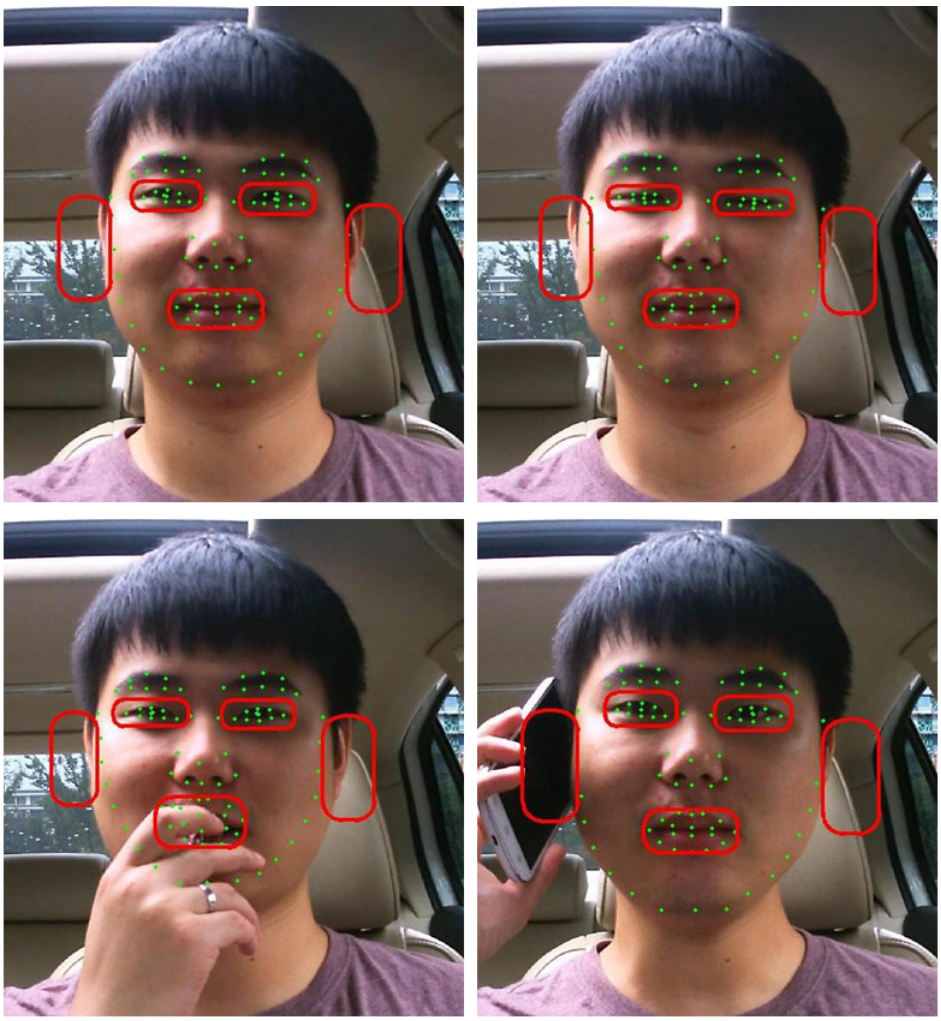
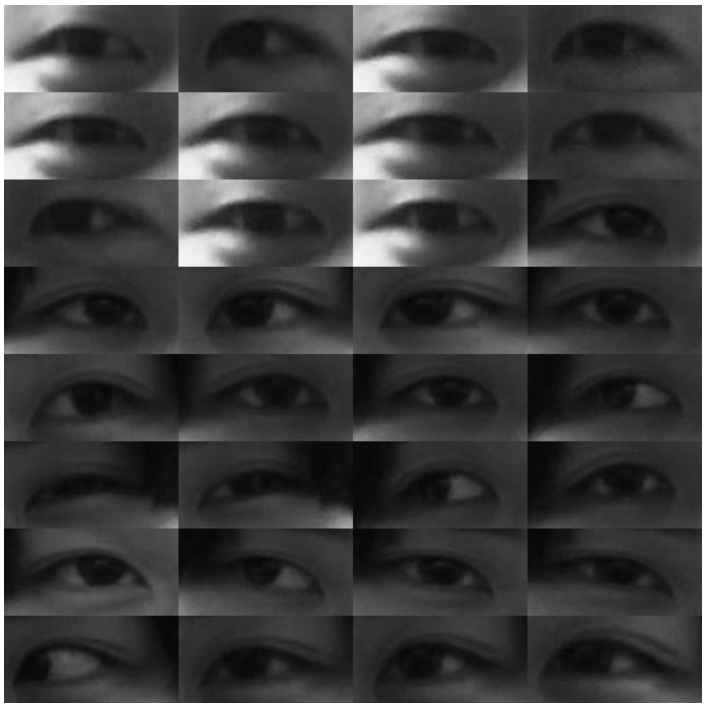
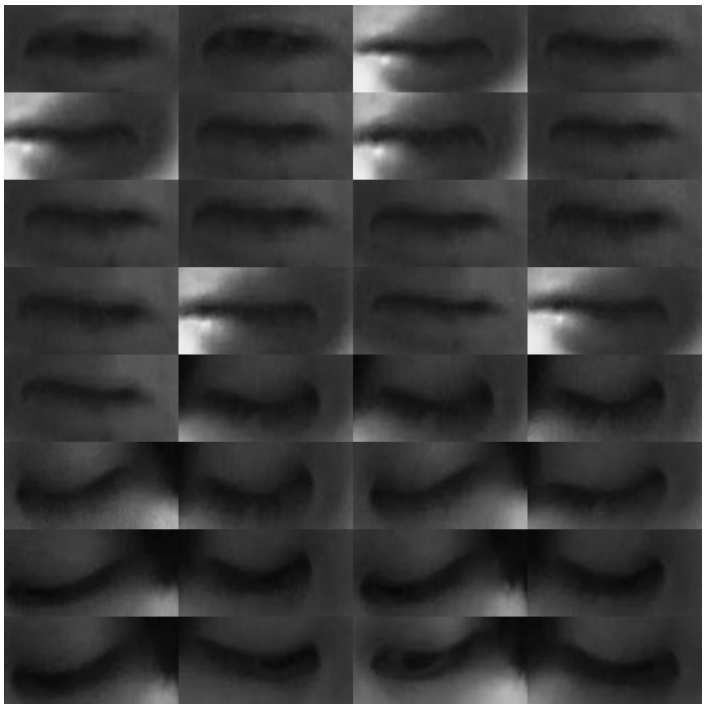


Fig. 6.3 Example images with landmarks. Up left: Normal driving. Up right: Sleepy. Bottom left: Eating. Bottom right: Responding a cell phone.

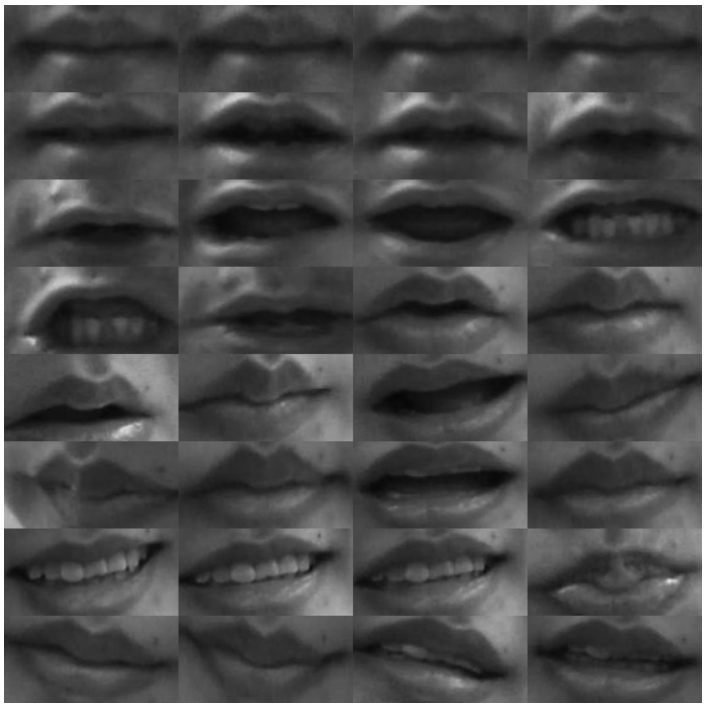


(a) Eye Open

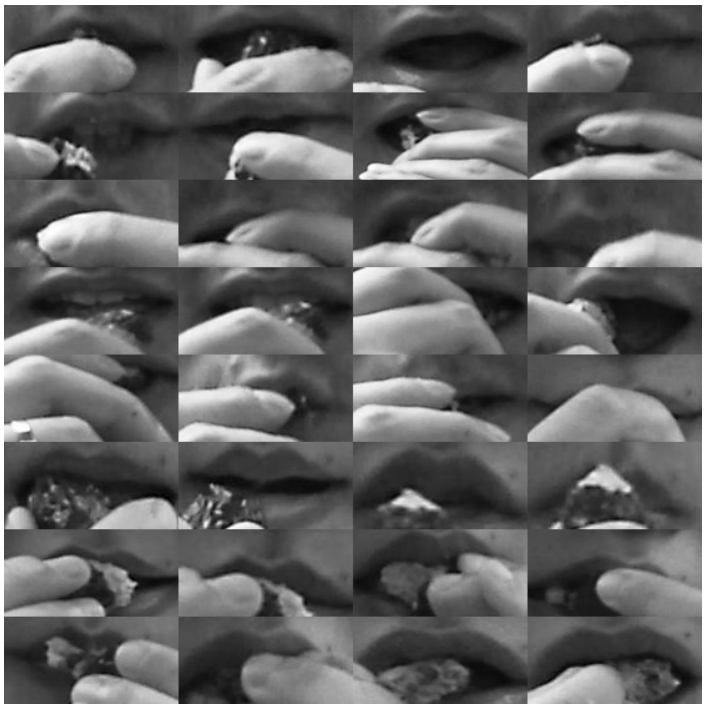


(b) Eye Close

Fig. 6.4 Samples of Eye Region

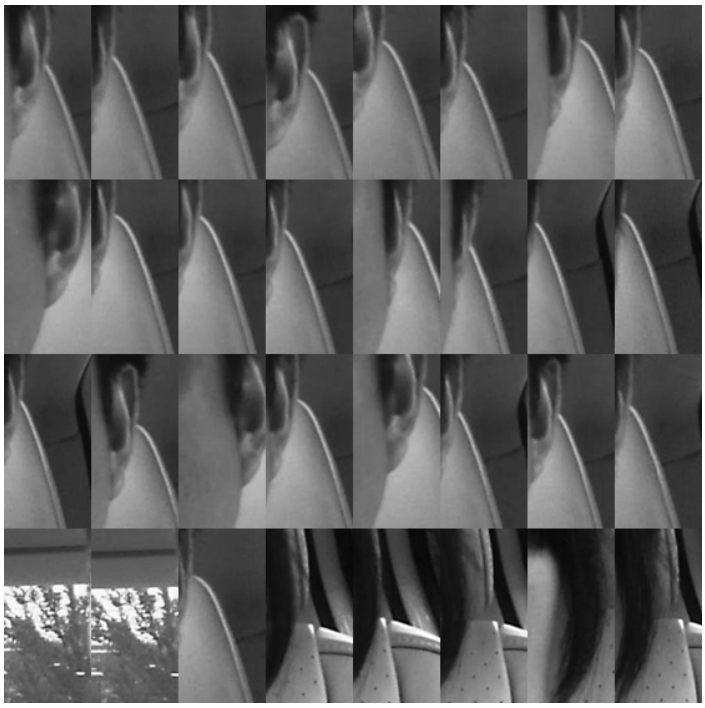


(a) Mouth Normal

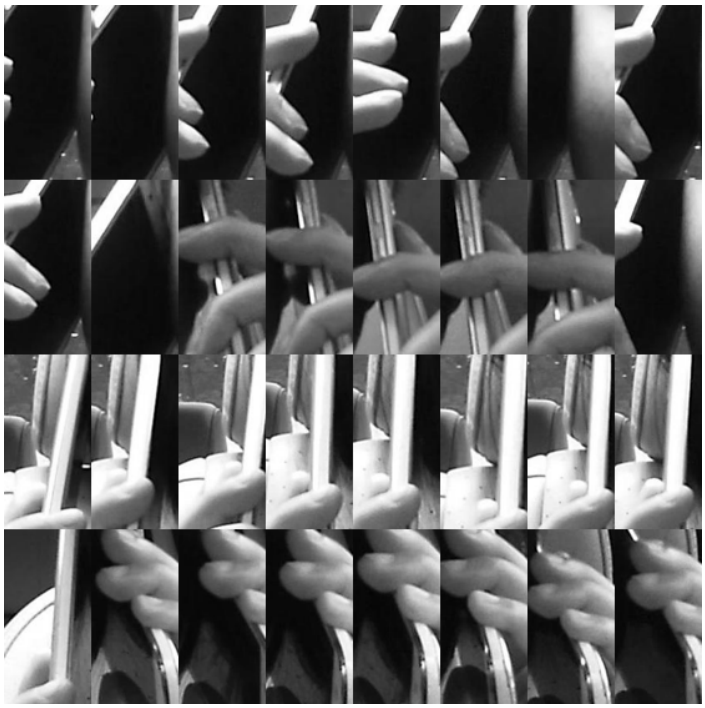


(b) Mouth Eating

Fig. 6.5 Samples of Mouth Region



(a) Ear Normal



(b) Responding Cellphone

Fig. 6.6 Samples of Ear Region

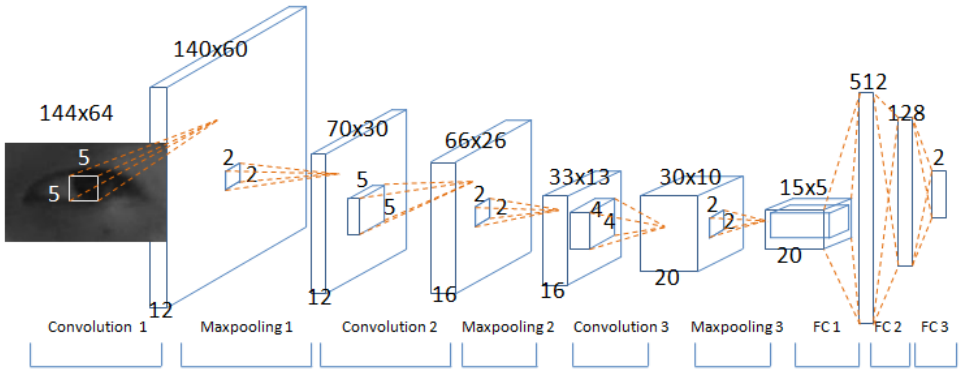


Fig. 6.7 The architecture of our unsupervised convolutional neural network. The network contains three stages, each of which is consisted of convolution layer, non-linear activation layer, local response normalization layer, and max-pooling layer. Only convolution and max-pooling layers which change the data size during operating, are illustrated here.

we setup cross-validation experiment to optimize the selection of network architecture in section7.2.2.

6.4 Experiment

In this section, we first conduct three evaluation experiments to select the CNN structure, activation function, pooling method and other hyperparameters in section7.2.2. Then the CNN model is applied to verify the effectiveness of the proposed algorithm on the driving database in section6.4.2.

6.4.1 Implementation Detail

We train our models using stochastic gradient decent with a batch size of 128 examples, momentum of 0.6, and weight decay of 0.0005. The learning rate is initialized as 0.01 for all trainable layers and adapted during training. How to adaptively control the learning rate in a reasonable value is an important issue in CNN learning. A too small learning rate makes the convergence rather slow, while a too big learning rate would make the network parameters vibrated. We proposed a adaptive learning rate by monitoring the loss function value and the validation error. To further prevent possible overfitting, we apply dropout and data augmentation as performed in [200].

The experiments were implemented on our GPU CNN package in C++ language based on NVIDIA CUDA and cuDNNv2. Our experiments are conducted on a NVIDIA GTX Titan GPU and a 4-core Intel(R) Core i7-3770 3.40-GHz computer.

6.4.2 Experiment

The experiment is conducted using five-fold cross-validation as told previously. The cross-validation process will then repeated five times. The training data are tested as a whole each five epoches and we stop to test training error after 210 epoches. The network is convergence after 250 epoches.

Table 6.1 Confusion matrix for the cross validation result

class	EyeOpen	EyeClosed	MouthNormal	MouthEating	EarNormal	RespondingCellPhone
EyeOpen	99.47	0.53	0	0	0	0
EyeClosed	0	100	0	0	0	0
MouthNormal	1.78	0	98.22	0	0	0
MouthEating	0	0	0.45	99.55	0	0
EarNormal	0	0	0	0	95.56	4.44
RespondingCellPhone	0	0	0	0	2.51	97.49

After we repeated five cross-validation experiments, to further evaluate the classification performance, 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. The confusion matrices of the our results are shown in Table6.1. The accuracy for six states of eye, mouth and ear are 99.47%, 100%, 98.22%, 99.55%, 95.56% and 97.49% respectively.

6.5 Conclusion

This chapter addresses the importance of automatic understanding and recognition of driver inattention in the scenario of reducing motor vehicle accidents, and presents a novel system for vision-based driving inattention recognition. We verify our approach on a driving dataset. The proposed approach applied deep convolutional neural network, which learns feature from

raw image automatically. The final results demonstrate promising performance, achieving an overall accuracy of 95.56%.

Chapter 7

Multi-attributes Pedestrian Gait Identification by Convolutional Neural Network

Gait is a biometric feature that can be measured remotely without physical contact and proximal sensing. The research of gait recognition is strongly motivated by the demands of security that require automatically identifying person at a distance. This chapter proposes a robust and effective gait recognition approach using convolutional neural networks(CNNs) and multi-task learning model(MTL). Firstly, we extract Gait Energy Image(GEI) from each walking period as the low level input for the CNNs. We train our multi-task CNN model through back-propagation using a joint loss of each task. Then, the high-level features for multiple tasks could be extracted simultaneously with the given input. Generally, there are two major advantages of our method:(i) Semantic features are learned via CNNs, which requires minimal domain knowledge of the problem, (ii) multi-attributes learning for CNN model improves the gait identification accuracy and increase the convergence speed for training. The performance of our design is verified on CASIA gait database B, achieved over 95.88% high accuracy for each task. The proposed method is also compared with the state-of-art approaches using CASIA gait database A and OU-ISIR Treadmill dataset A, demonstrated competitive performance. To the authors' best knowledge, this is the first time that multi-attributes based gait identification is proposed.

7.1 Introduction

In a real world application, a satisfactory gait identification is a highly challenging task due to: (i) exterior factors such as clothing, shoes, carrying objects, and environmental context; (ii)the subject's physical and mental factors, e.g., walking speed, leg injury, drunkenness, illness, fatigue, pregnancy, etc; (iii) observation factors including occlusions in the scene, variations in viewpoint, shape distortions due to carrying conditions, shadows under feet and uneven ground.

In the past few years, many researches [100, 121, 123, 7, 110, 112, 124] on human gait recognition have been conducted. Generally, these approaches are highly problem dependent. Specifically, most methods use hand-crafted features from raw data with certain assumptions about the circumstances under which the data was taken. It cannot reach an optimal balance between robustness and discriminative. Especially for human gait recognition, the appearance may be dramatically different for a same person in terms of their exterior factors, subject's condition factors and observation factors. Thus, gait identification is still a challenging task in the real-world application.

In recent years, with respect to feature matters, there has been a growing interest in deep learning models [194–196], such. They are a class of machines that can build multiple layers of feature hierarchies and automatically learn high-level features from low-level ones. The feature representation hence become more generic since the construction process is fully automated. Such learning machines can be trained using either supervised or unsupervised method, and the they have been shown competitive performance in speech recognition [197, 198, 236], brain electroenceephalogram (EEG) signal recognition[237], natural language sentences recognition [199], visual classification task[200–203], visual detection task[238, 204], and other visual task[239, 205–209]. One of the most used deep learning models is, the Convolutional Neural Network architecture(CNN) [196, 200], a bio-inspired hierarchical multilayered neural network able to learn visual patterns directly from the image pixels without any pre-processing step. It provides a level of invariance to shift, scale and rotation as the local receptive field allows the neuron or processing unit access to elementary features such as oriented edges or corners. Hence, in this chapter, we use CNN to identify human gait via learning rich features from the data, which hasn't been applied in human gait identification yet to the best knowledge of author.

Further more, recent works [240–242] on visual recognition that applying convolutional neural network(CNN) with multi-task learning demonstrated improved performance than standalone task. Multi-task model is a machine learning approach that jointly trains one task together with other related tasks at the same time sharing the same lower feature layers, which uses the commonality among tasks and therefore learns shared feature representation

benefits all tasks. Since the difficulties in human gait identification are mainly caused by the multi factor's effects, it is very natural to use multi-tasks learning to simultaneously identify the multiple attributes of the gait. Thus, this chapter aims to investigates a convolutional neural network model for identifying the human gait while simultaneously predicting other human attributes at same time.

As above, in the context of (i)to extract semantic features for gait representation with minimal domain knowledge of the problem, (ii)to produce more attributes information along with the person identification, this chapter propose a human gait identification method, where convolutional neural network model and multi-task learning model(MLT)are used. The proposed design is evaluated on the CASIA gait database A (CASIA-A), CASIA gait database B (CASIA-B) and OU-ISIR Treadmill dataset A, achieved competitive performance on accuracy. The key contributions of this work can be summarized as follows:

1. To identify human gait, this chapter aims to build a CNN model to automatically extract semantic features. The proposed model training requires minimal domain knowledge of the problem compared to manually designed feature in previous approaches.

2. To the best of our knowledge, this is the first approach that using MTL to investigate how human gait can be identified together with other auxiliary tasks. The approach is verified on CASIA gait database B, CASIA gait database A and OU-ISIR Treadmill dataset, demonstrating promising performance. Generally, our proposed method outperformed the state-of-art methods on the gait recognition while the attributes information are acquired along with the person identification.

The rest of the chapter is organized as follows. Section 7.2 presents an overview of our proposed multi-attributes gait identification, including the subsection 7.2.1 for introducing the data preprocessing and subsection 7.2.2 for describing the architecture of the CNN model. Section 7.3 introduces training details, including multi-task learning, regularization rules and implementation details. Section 7.4 reports the conducted evaluation and the experiment results, this is followed by some conclusions presented in Section 7.5.

7.2 System Overview

The general workflow of our design can be summarised into following steps: 1)low level gait sequences description, 2) high level feature representation and attributes identification:

Step 1 Pre-processing Extracting the low level feature representation of each piece of human gait silhouette sequence, where the Gait Energy Image (GEI) [100] is used GEI is a compact representation method for preserving spatial-temporal information of human gait, which has been demonstrated robust performance as a low level feature and widely used in many recent human identification approaches [100, 243–246, 121, 123], the detailed will be introduced in the section 7.2.1.

Step 2 High level feature representation CNN models will be used to extract high-level features from the low level GEI input. The CNN model is constructed with several layers. Each layer generates feature maps using sliding filters (kernals) on a local receptive field in the maps of the preceding layer. Thus the feature map will be processed and feeded forward layer by layer. At the end stage, the multi-layer perceptrons (MLPs) are applied classifier are assigned separately to each task , where the features that feeded from earlier stage will be used to identify the gait attributes. The detail of our CNN model architecture will be introduced in 7.2.2.

Step 3 gait attributes identification With a given input gait, it will be firstly translated to the low level features, then it will feed forward the trained CNN model to extract the high level features. Finally the gait attributes will be identified .

A overview of proposed multi-attributes gait identification system is shown in Fig.7.1

7.2.1 Pre-processing: Low Level Feature Extraction

Previously, majority approaches in gait recognition derive representation from human silhouette[119–124], aiming appearance invariant to identify human gait. The well-known Gait Energy Image(GEI) have been demonstrated powerful performance[98, 113, 100, 114, 115, 117, 118, 121–123] in representing human gaits. For a given gait silhouette sequence, assume $B_t(x,y)$ stands for the binary gait silhouette image at t frame, the GEI is defined as

$$G(x,y) = \frac{1}{N}\sum_{t=1}^{N} B_t(x,y) \tag{7.1}$$

where N is the total number of frames in the sequence, and x and y are values in the 2D image coordinate. GEI reflects gait rhythm by holding several key information of human gait including motion frequency, temporal and spatial changes of human body, and global body shape statistic. Hence, we use GEI as low level input feature map to our CNN model. Besides, the completed walking cycle(s) were determined as the periodic motion essence of

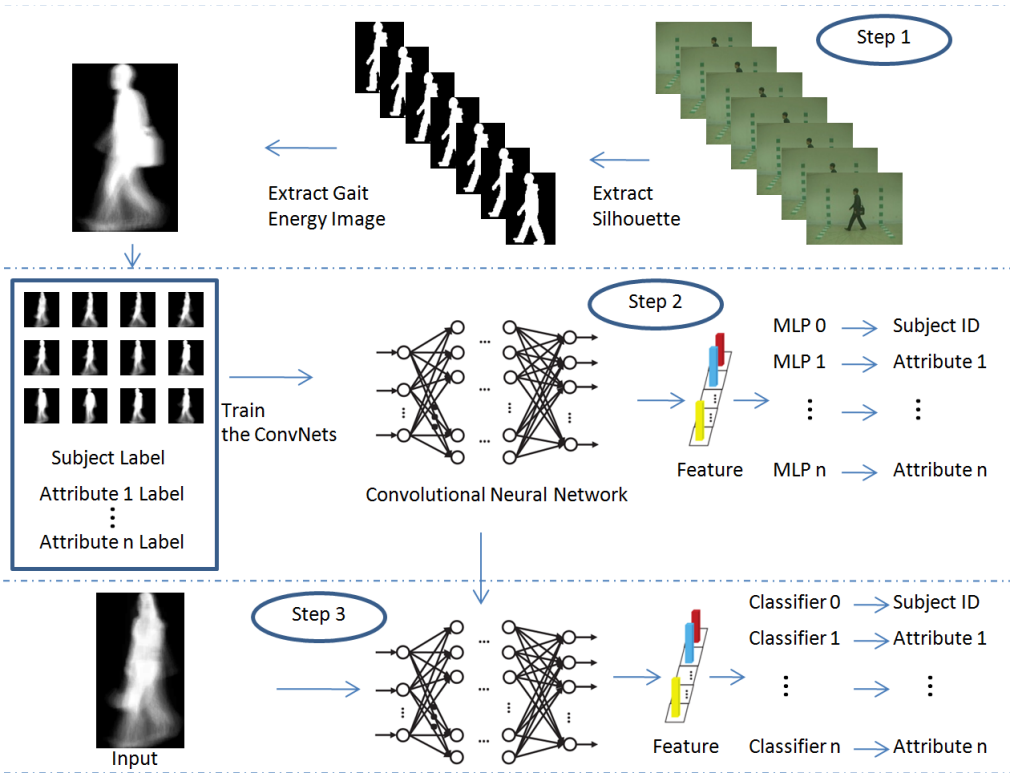


Fig. 7.1 The frameworks of our method.

gait, where the normalization and autocorrelation method[118] are applied to analysis the gait period. Fig.7.2 shows some examples of GEIs from various subjects under various views on the CASIA-B.

7.2.2 Multi-Task Convolutional Neural Network: High Level Feature Extraction

In this section , we will firstly introduce the overall view of our CNN model and then describe how we select the architecture.

Overall of the network

The CNN models are constructed with several layers, where each layer has maps and parameters. As shown in Fig 7.3,the maps of each layer would be calculated according to this layer's parameter and the maps of the previous layer, since its input maps are the output maps of previous layer.

Generally, a CNN model includes following important components:

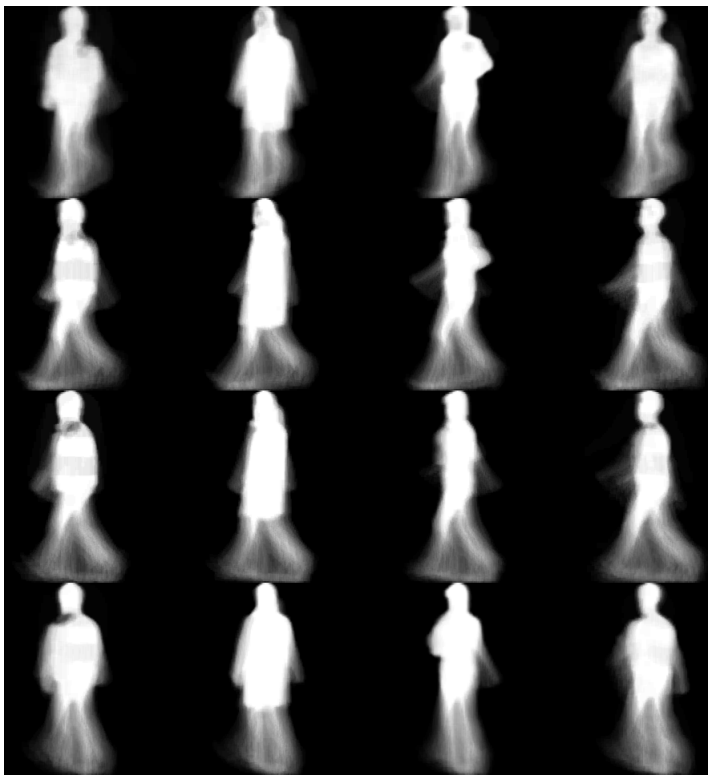


Fig. 7.2 Examples of gait energy images (GEIs).

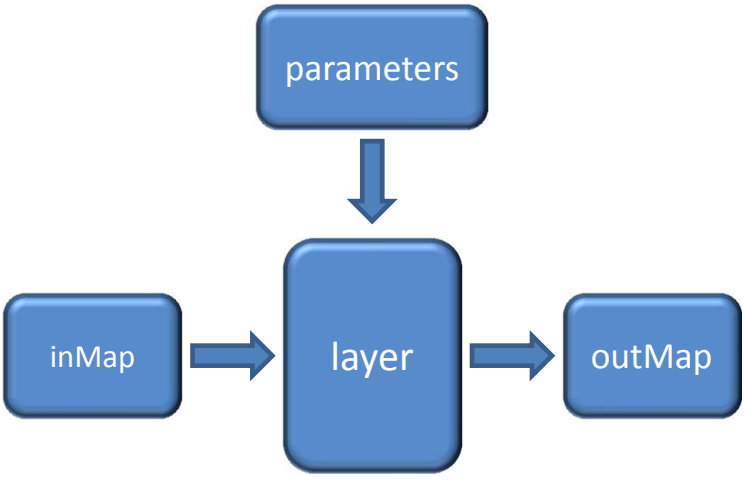


Fig. 7.3 Structure of a layer

Convolutional layer The purpose of convolutional layer is to extract the local patterns. It is parameterized by following factors: the number of maps, the kernel sizes and stride. In this approach, we denotes $C(d, f, s)$ to represent a convolutional layer, where d is the number of feature maps, f is the kernel size and s is the stride. For instance, $C(10, 5, 1)$ means the convolutional layer have 10 maps with vernal size 5 and the stride is 1. The size of output are determined by the size of input, the kernel size and the stride.

Non-linear activation function The convolutional layer works as a linear filter. In order to form a nonlinear complex model. The nonlinear activation function would be added with the convolutional layer. The common activation function includes *tanh*, *sigmod*, *relu*, *softplus*, etc. Here is an example of activation functions shown in Fig 7.4.

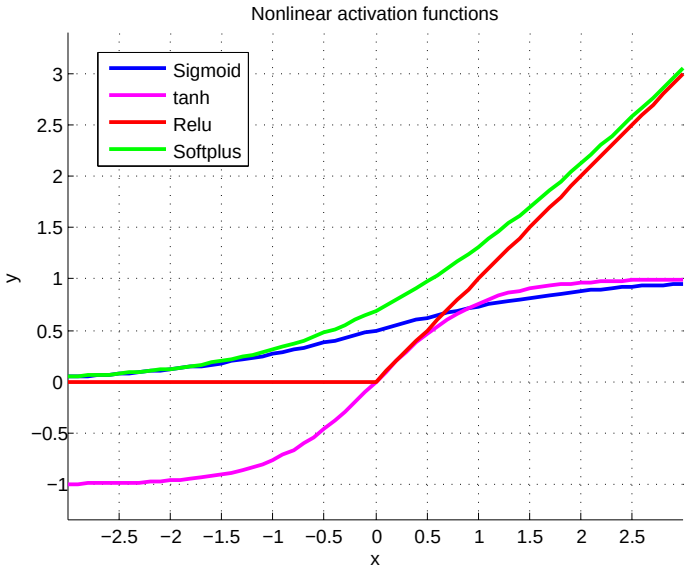


Fig. 7.4 An example of non-linear activation function

Max-pooling layer The purpose of Pooling layer is to subsample the feature maps and reduce the resolution. The reason is that those features extracted from convolutional layer have precise positions. It maybe harmful for following procedure, e.g. classifying since different training instances with the same label have different precise positions. Inspired by [200, 209], the max-pooling method is used in this approach

Normalization The normalization operation is able to give the trained model a better generalization. Inspired by [200], the local response normalization (LRN) is applied for

generalization purpose. The equation of LRN is shown in eq 7.2

$$y^i_{m,n} = \frac{x^i_{m,n}}{[k + \alpha \sum_{j=\max(0,i-l/2)}^{\min(N-1,i+l/2)} (x^j_{m,n})^2]^\beta} \tag{7.2}$$

where the summation operates over a l "adjacent" kernel maps at the same spatial position m,n, and N is the total number of kernels in the layer. In this approach, we followed [200] use the empirical parameters: $k = 2$, $n = 5$, $\alpha = 10^{-4}$, $\beta = 0.5$.

Fullly connected layer The previous operations in convolution layer, nonlinear activation, normalization and pooling layer could be defined as a large convolutional stage. The input has been converted to lots of low-resolution feature maps. In full link layer, these small size feature maps are concatenated in to a long vector, where such a vector plays a same role as those manually designed features. Then, the long vector is fed to a one-hidden-layer neural network, which works similar as linear regression. After that, an nonlinear activation function is applied to finally form the output feature map.

Output layer - Multiple layer perception (MLP) The final stage of the model is MLPs. Each MLP is corresponding to an attribute identification task, e.g. Subject, Probe Scene (the wearing conditions), Probe view, etc. A softmax function will be added to each MLP to finally predict the attributes.

The structure of the networks and the hyper-parameters were empirically initialised based on the training data and previous works using CNNs, then we setup cross-validation experiment to optimize the selection of network architecture in section 7.2.2.

Architecture Selection

The overall CNN architecture is shown in Fig.7.5. The network consists of three convolution stages followed by one fully connected layers. Each convolution stage includes convolutional layer, non-linear activation function, local normalization and pooling layer. The lower layers are finally shared by N split layers, each of which is fulled connected with the topmost shared layers. Each split layer is respected to one attribute identification task and contains several additional fully connected layers with size 128. The size of split layer is k, which equals to the class number in respected task.

In order to determine the parameters, we use the training data to do cross validation experiment. We firstly set default values and choose the best one according to the cross validation (CV) performance, which is shown in Fig 7.6.

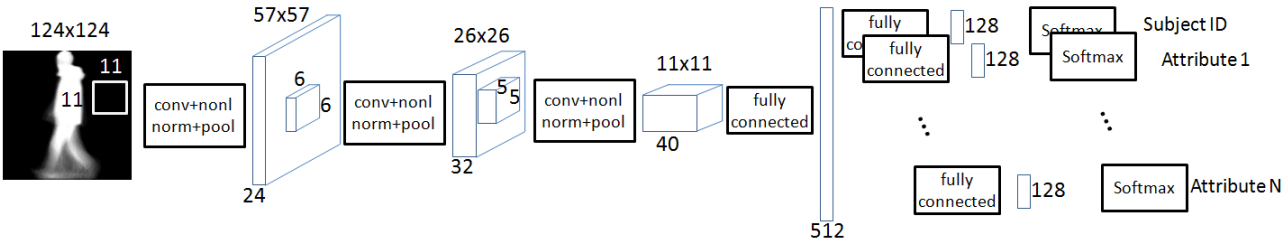
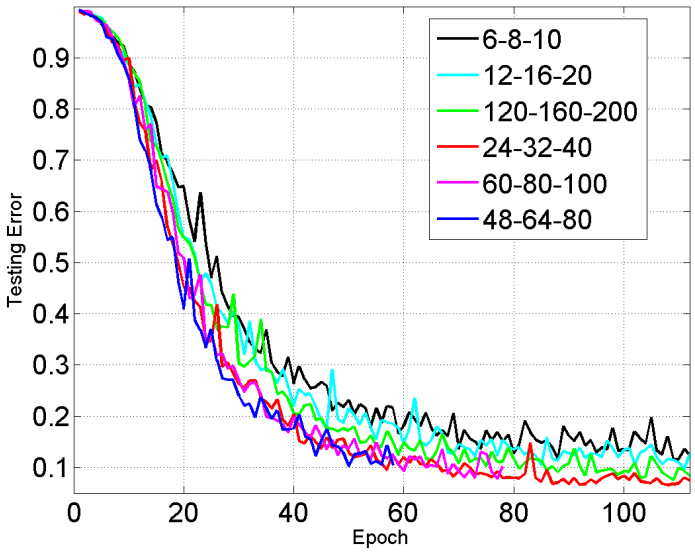


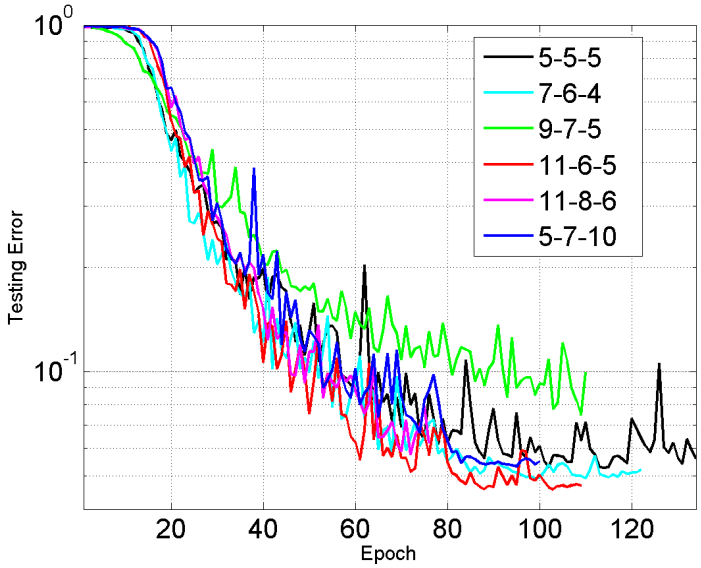
Fig. 7.5 The architecture of our convolutional neural network. *(conv)* stands for convolutional layer, *(nonl)* stands for nonlinear activation function, *(norm)* stands for normalization and *(pool)* stands for the max-pooling layer. The network contains three stages, each of which is consisted of convolution layer, non-linear activation layer, local response normalization layer, and max-pooling layer. Only convolution and max-pooling layers which change the data size during operating, are illustrated here.

Table 7.1 Architecture of our CNN

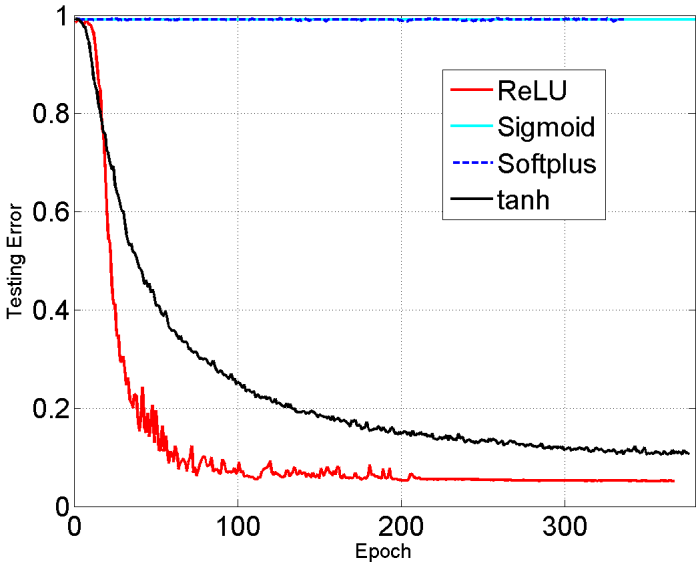
Layer	Type	Feature maps and Size			Kernel
1	Input	1 map with 124×124 neurons			
2	Convolution $C_1(24,11,1)$	24 maps with 114×114 neurons			11×11
3	Max pooling	24 maps with 57×57 neurons			2×2
4	Convolution $C_2(32,6,1)$	32 maps with 52×52 neurons			6×6
5	Max pooling	32 maps with 26×26 neurons			2×2
6	Convolution $C_3(40,5,1)$	40 maps with 22×times22 neurons			5×5
7	Max pooling	40 maps with 11×times11 neurons			2×2
8	Fully connection	512 neurons			
		Task 1	Task 2	Task 3	
9	Fully connection	128 neurons	128 neurons	128 neurons	
10	Softmax output	124 neurons	11 neurons	3 neurons	



(a) Cross validation to evaluate six group of map numbers



(b) Cross validation for six group of filter size



(c) Cross validation error of four activation functions

Fig. 7.6 Architecture selection

Max-pooling layer The parameter of max-pooling layer is chosen empirically based on the input GEI image size and the conditional layer at same convolutional stage.

Kernel numbers in convolutional layer With respect to the width of the network. The pooling layer which follows the convolution layer, will decrease resolution of the feature map. To prevent the information from being lost too quickly, the filter size will generally increased. Here, we setup experiment to compare with six groups of filter number. The validation testing error with epoch are shown in Fig.7.6a. From the figure, the red line which stands the filter number group of 24-32-40, has the least testing error. Hence, we choose this group as our filter numbers in each convolution layer.

Kernel size in in convolutional layer According to learning theory, if the architecture has too much capacity, it tends to overfit the training data and has poor generalization. We first evaluate the filter size by comparing six empirically selected group of kernel size. The average result of cross validation among the six group of filter size is plotted in Fig.7.6b. We select the filter size of 11-6-5 for the network as its lowest CV error.

Activation Function Selection. After the convolutional layer parameters are determined, the ReLU has been reported faster convergence than others in next step CV test, where the result is shown in Fig.7.6c. From the figure, the network is not convergence using softplus or sigmoid activation function in a learning rate of 0.01 within the observed epoches due to the diminishing gradient flow [214]. While the ReLU makes the convergence faster and yield a better performance as reported in most recent CNN approaches.

After the cross validation experiment, the parameters chosen in our final architecture are listed in Table 7.1.

7.2.3 Pre-train

By plain supervised gradient descent with sufficient annotated training data, learning Convolutional Neural Networks (CNN) that contains millions of parameters , have been demonstrated competitive performance for visual recognition tasks [200–203, 208, 209] when starting from a random initialization. However, CNN architecture has a property that it strongly depends on large amounts of training data for good generalization. When the amount of labeled training data is limited, directly training a high capacitor CNN from only a few thousand training images is problematic. For example, applying CNN on tasks such as video classification and human gait classification, the performance may suffers from limited datasets. However, researches [227, 247] have shown an alternative solution to compensate the property of

CNN, that is, choosing a optimised stating point which can be pre-trained by transferring parameters either supervised or unsupervised, as opposite to random initialized start.

The scheme of pre-training a deep convolutional neural network by transferring parameters may not a tricked solution but a nature essence, which can be considering inspired by a curious phenomenon. The first layer of many deep neural networks trained on natural images learns features similar to Gabor filters and color blobs. This phenomenon occurs in different datasets and training objectives, including supervised image classification [200], unsupervised density learning [248], and unsupervised learning of sparse representations [249]. It seems that the first-layer features appear not to be a particular dataset or task, but general in that they are applicable to common visual datasets and tasks. Therefore, an intuitive hypothesis is given[228] that features must eventually transition from general to specific layers by layers from bottom to top in a conventional deep neural network, which may provides the theoretical support for the pre-training by transferring parameters.

With respect to the study on pre-train. Erhan et al. [227] empirically explored the influence of pre-training, (i) confirming the advantage of unsupervised pre-training which guides the learning towards basins of attraction of minima that support better generalization from the training data set, (ii) supporting the regularization explanation for the effect of pre-training. Yoshinski et al.[228] proposed two negatively affection issues in transferability among different tasks and also documented that pre-train is always better for the target task even from distant base tasks compared to random initialization. Oquab et al.[250] transferred the parameters from ImageNet classification[200] and fine-tuned on PASCAL VOC 2007 and 2012, demonstrating significantly improved by transferred representation. Alternatively, as lack of near base task and small volume of available target tasks, Simonyan et al.[202] applied multi-task learning for two target tasks. demonstrating (i) an effect method to increase the amount of training data, (ii) improved performance on both tasks and (iii) better performance than pre-training on one target task and fine-tuning on another target task.

As above, we summarise two reasons for pre-training our deep ConvNet as opposite to starting from random initialization, (i)to speed up the training procedure to reach a satisfying convergence point (ii) to compensate the small volume of current available gait dataset for better generalization and performance. We setup experiment to evaluate different options of pre-training. Here we regarded CASIA-B gait dataset as our target dataset,and CASIA-A,CASIA-C and OU-ISIR-Treadmill A as our base dataset. We evaluate four different options of pre-train as follow:

1. use CASIA-C and OU-ISIR-Treadmill to fine-tune the network one by one, which is pre-trained using CASIA-A firstly.

2. Assume there is no human overlapping among the base datasets, and simply add these base datasets together to form a super base dataset for pre-training.

3. Unsupervised pre-train the network using sparse filter[210] based on the base datasets, which is easy to implement and hyperparameter-free compared to other unsupervised training scheme.

4. pre-train the network using multi-task scheme, the shared representation is connected to five softmax layers, each of which is respected to one of the base database and equipped with its own loss function. The overall training loss is computed as the sum of the individual tasks' losses.

Table 7.2 Evaluation on pre-train options

Pre-train setting	Accuracy
Training on CASIA-B without pre-train	92.20
Pre-trained on all base datasets one by one	94.12
Pre-trained on the super base dataset	94.66
Unsupervised pre-trained using sparse filter	93.33
Multi-task pre-training	95.17

The results are reported in Table7.2. As expected, it is beneficial to pre-train a ConvNets and transfer the parameters to the target task. Among the four options of pre-train method, multi-task learning performs the best, as it allows the training procedure to exploit all available training data simultaneously, which is identical to the conclusion in [202].

7.3 Training

We developed our CNN model in C++ language based on NVIDIA CUDA and cuDNNv2. Our experiments are conducted on a NVIDIA GTX Titan GPU and a 4-core Intel(R) Core i7-3770 3.40-GHz computer.

Back-Propagation(BP). We use BP algorithm to train our CNN model. We use cross-entropy loss as loss function of softmax output layer, which is defined as:

$$L = -\sum_{j=1}^{k} [t^j \log(p^j) + (1-t^j) log(1-p^j)] \tag{7.3}$$

where, p^j is the probability output of jth class, k is the total number of class, and t^j is the corresponding one-hot training label.

Multi-task Learning. In our experiment, given a low-level visual feature space together with attribute- and identity-labeled image data, we learn a feature subspace for all labeling tasks based on a joint loss function that favors common sparsity. When multi-task learning is performed, we minimize the linear combination of each task loss as:

$$L_{joint} = \sum_{i=1}^{N} \alpha_i L_i \tag{7.4}$$

where N is the total number of task, α_i is the i-th task linear weights and L_i is the i-th task loss. The optimization process alternates between regularizing towards shared features, and retraining task-specific classifiers based on those features.

Regularization method. With respect to prevent overfitting, we apply dropout and data augmentation as performed in [200].

Implementation Detail. We train our models using stochastic gradient decent with a batch size of 128 examples, momentum of 0.6, and weight decay of 0.0005. The learning rate is initialized as 0.01 for all trainable layers and adapted during training. Control the learning rate in a reasonable value is important. A too small learning rate makes the convergence rather slow, while a too big learning rate makes the network not converge. Inspired by [200], we monitor the loss function value and the validation error, and decay the learning rate manually during training.

7.4 Experimental Result

The CASIA gait database B is used to evaluate the performance of the proposed method. The experiments are designed for two scenarios. The first scenario is to evaluate our model's performance and make comparison with the state-of-art methods. The second scenario is to analysis the characteristics of our multi-task CNN. The performance of the multi-task learning will be analyze and compared with single task learning. Moreover, the difference of model training using GPU and CPU are also shown.

7.4.1 Evaluation of Multi-task Gait Identification on CASIA-B Gait Database

In CASIA gait database B, it contains 124 subjects, while view angle and extra walking conditions are two variants for each action pieces. The view angle uniformly varies from

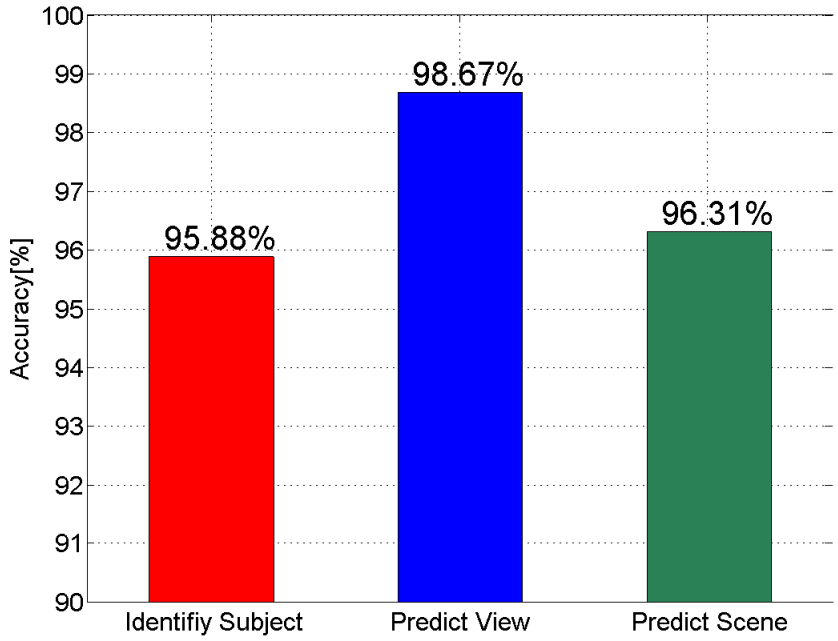
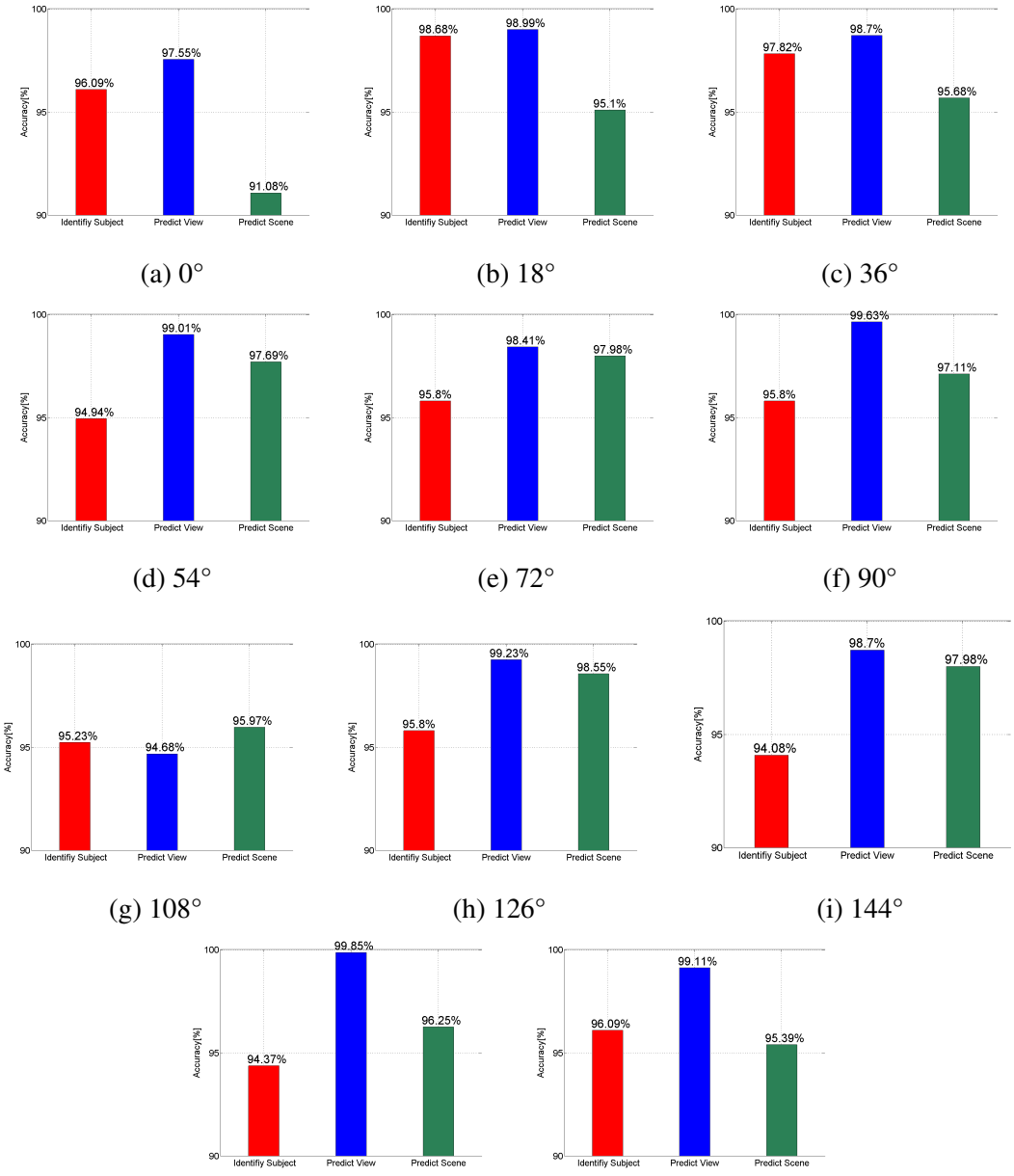


Fig. 7.7 The average performance of multi-task learning on CASIA-B for all three tasks.

0°to 180°which totally contains 11 classes. The walking conditions are divided into three categories: walking with a coat(cl), walking with a bag(bg), and normally walking without anything(nm). We therefore connect three multi-layer perceptron(MLP) to one shared low layers of convolutional neural networks, which are used to predict subject identity, view angle and walking scene respectively. The networks are trained through back-propagation by the joint loss of three tasks. We set an equivalent linear weight L_i to $\frac{1}{N}$ for each task as in Equ.7.4.

We choose a standard experimental procedures, namely, holdout approach to verify the experiment performance. In the holdout experiment, as each subject walked 10 times in the scene (6 nm + 2 cl + 2 bg), we randomly hold out one from six nm scene , one from two cl scene and one from two bg scene , that is , a total of $(1+1+1)\times 11\times 124 = 4092$ videos for testing. the remained are used as training data. The overall performance are shown in Fig.7.7. The average accuracy of identifying subject, predicting view and predicting scene are 95.88%, 98.67% and 96.31%, respectively.

The detailed performance of three tasks in different probe view are illustrated in Fig.7.8.It is observed that subject identification task has higher accuracy in the probe view of 180°, 0°, 18°and 36°than other views, which means gait is easier to identity in front view than lateral view. However, scene is easier to predict in lateral view than front view. Fig.7.9 shows the



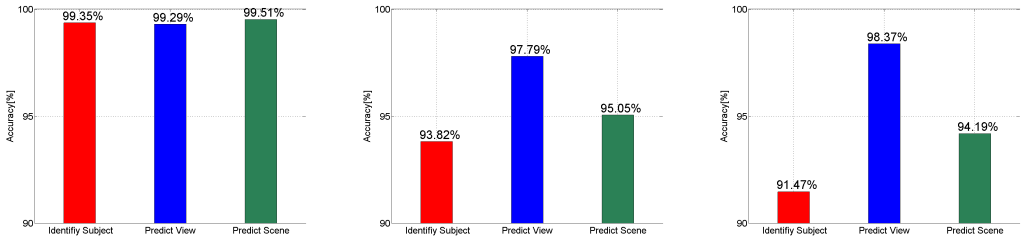
(a) 0° (b) 18° (c) 36°

(d) 54° (e) 72° (f) 90°

(g) 108° (h) 126° (i) 144°

(j) 162° (k) 180°

Fig. 7.8 The performance of multi-task learning on CASIA-B from all the 11 views. The probe viewing angles are (a) 0°, (b) 18°, (c) 36°, (d) 54°, (e) 72°, (f) 90°, (g) 108°, (h) 126°, (i) 144°, (j) 162°, and (k) 180°respectively.



(a) nm (b) cl (c) bg

Fig. 7.9 The performance of multi-task learning on CASIA-B from all the 3 scenes. The probe scenes are (a) nm, (b) cl, and (c) bg respectively.

performance of multi-task learning on CASIA-B from all the 3 scenes. The cl and bg scene attribute yield a negative effect to all tasks, especially for subject identification task.

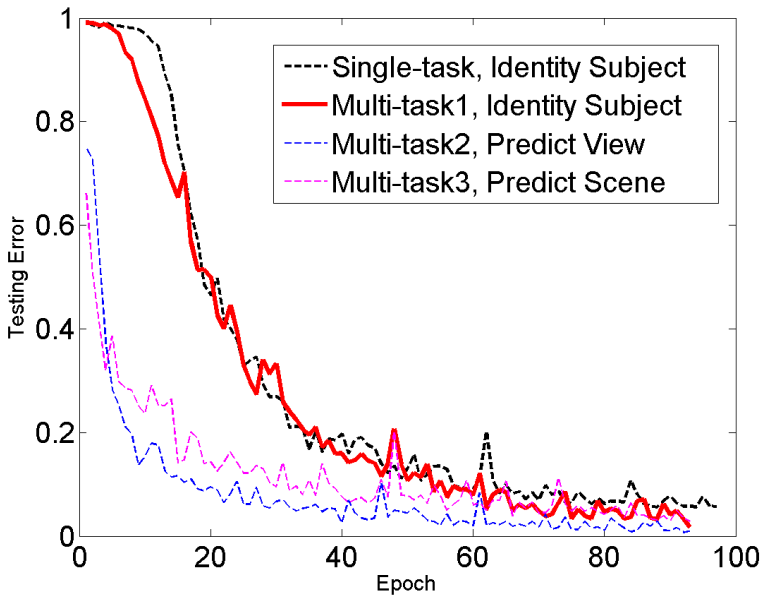


Fig. 7.10 Performance on CASIA-B: Multi-task v.s. Single-task

To further analysis the effect of multi-task learning, we compare our proposed model with a single gait identification task CNN, where the same distribution of training, validation and testing data are used. The result of training convergence is shown in Fig.7.10, the gait identification in single task is donated as black dash line, while the the gait identification(main task), view prediction and scene prediction in multi-task learning is donated as red line, pink dash line and blue dash line respectively. Comparing the red line with black dash line, the multi-task learning terminates at 92th epoch due to 0 training error while single-task learning terminates at 97th epoch also by 0 training error. Compared with the single task scheme, the result shows that multi-task learning is able to increase the convergence speed for model

training. Besides,the multi-task learning reduces the test error of gait identification task by 1.87% in the experiment. Hence, both the convergence speed and identification accuracy are benefitted from exploiting multi attribute description.

In addition, we have also evaluated the time cost for 1 epoch in the model training. For CPU training, it was implemented in parallel with 4-core 8-thread i7-3770 3.40-GHz CPU, and for GPU training, it was implemented with a NVIDIA GTX Titan GPU. The result of average training time for 1 epoch is shown in Fig 7.11

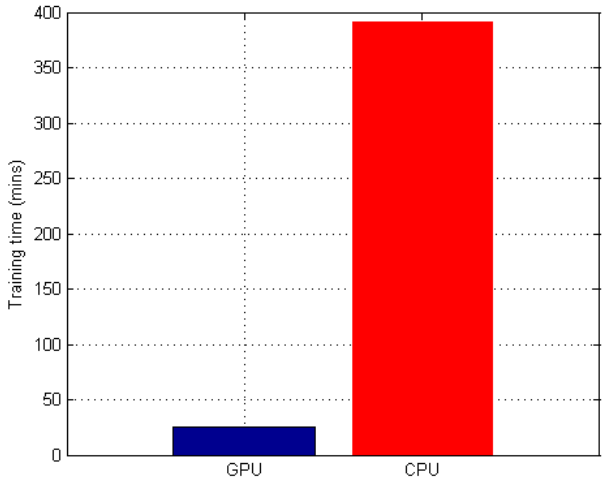


Fig. 7.11 Comparison of training time for 1 epoch on CASIA-B: GPU v.s. CPU

7.4.2 Comparison with State-of-art Methods

Finally, we compare our design with state-of-are methods using CASIA-B. In addition, we evaluate our multi-task gait identification approach using other two gait database including CASIA-A gait databse and OU-ISIR treadmill dataset A(speed variation).

Comparison of Performance using CASIA-B

The proposed method is further compared with existing methods, under changes of scene, that is walking with a coat(cl), walking with a bag(bg), and normally walking without anything(nm). The comparison among best reported results are shown in Table.7.3. The proposed method significantly outperforms other existing method under each probe scene (subject wearing conditions).

Table 7.3 Comparisons with other existing methods under changes of clothing and carrying condition

Gallery-Probe	nm	bg	cl	Avg
LF+AVG[251]	71.4	63.1	60.7	65.1
LF+DTW[251]	61.9	17.9	0.0	26.6
LF+oHMM[251]	63.8	31.8	21.4	39.0
LF+iHMM[251]	94.0	64.2	57.1	71.8
GEI+PCA+LDA[99]	90.5	3.6	3.6	32.6
GPPE[252]	93.4	62.2	55.1	70.2
GEnI[253]	92.3	65.3	55.1	70.9
STIP [6]	94.5	81.5	82.3	86.1
The proposed	99.4	97.8	94.2	97.1

Comparison of Performance using CASIA-A

The CASIA gait database A [7] includes 20 subjects, each of which contains three views, namely frontal (0°), oblique (45°) and lateral (90°) views. We setup a two-task experiment including subject identification and view prediction. Fig.7.12, illustrates the comparison result with other six methods using their best reported performance, including 3D deformation [1], 2D polar-plane [2], Neural network [3], PSC-PSA [4], Partial silhouette [5] and STIP [6]. Although, the performance of proposed method is limited by the amount of training data, it still demonstrated a competitive performance.

Comparison of Performance using Treadmill dataset A

The OU-ISIR gait database: Treadmill dataset A [254] contains six different walking speeds from 2 to 7 km/h with 1 km/h interval. A total of 34 subjects are used in this experiment. We setup a two-task experiment including subject identification and walking speed prediction. In Fig.7.13, the best reported average performance of subject identification is compared with other six methods including PSA [7], FD [8], MHI-HOG [9], GEI-HOG [10], TAMHI [11] and STIP [6]. The proposed method significantly outperforms other existing methods.

7.5 Conclusion

This chapter proposed a novel approach to identify human gait and predict multiple attributes using CNN and multi-task learning model. The approach was evaluated on the CASIA-B public gait benchmark dataset, achieving over 95% accuracy for each task. To the authors's

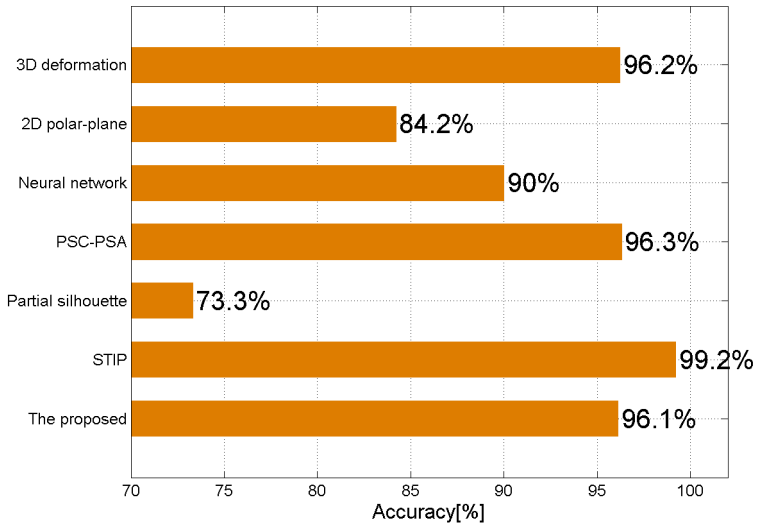


Fig. 7.12 Performance on CASIA-A: Best reported subject identification accuracy is compared with other six approaches including 3D deformation [1], 2D polar-plane [2], Neural network [3], PSC-PSA [4], Partial silhouette [5] and STIP [6].

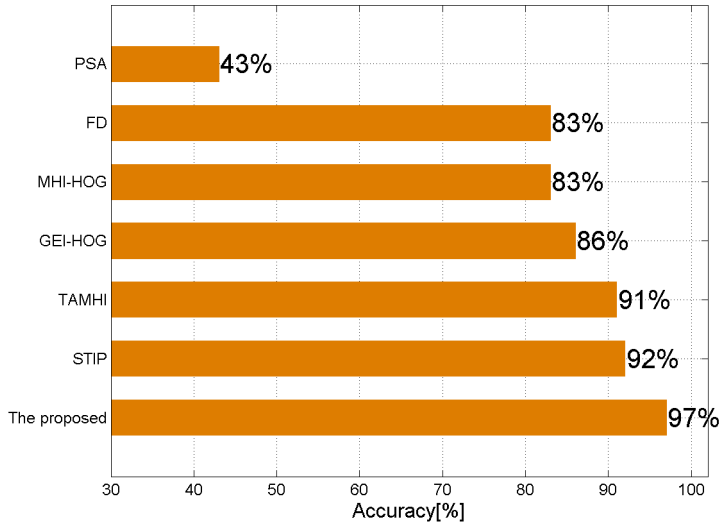


Fig. 7.13 Performance on Treadmill dataset A: Best reported subject identification accuracy is compared with other six approaches including PSA [7], FD [8], MHI-HOG [9], GEI-HOG [10], TAMHI [11] and STIP [6].

best knowledge, this is the first time that multi-attributes learning are applied to gait identification. The experimental result shows that our method generally outperform the state-of-art methods in the perspective of accuracy and robustness. Besides, our design can also provides more attributes information along with the identification. As the strength of the deep learning model, it is able to achieve competitive performance while dealing with large data processing. Thus, the performance of proposed method could be improved while dealing with large amount of data. It has the potential to be implemented in the real world practice. Although the model training requires large computational cost, the training process is offline and it can be accelerated via parallel GPU programming.

Chapter 8

Conclusion

In this chapter, a brief summary of the main contributions in this thesis is given below.

1. The driver distraction activities including operating the shift lever, talking on a cell phone, eating, and smoking, are explored to be recognised under the framework of human action recognition. Many published works on drivers' posture based on static images from drivers' action sequence 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 [29] which is based on the combination of basic movements, we propose driving activity as space-time action instead of static space-limited posture. (Chapter 3)

2. The proposal of a global grid-based representation for the driving actions, which is a combination of the motion history image (MHI) [30] and pyramid histogram of oriented gradients (POHG) [31], and the application of random forest classifier (RF) for the driving actions recognition. (Chapter 3)

3. A two stage intensity normalization preprocessing technique to minimize the influence from illumination variation. The first stage comprised a moving average method that smoothed the intensity variation caused by periodic lighting change. The second stage comprised application of the three frame difference method[32] to detect motion. For the task of motion detection and segmentation in video, it was found that the proposed two-stage pre-processing technique performed well in context of compensating for noise and illumination variation in video data. (Chapter 4)

4. A hierarchal classification system for driving behaviour recognition which considers different sets of features at different levels. Hierarchical classification is specifically intended for data where the features of interest can be arranged in a hierarchical manner. As such it offers advantages in terms of learning and representation in comparison to attempts to use "flat" classification techniques for the purposed of classifying hierarchical data[33]. These efficiency gains are realised because only a subset of the complete set of available features is considered at each node in the hierarchy. Hierarchical classification schemes have been applied in many areas [34–36]. However, it should be noted here that, to the best knowledge of the authors, they have not been applied to driving behaviour recognition. (Chapter 4)

5. To recognise driving posture, eye state, mouth state, ear state and pedestrian subject, we proposed to build a deep convolutional neural network in which trainable filters and local neighborhood pooling operations are applied alternatively to automatically explore salient features. Using CNN to learn rich features from the training set is more generic and requires minimal domain knowledge of the problem compared to hand crafted feature in previous approaches. (Chapter 5, Chapter 6, and Chapter 7)

6. We proposed to applied Face++ Research Toolkit [27] to localize the facial landmark [28] on the driver's face, which is used to propose the region of eye, mouth and ear. It is much robust under the effect of illumination variation and occlusion in real driving condition than previous approaches. (Chapter 6)

7. Multi-task model is a machine learning approach that jointly trains one task together with other related tasks at the same time sharing the same lower feature layers, which uses the commonality among tasks and therefore learns shared feature representation benefits all tasks. Since the difficulties in human gait identification are mainly caused by the multi factor's effects, it is very natural to use multi-tasks learning to simultaneously identify the multiple attributes of the gait. Thus, the approach in this chapter aims to investigates a convolutional neural network model for identifying the human gait while simultaneously predicting other human attributes at same time. To the best of our knowledge, this is the first approach that using MTL to investigate how human gait can be identified together with other auxiliary tasks. (Chapter 7)

References

[1] M. Goffredo, J.N. Carter, and M.S. Nixon. Front-view gait recognition. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–6, Sept 2008.

[2] Shi Chen and Youxing Gao. An invariant appearance model for gait recognition. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1375–1378, July 2007.

[3] Heesung Lee, Sungjun Hong, and Euntai Kim. An efficient gait recognition based on a selective neural network ensemble. *International Journal of Imaging Systems and Technology*, 18(4):237–241, 2008.

[4] W. Kusakunniran, Qiang Wu, Jian Zhang, and Hongdong Li. Pairwise shape configuration-based psa for gait recognition under small viewing angle change. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 17–22, Aug 2011.

[5] S.H. Shaikh, K. Saeed, and N. Chaki. Gait recognition using partial silhouette-based approach. In *Signal Processing and Integrated Networks (SPIN), 2014 International Conference on*, pages 101–106, Feb 2014.

[6] Worapan Kusakunniran. Attribute-based learning for gait recognition using spatio-temporal interest points. *Image and Vision Computing*, 32(12):1117 – 1126, 2014.

[7] Liang Wang, Tieniu Tan, Weiming Hu, and Huazhong Ning. Automatic gait recognition based on statistical shape analysis. *Image Processing, IEEE Transactions on*, 12(9):1120–1131, Sept 2003.

[8] Chin Poo Lee, Alan W.C. Tan, and Shing Chiang Tan. Gait recognition via optimally interpolated deformable contours. *Pattern Recognition Letters*, 34(6):663 – 669, 2013.

[9] Chin-Pan Huang, Chaur-Heh Hsieh, Kuan-Ting Lai, and Wei-Yang Huang. Human action recognition using histogram of oriented gradient of motion history image. In *Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on*, pages 353–356, Oct 2011.

[10] TenikaP. Whytock, Alexander Belyaev, and NeilM. Robertson. Improving robustness and precision in gei + hog action recognition. In *Advances in Visual Computing*, volume 8033 of *Lecture Notes in Computer Science*, pages 119–128. Springer Berlin Heidelberg, 2013.

[11] Chin-Poo Lee, Alan W. C. Tan, and Shing Chiang Tan. Time-sliced averaged motion history image for gait recognition. *J. Visual Communication and Image Representation*, 25(5):822–826, 2014.

[12] Online. Who world report on road traffic injury prevention, 2004. http://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/en/.

[13] Online. Traffic safety facts 2012: A compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system, 2012. http://www-nrd.nhtsa.dot.gov/Pubs/812032.pdf.

[14] Online. Transportation forecast: Light duty vehicles, 2014. http://www.navigantresearch.com/research/transportation-forecast-light-duty-vehicles.

[15] L.M. Bergasa, J. Nuevo, M.A. Sotelo, R. Barea, and M.E. Lopez. Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):63–77, March 2006.

[16] Qiang Ji, Zhiwei Zhu, and P. Lan. Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE Transactions on Vehicular Technology*, 53(4):1052–1068, July 2004.

[17] M.M. Trivedi, T. Gandhi, and J. McCall. Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *Intelligent Transportation Systems, IEEE Transactions on*, 8(1):108–120, March 2007.

[18] A. Doshi and M.M. Trivedi. On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes. *Intelligent Transportation Systems, IEEE Transactions on*, 10(3):453–462, Sept 2009.

[19] Chunsheng Liu, Faliang Chang, and Zhenxue Chen. Rapid multiclass traffic sign detection in high-resolution images. *Intelligent Transportation Systems, IEEE Transactions on*, 15(6):2394–2403, Dec 2014.

[20] E.D. Dickmanns, B. Mysliwetz, and T. Christians. An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(6):1273–1284, Nov 1990.

[21] D. Pomerleau and T. Jochem. Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert*, 11(2):19–27, Apr 1996.

[22] U. Franke, D. Gavrila, S. Gorzig, Frank Lindner, F. Puetzold, and C. Wohler. Autonomous driving goes downtown. *Intelligent Systems and their Applications, IEEE*, 13(6):40–48, Nov 1998.

[23] alessandra fascioli massimo bertozzi, alberto broggi. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 2000.

[24] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta. Artificial vision in road vehicles. *Proceedings of the IEEE*, 90(7):1258–1271, Jul 2002.

[25] F. Heimes and H.-H. Nagel. Towards active machine-vision-based driver assistance for urban areas. *International Journal of Computer Vision*, 50(1):5–34, 2002.

[26] Wilfried Enkelmann. Video-based driver assistance–from basic functions to applications. *International Journal of Computer Vision*, 45(3):201–221, 2001.

[27] Megvii Inc. Face++ research toolkit. www.faceplusplus.com, December 2013.

[28] Erjin Zhou, Haoqiang Fan, Zhimin Cao, Yuning Jiang, and Qi Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 386–391, Dec 2013.

[29] A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006. Special Issue on Modeling People: Vision-based understanding of a person's shape, appearance, movement and behaviour.

[30] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, Mar 2001.

[31] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 401–408, New York, NY, USA, 2007. ACM.

[32] Marie-Pierre Dubuisson and Anil K. Jain. Contour extraction of moving objects in complex outdoor scenes. *International Journal of Computer Vision*, 14(1):83–105, 1995.

[33] A. Zimek, F. Buchwald, E. Frank, and S. Kramer. A study of hierarchical and flat classification of proteins. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 7(3):563–571, July 2010.

[34] Jian xiong Dong, L. Devroye, and C.Y. Suen. Fast svm training algorithm with decomposition on very large data sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):603–618, April 2005.

[35] H. Sahbi and D. Geman. A hierarchy of support vector machines for pattern detection. *Journal of Machine Learning Research*, 7:2087–2123, 2006.

[36] Carlos N., Silla Jr., and Alex A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.

[37] Online, Accessed 2015. https://en.wikipedia.org/wiki/Advanced_driver_assistance_systems.

[38] F.-Y. Wang, C. Herget, and D. Zeng. Guest editorial developing and improving transportation systems: The structure and operation of ieee intelligent transportation systems society. *Intelligent Transportation Systems, IEEE Transactions on*, 6(3):261–264, Sept 2005.

[39] Online, Accessed 2015. https://en.wikipedia.org/wiki/Intelligent_transportation_system.

[40] Xinping Yan, Hui Zhang, and Chaozhong Wu. Research and development of intelligent transportation systems. In *Distributed Computing and Applications to Business, Engineering Science (DCABES), 2012 11th International Symposium on*, pages 321–327, Oct 2012.

[41] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner. Three decades of driver assistance systems: Review and future perspectives. *Intelligent Transportation Systems Magazine, IEEE*, 6(4):6–22, winter 2014.

[42] Anton van Zanten and Friedrich Kost. Bremsenbasierte assistenzfunktionen. In Hermann Winner, Stephan Hakuli, and Gabriele Wolf, editors, *Handbuch Fahrerassistenzsysteme*, pages 356–394. Vieweg+Teubner Verlag, 2012.

[43] M. Aga and A. Ogada. Analysis of vehicle stability control effectiveness from accident data. *Proceedings of 18th International Technical Conference on the Enhanced Safety of Vehicles Held Nagoya Japan 19 22 May 2003*, 2003.

[44] J. Y. LeCoz R. Sferco, Y. Page and P. Fay. Potential effectiveness of the electronic stability programs-what european field studies tell us. *Proceedings of 17th International Technical Conference on the Enhanced Safety of Vehicles Cd Rom*, 2001.

[45] Jane C. Stutts, Donald W. Reinfurt, Loren Staplin, and Eric A. Rodgman. The role of driver distraction in traffic crashes. *AAA Foundation for Traffic Safety*, 2001.

[46] S. Kaplan, M.A. Guvensan, A.G. Yavuz, and Y. Karalurt. Driver behavior analysis for safe driving: A survey. *Intelligent Transportation Systems, IEEE Transactions on*, PP(99):1–16, 2015.

[47] E. Wahlstrom, O. Masoud, and N. Papanikolopoulos. Vision-based methods for driver monitoring. In *Proceedings of the IEEE Intelligent Transportation Systems*, volume 2, pages 903–908, Shanghai, China, Octuber 2003.

[48] 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*, pages 339–344, Parma, Italy, June 2004.

[49] Xuetao Zhang, Nanning Zheng, Fan Mu, and Yongjian He. Head pose estimation using isophote features for driver assistance systems. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 568–572, Xi'an, China, June 2009.

[50] P. Watta, S. Lakshmanan, and Yulin Hou. Nonparametric approaches for estimating driver pose. *Vehicular Technology, IEEE Transactions on*, 56(4):2028–2041, July 2007.

[51] E. Murphy-Chutorian and M.M. Trivedi. Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):300–311, June 2010.

[52] D. Demirdjian and C. Varri. Driver pose estimation with 3d time-of-flight sensor. In *IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*, pages 16–22, Nashville, USA, March 2009.

[53] Cuong Tran and M.M. Trivedi. Towards a vision-based system exploring 3d driver posture dynamics for driver assistance: Issues and possibilities. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 179–184, San Diego, USA, June 2010.

[54] Shinko Y. Cheng, Sangho Park, and Mohan M. Trivedi. Multi-spectral and multi-perspective video arrays for driver body tracking and activity analysis. volume 106, pages 245–257, 2007.

[55] Harini Veeraraghavan, Nathaniel Bird, Stefan Atev, and Nikolaos Papanikolopoulos. Classifiers for driver activity monitoring. *Transportation Research Part C: Emerging Technologies*, 15(1):51 – 67, 2007.

[56] S.Y. Cheng and M.M. Trivedi. Vision-based infotainment user determination by hand recognition for driver assistance. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):759–764, Sept 2010.

[57] Cuong Tran, Anup Doshi, and Mohan Manubhai Trivedi. Modeling and prediction of driver behavior by foot gesture analysis. *Computer Vision and Image Understanding*, 116(3):435 – 445, 2012. Special issue on Semantic Understanding of Human Behaviors in Image Sequences.

[58] Bing-Fei Wu, Ying-Han Chen, and Chung-Hsuan Yeh. Driving behaviour-based event data recorder. *Intelligent Transport Systems, IET*, 8(4):361–367, June 2014.

[59] F. Jiménez, J.E. Naranjo, and O. Gómez. Autonomous collision avoidance system based on accurate knowledge of the vehicle surroundings. *Intelligent Transport Systems, IET*, 9(1):105–117, 2015.

[60] Birgitta Thorslund. Electrooculogram analysis and development of a system for defining stages of drowsiness. 2004.

[61] M. Tada, H. Noma, A. Utsumi, M. Segawa, M. Okada, and K. Renge. Elderly driver retraining using automatic evaluation system of safe driving skill. *Intelligent Transport Systems, IET*, 8(3):266–272, May 2014.

[62] A. Punitha, M.K. Geetha, and A. Sivaprakash. Driver fatigue monitoring system based on eye state analysis. In *Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference on*, pages 1405–1408, March 2014.

[63] A.B. Albu, B. Widsten, Tiange Wang, J. Lan, and J. Mah. A computer vision-based system for real-time detection of sleep onset in fatigued drivers. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 25–30, June 2008.

[64] Xinghua Sun, Lu Xu, and Jingyu Yang. Driver fatigue alarm based on eye detection and gaze estimation, 2007.

[65] D.Jayanthi and M.Bommy. *International Journal of Engineering and Advanced Technology*, (1):238, 2012.

[66] A.K. Jain, R.P.W. Duin, and Jianchang Mao. Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, Jan 2000.

[67] A.M. Malla, Paul R. Davidson, P.J. Bones, R. Green, and R.D. Jones. Automated video-based measurement of eye closure for detecting behavioral microsleep. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 6741–6744, Aug 2010.

[68] AA Lenskiy and JS Lee. Driver's eye blinking detection using novel color and texture segmentation algorithms. *INTERNATIONAL JOURNAL OF CONTROL AUTOMATION AND SYSTEMS*, 10(2):317 – 327, n.d.

[69] Jian-Da Wu and Tuo-Rung Chen. Development of a drowsiness warning system based on the fuzzy logic images analysis. *Expert Systems With Applications*, 34:1556 – 1561, 2008.

[70] *PERCLOS, a valid psychophysiological measure of alertness as assessed by psychomotor vigilance [microform].* Techbrief. Washington, DC (400 Seventh St., SW, Room 3107, Washington 20590) : Federal Highway Administration, Office of Motor Carriers, Office of Motor Carrier Research and Standards, [1998], 1998.

[71] alice caplier alexandre benoit. Hypovigilence analysis: open or closed eye or mouth? blinking or yawning frequency? *Advanced Video and Signal Based Surveillance 2005 Avss 2005 Ieee Conference on*, pages 207–212, 2005.

[72] preeti r bajaj mandalapu saradadevi. Driver fatigue detection using mouth and yawning analysis. 2008.

[73] Driver drowsiness monitoring based on yawning detection. 2012.

[74] Tiesheng Wang and Pengfei Shi. Yawning detection for determining driver drowsiness. In *VLSI Design and Video Technology, 2005. Proceedings of 2005 IEEE International Workshop on*, pages 373–376, May 2005.

[75] J Jo, SJ Lee, HG Jung, KR Park, and J Kim. Vision-based method for detecting driver drowsiness and distraction in driver monitoring system. *OPTICAL ENGINEERING*, 50(12), n.d.

[76] Xiao Fan, Bao-Cai Yin, and Yan-Feng Sun. Yawning detection for monitoring driver fatigue. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 2, pages 664–668, Aug 2007.

[77] Esra Vural, Mujdat Cetin, Aytul Ercil, Gwen Littlewort, Marian Bartlett, and Javier Movellan. Drowsy driver detection through facial movement analysis. In *Human-Computer Interaction*, volume 4796 of *Lecture Notes in Computer Science*, pages 6–18. 2007.

[78] Paul Viola and MichaelJ. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[79] Azim Eskandarian. *Advanced driver fatigue research [electronic resource] / [Azim Eskandarian ...et al.].* [Washington, D.C.] : U.S. Dept. of Transportation, Federal Motor Carrier Safety Administration, [2007], 2007.

[80] PROF. V.K.BANGA ITENDERPAL SINGH. Development of a drowsiness warning system using neural network. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, (8):3614, 2013.

[81] Vijayalaxmi, P. Sudhakara Rao, and S. Sreehari. Neural network approach for eye detection. 2012.

[82] N.V. Boulgouris, D. Hatzinakos, and K.N. Plataniotis. Gait recognition: a challenging signal processing technology for biometric identification. *Signal Processing Magazine, IEEE*, 22(6):78–90, Nov 2005.

[83] TraceyK.M. Lee, Mohammed Belkhatir, and Saeid Sanei. A comprehensive review of past and present vision-based techniques for gait recognition. *Multimedia Tools and Applications*, 72(3):2833–2869, 2014.

[84] M.S. Nixon and J.N. Carter. Automatic recognition by gait. *Proceedings of the IEEE*, 94(11):2013–2024, Nov 2006.

[85] TraceyK.M. Lee, Mohammed Belkhatir, and Saeid Sanei. A comprehensive review of past and present vision-based techniques for gait recognition. *Multimedia Tools and Applications*, 72(3):2833–2869, 2014.

[86] Jin Wang, M. She, S. Nahavandi, and A. Kouzani. A review of vision-based gait recognition methods for human identification. In *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, pages 320–327, Dec 2010.

[87] Imed Bouchrika and MarkS. Nixon. Model-based feature extraction for gait analysis and recognition. In Andre Gagalowicz and Wilfried Philips, editors, *Computer Vision/Computer Graphics Collaboration Techniques*, volume 4418 of *Lecture Notes in Computer Science*, pages 150–160. Springer Berlin Heidelberg, 2007.

[88] Chew-Yean Yam and MarkS. Nixon. Gait recognition, model-based. In StanZ. Li and Anil Jain, editors, *Encyclopedia of Biometrics*, pages 633–639. Springer US, 2009.

[89] C. BenAbdelkader, R. Cutler, and L. Davis. Stride and cadence as a biometric in automatic person identification and verification. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 372–377, May 2002.

[90] A.F. Bobick and A.Y. Johnson. Gait recognition using static, activity-specific parameters. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–423–I–430 vol.1, 2001.

[91] Jang-Hee Yoo, Doosung Hwang, Ki-Young Moon, and M.S. Nixon. Automated human recognition by gait using neural network. In *Image Processing Theory, Tools and Applications, 2008. IPTA 2008. First Workshops on*, pages 1–6, Nov 2008.

[92] R. Tanawongsuwan and A. Bobick. Gait recognition from time-normalized joint-angle trajectories in the walking plane. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–726–II–731 vol.2, 2001.

[93] ChewYean Yam, Mark S. Nixon, and John N. Carter. Automated person recognition by walking and running via model-based approaches. *Pattern Recognition*, 37(5):1057 – 1072, 2004.

[94] David Cunado, Mark S. Nixon, and John N. Carter. Automatic extraction and description of human gait models for recognition purposes. *Computer Vision and Image Understanding*, 90(1):1 – 41, 2003.

[95] Dynamic feature extraction via the velocity hough transform. *Pattern Recognition Letters*, 18(10):1035 – 1047, 1997.

[96] Liang Wang, Huazhong Ning, Tieniu Tan, and Weiming Hu. Fusion of static and dynamic body biometrics for gait recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(2):149–158, Feb 2004.

[97] Human gait recognition based on matching of body components. *Pattern Recognition*, 40(6):1763 – 1770, 2007.

[98] Xuelong Li, S.J. Maybank, Shuicheng Yan, Dacheng Tao, and Dong Xu. Gait components and their application to gender recognition. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):145–155, March 2008.

[99] S. Sarkar, P.J. Phillips, Z. Liu, I.R. Vega, P. Grother, and K.W. Bowyer. The humanid gait challenge problem: data sets, performance, and analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(2):162–177, Feb 2005.

[100] J. Han and B. Bhanu. Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):316–322, Feb 2006.

[101] Jianyi Liu and Nanning Zheng. Gait history image: A novel temporal template for gait recognition. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 663–666, July 2007.

[102] Frame difference energy image for gait recognition with incomplete silhouettes. *Pattern Recognition Letters*, 30(11):977 – 984, 2009.

[103] Silhouette quality quantification for gait sequence analysis and recognition. *Signal Processing*, 89(7):1417 – 1427, 2009.

[104] Infrared gait recognition based on wavelet transform and support vector machine. *Pattern Recognition*, 43(8):2904 – 2910, 2010.

[105] A. Kale, A.N. Rajagopalan, N. Cuntoor, and V. Kruger. Gait-based recognition of humans using continuous hmms. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 336–341, May 2002.

[106] A. Kale, A. Sundaresan, A.N. Rajagopalan, N.P. Cuntoor, A.K. Roy-Chowdhury, V. Kruger, and R. Chellappa. Identification of humans using gait. *Image Processing, IEEE Transactions on*, 13(9):1163–1173, Sept 2004.

[107] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1505–1518, Dec 2003.

[108] F. Dadashi, B.N. Araabi, and H. Soltanian-Zadeh. Gait recognition using wavelet packet silhouette representation and transductive support vector machines. In *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, pages 1–5, Oct 2009.

[109] N.V. Boulgouris and Z.X. Chi. Gait recognition using radon transform and linear discriminant analysis. *Image Processing, IEEE Transactions on*, 16(3):731–740, March 2007.

[110] Chen Wang, Junping Zhang, Liang Wang, Jian Pu, and Xiaoru Yuan. Human identification using temporal information preserving gait template. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2164–2176, Nov 2012.

[111] Z. Liu and S. Sarkar. Improved gait recognition by gait dynamics normalization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(6):863–876, June 2006.

[112] Sruti Das Choudhury and Tardi Tjahjadi. Gait recognition based on shape and motion analysis of silhouette contours. *Computer Vision and Image Understanding*, 117(12):1770 – 1785, 2013.

[113] Z. Liu and S. Sarkar. Simplest representation yet for gait recognition: averaged silhouette. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 211–214 Vol.4, Aug 2004.

[114] Dong Xu, Shuicheng Yan, Dacheng Tao, Lei Zhang, Xuelong Li, and Hong-Jiang Zhang. Human gait recognition with matrix representation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 16(7):896–903, July 2006.

[115] Junping Zhang, Jian Pu, Changyou Chen, and R. Fleischer. Low-resolution gait recognition. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(4):986–996, Aug 2010.

[116] Toby H.W. Lam, K.H. Cheung, and James N.K. Liu. Gait flow image: A silhouette-based gait representation for human identification. *Pattern Recognition*, 44(4):973 – 987, 2011.

[117] Dong Xu, Yi Huang, Zinan Zeng, and Xinxing Xu. Human gait recognition using patch distribution feature and locality-constrained group sparse representation. *Image Processing, IEEE Transactions on*, 21(1):316–326, Jan 2012.

[118] W. Kusakunniran, Qiang Wu, Jian Zhang, and Hongdong Li. Gait recognition under various viewing angles based on correlated motion regression. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(6):966–980, June 2012.

[119] Yasushi Makihara, Ryusuke Sagawa, Yasuhiro Mukaigawa, Tomio Echigo, and Yasushi Yagi. Gait recognition using a view transformation model in the frequency domain. In Ale? Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision - ECCV 2006*, volume 3953 of *Lecture Notes in Computer Science*, pages 151–163. Springer Berlin Heidelberg, 2006.

[120] Khalid Bashir, Tao Xiang, and Shaogang Gong. Gait recognition without subject cooperation. *Pattern Recognition Letters*, 31(13):2052 – 2060, 2010. Meta-heuristic Intelligence Based Image Processing.

[121] Maodi Hu, Yunhong Wang, Zhaoxiang Zhang, J.J. Little, and Di Huang. View-invariant discriminative projection for multi-view gait-based human identification. *Information Forensics and Security, IEEE Transactions on*, 8(12):2034–2045, Dec 2013.

[122] Haifeng Hu. Enhanced gabor feature based classification using a regularized locally tensor discriminant model for multiview gait recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(7):1274–1286, July 2013.

[123] W. Kusakunniran, Qiang Wu, Jian Zhang, Hongdong Li, and Liang Wang. Recognizing gaits across views through correlated motion co-clustering. *Image Processing, IEEE Transactions on*, 23(2):696–709, Feb 2014.

[124] Jiwen Lu, Gang Wang, and P. Moulin. Human identity and gender recognition from gait sequences with arbitrary walking directions. *Information Forensics and Security, IEEE Transactions on*, 9(1):51–61, Jan 2014.

[125] Worapan Kusakunniran, Qiang Wu, Jian Zhang, and Hongdong Li. Cross-view and multi-view gait recognitions based on view transformation model using multi-layer perceptron. *Pattern Recognition Letters*, 33(7):882 – 889, 2012. Special Issue on Awards from {ICPR} 2010.

[126] Shiqi Yu, Tieniu Tan, Kaiqi Huang, Kui Jia, and Xinyu Wu. A study on gait-based gender classification. *Image Processing, IEEE Transactions on*, 18(8):1905–1910, Aug 2009.

[127] Maodi Hu, Yunhong Wang, Zhaoxiang Zhang, and De Zhang. Gait-based gender classification using mixed conditional random field. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(5):1429–1439, Oct 2011.

[128] Jiwen Lu and Yap-Peng Tan. Gait-based human age estimation. *Information Forensics and Security, IEEE Transactions on*, 5(4):761–770, Dec 2010.

[129] Y. Makihara, M. Okumura, H. Iwama, and Y. Yagi. Gait-based age estimation using a whole-generation gait database. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–6, Oct 2011.

[130] R. Grace, V.E. Byrne, D.M. Bierman, J.-M. Legrand, D. Gricourt, B.K. Davis, J.J. Staszewski, and B. Carnahan. A drowsy driver detection system for heavy vehicles. In *Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE*, volume 2, pages I36/1–I36/8 vol.2, Oct 1998.

[131] E. Wahlstrom, O. Masoud, and N. Papanikolopoulos. Vision-based methods for driver monitoring. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 2, pages 903–908 vol.2, Oct 2003.

[132] Jaeik Jo, Sung Joo Lee, Jaihie Kim, Ho Gi Jung, and Kang Ryoung Park. Vision-based method for detecting driver drowsiness and distraction in driver monitoring system. *Optical Engineering*, 50(12):127202–127224, December 2011.

[133] Xia Liu, Youding Zhu, and K. Fujimura. Real-time pose classification for driver monitoring. In *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, pages 174–178, September 2002.

[134] T. Kato, T. Fujii, and M. Tanimoto. Detection of driver's posture in the car by using far infrared camera. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 339–344, June 2004.

[135] H. Veeraraghavan, S. Atev, N. Bird, P. Schrater, and N. Papanikolopoulos. Driver activity monitoring through supervised and unsupervised learning. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 580–585, Sept 2005.

[136] C.H. Zhao, B.L. Zhang, J. He, and J. Lian. Recognition of driving postures by contourlet transform and random forests. *Intelligent Transport Systems, IET*, 6(2):161–168, June 2012.

[137] ChihangH. Zhao, BailingL. Zhang, XiaozhengZ. Zhang, SanqiangQ. Zhao, and HanxiX. Li. Recognition of driving postures by combined features and random subspace ensemble of multilayer perceptron classifiers. *Neural Computing and Applications*, 22(1):175–184, 2013.

[138] J.C. McCall and M.M. Trivedi. Visual context capture and analysis for driver attention monitoring. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 332–337, Oct 2004.

[139] K. Torkkola, N. Massey, and C. Wood. Driver inattention detection through intelligent analysis of readily available sensors. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 326–331, Oct 2004.

[140] D.A. Johnson and M.M. Trivedi. Driving style recognition using a smartphone as a sensor platform. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1609–1615, Oct 2011.

[141] Vigilance monitoring for operator safety: A simulation study on highway driving. *Journal of Safety Research*, 37(2):139 – 147, 2006.

[142] George Rigas, Yorgos Goletsis, Panagiota Bougia, and Dimitrios I. Fotiadis. Towards driver's state recognition on real driving conditions. *International Journal of Vehicular Technology*, 2011:1 – 14, 2011.

[143] MH Kutila, M Jokela, T Makinen, J Viitanen, G Markkula, and TW Victor. Driver cognitive distraction detection: feature estimation and implementation. *PROCEEDINGS OF THE INSTITUTION OF MECHANICAL ENGINEERS PART D-JOURNAL OF AUTOMOBILE ENGINEERING*, 221(9):1027–1040, 2007.

[144] Ying Wang, Kaiqi Huang, and Tieniu Tan. Human activity recognition based on r transform. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[145] Zhang Fan, Guo Li, Lu Haixian, Gui Shu, and Li Jinkui. Star skeleton for human behavior recognition. In *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*, pages 1046–1050, July 2012.

[146] Liang Wang and D. Suter. Informative shape representations for human action recognition. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 1266–1269, August 2006.

[147] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 726–733 vol.2, Oct 2003.

[148] M.A.R. Ahad, T. Ogata, J.K. Tan, H.S. Kim, and S. Ishikawa. Motion recognition approach to solve overwriting in complex actions. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–6, Sept 2008.

[149] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(2):288–303, Feb 2010.

[150] Somayeh Danafar and Niloofar Gheissari. Action recognition for surveillance applications using optic flow and svm. In *Computer Vision - ACCV 2007*, volume 4844 of *Lecture Notes in Computer Science*, pages 457–466. 2007.

[151] I. Laptev and T. Lindegerg. Space-time interest points. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, volume 1, pages 432–439, Oct 2003.

[152] Ivan Laptev, Barbara Caputo, Christian Schuldt, and Tony Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, 108(3):207 – 229, 2007. Special Issue on Spatiotemporal Coherence for Visual Motion Analysis.

[153] S. Park and M. Trivedi. Driver activity analysis for intelligent vehicles: issues and development framework. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 644–649, June 2005.

[154] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[155] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, Jan 1979.

[156] G.R. Bradski and J. Davis. Motion segmentation and pose recognition with motion history gradients. In *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, pages 238–244, 2000.

[157] Osama Masoud and Nikos Papanikolopoulos. A method for human action recognition. *Image and Vision Computing*, 21(8):729 – 743, 2003.

[158] Haoran Yi, Deepu Rajan, and Liang-Tien Chia. A new motion histogram to index motion content in video segments. *Pattern Recognition Letters*, 26(9):1221 – 1231, 2005.

[159] Jan Flusser, Tomas Suk, and Barbara Zitov. *Moments and moment invariants in pattern recognition.* Online access with subscription: Ebrary. Chichester, West Sussex, U.K. ; J. Wiley, 2009., 2009.

[160] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178, 2006.

[161] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[162] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[163] C. Yan, F. Coenen, and B. Zhang. Driving posture recognition by joint application of motion history image and pyramid histogram of oriented gradients. *International Journal of Vehicular Technology*, 2014, 2014.

[164] Chihang Zhao, Yongsheng Gao, Jie He, and Jie Lian. Recognition of driving postures by multiwavelet transform and multilayer perceptron classifier. *Engineering Applications of Artificial Intelligence*, 25(8):1677 – 1686, 2012.

[165] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[166] Di Wu and Ling Shao. Silhouette analysis-based action recognition via exploiting human poses. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):236–243, Feb 2013.

[167] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, Dec 2007.

[168] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099–3104 vol.4, Oct 2004.

[169] J. Schmudderich, V. Willert, J. Eggert, S. Rebhan, C. Goerick, G. Sagerer, and E. Korner. Estimating object proper motion using optical flow, kinematics, and depth information. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(4):1139–1151, Aug 2008.

[170] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):827–832, May 2005.

[171] Fan-Chieh Cheng, Shih-Chia Huang, and Shanq-Jang Ruan. Illumination-sensitive background modeling approach for accurate moving object detection. *IEEE Transactions on Broadcasting*, 57(4):794–801, Dec 2011.

[172] JinMin Choi, Hyung Jin Chang, Yung Jun Yoo, and Jin Young Choi. Robust moving object detection against fast illumination change. *Computer Vision and Image Understanding*, 116(2):179–193, 2012.

[173] J.-E. Ha and W.-H. Lee. Foreground objects detection using multiple difference images. *Optical Engineering*, 49(4), 2010.

[174] Jalal A. Nasiri, Nasrollah Moghadam Charkari, and Kourosh Mozafari. Energy-based model of least squares twin support vector machines for human action recognition. *Signal Processing*, 104(0):248 – 257, 2014.

[175] S.N. Vitaladevuni, V. Kellokumpu, and L.S. Davis. Action recognition using ballistic dynamics. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage , USA., June 2008.

[176] Cen Rao, Alper Yilmaz, and Mubarak Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[177] Zhou Feng and Tat-Jen Cham. Video-based human action classi.cation with ambiguous correspondences. In *Computer Vision and Pattern Recognition - Workshops*, pages 82–82, San Diego, USA, June 2005.

[178] Yan Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, Rio de Janeiro, Brazil, Oct 2007.

[179] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):844–851, Aug 2000.

[180] P. Peursum, H.H. Bui, S. Venkatesh, and G. West. Human action segmentation via controlled use of missing data in hmms. In *17th International Conference on Pattern Recognition*, volume 4, pages 440–445 Vol.4, Cambridge, UK, Aug 2004.

[181] Xiaotao Zou and B. Bhanu. Human activity classification based on gait energy image and coevolutionary genetic programming. In *18th International Conference on Pattern Recognition*, volume 3, pages 556–559, Hong Kong, China, Aug 2006.

[182] H.-W. Lin, J.-L. Wu, and M.-C. Hu. *Gait-based action recognition via accelerated minimum incremental coding length classifier*, volume 7131 LNCS of *Lecture Notes in Computer Science*. Dept. of CSIE, National Taiwan University, 2012.

[183] Lin Chunli and Wang KeJun. A behavior classification based on enhanced gait energy image. In *2010 2nd International Conference on Networking and Digital Society*, volume 2, pages 589–592, Wenzhou, China, May 2010.

[184] Tenika Whytock, Alexander Belyaev, and Neil Robertson. *Improving robustness and precision in GEI + HOG action recognition*. Lecture Notes in Computer Science. 2013.

[185] V. Kecman. *Learning and soft computing [electronic book] : support vector machines, neural networks, and fuzzy logic models / Vojislav Kecman.* Complex adaptive systems. Cambridge, Mass. : MIT Press, 2001., 2001.

[186] Manuel Davy, Frederic Desobry, Arthur Gretton, and Christian Doncarli. An online support vector machine for abnormal events detection. *Signal Processing*, 86(8):2009–2025, 2006. Special Section: Advances in Signal Processing-assisted Cross-layer Designs.

[187] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[188] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. A skin tone detection algorithm for an adaptive approach to steganography. *Signal Processing*, 89(12):2465–2478, 2009. Special Section: Visual Information Analysis for Security.

[189] Wei Ren Tan, Chee Seng Chan, P. Yogarajah, and J. Condell. A fusion approach for efficient human skin detection. *IEEE Transactions on Industrial Informatics*, 8(1):138–147, Feb 2012.

[190] Ines Teyeb, Olfa Jemai, Mourad Zaied, and Chokri Ben Amar. A drowsy driver detection system based on a new method of head posture estimation. In Emilio Corchado, JoseA. Lozano, Hector Quintian, and Hujun Yin, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2014*, volume 8669 of *Lecture Notes in Computer Science*, pages 362–369. Springer International Publishing, 2014.

[191] I. Teyeb, O. Jemai, M. Zaied, and C. Ben Amar. A novel approach for drowsy driver detection using head posture estimation and eyes recognition system based on wavelet network. In *Information, Intelligence, Systems and Applications, IISA 2014, The 5th International Conference on*, pages 379–384, July 2014.

[192] C.H. Zhao, B.L. Zhang, J. He, and J. Lian. Recognition of driving postures by contourlet transform and random forests. *Intelligent Transport Systems, IET*, 6(2):161–168, June 2012.

[193] Olfa Jemai, Ines Teyeb, Tahani Bouchrika, and Chokri Ben amar. A novel approach for drowsy driver detection using eyes recognition system based on wavelet network. *International Journal of Recent Contributions from Engineering, Science & IT (iJES)*, 1(1):46–52, 2013.

[194] G Hinton, S Osindero, and Y Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, July 2006.

[195] Q.V. Le, W.Y. Zou, S.Y. Yeung, and A.Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368, June 2011.

[196] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[197] O. Abdel-Hamid, A.-R. Mohamed, Hui Jiang, Li Deng, G. Penn, and Dong Yu. Convolutional neural networks for speech recognition. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(10):1533–1545, Oct 2014.

[198] Qirong Mao, Ming Dong, Zhengwei Huang, and Yongzhao Zhan. Learning salient features for speech emotion recognition using convolutional neural networks. *Multimedia, IEEE Transactions on*, 16(8):2203–2213, Dec 2014.

[199] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2042–2050. Curran Associates, Inc., 2014.

[200] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[201] J. Krause, T. Gebru, Jia Deng, Li-Jia Li, and Li Fei-Fei. Learning features and parts for fine-grained recognition. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 26–33, Aug 2014.

[202] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc., 2014.

[203] Ning Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1637–1644, June 2014.

[204] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587, June 2014.

[205] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, Aug 2013.

[206] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1385–1392, Dec 2013.

[207] Dong Yi, Zhen Lei, Shengcai Liao, and S.Z. Li. Deep metric learning for person re-identification. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 34–39, Aug 2014.

[208] Y. Taigman, Ming Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708, June 2014.

[209] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1891–1898, June 2014.

[210] Jiquan Ngiam, Zhenghao Chen, Sonia A. Bhaskar, Pang W. Koh, and Andrew Y. Ng. Sparse filtering. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1125–1133. Curran Associates, Inc., 2011.

[211] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. *Journal of Machine Learning Research*, 15(Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, AISTATS 2011):315–323, 2011.

[212] Junqi Jin, Kun Fu, and Changshui Zhang. Traffic sign recognition with hinge loss trained convolutional neural networks. *Intelligent Transportation Systems, IEEE Transactions on*, 15(5):1991–2000, Oct 2014.

[213] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148:574 – 591, 1959.

[214] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jurgen Schmidhuber. Chapter 14: Gradient flow in recurrent nets: The difficulty of learning long-term dependencies: 14.2 exponential error decay. *Field Guide to Dynamical Recurrent Networks*, page 237, 2009.

[215] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 472–478. MIT Press, 2001.

[216] Y-Lan Boureau, Jean Ponce, and Yann Lecun. A theoretical analysis of feature pooling in visual recognition. In *27TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, HAIFA, ISRAEL*, 2010.

[217] Matthew D. Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *http://arxiv.org/abs/1301.3557*, abs/1301.3557, 2013.

[218] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153, Sept 2009.

[219] Zhen Dong, Mingtao Pei, Yang He, Ting Liu, Yanmei Dong, and Yunde Jia. Vehicle type classification using unsupervised convolutional neural network. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 172–177, Aug 2014.

[220] E.P. Simoncelli. Statistical models for images: compression, restoration and synthesis. In *Signals, Systems amp; Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, volume 1, pages 673–678 vol.1, Nov 1997.

[221] M. Bethge. Factorial coding of natural images: How effective are linear model in removing higher-order dependencies? *Journal of the Optical Society of America A*, Jun 2006.

[222] Nicolas Pinto, David D. Cox, and James J. Dicarlo. Why is Real-World Visual Object Recognition Hard? *PLOS Computational Biology*, 4, 2008.

[223] Siwei Lyu and E.P. Simoncelli. Nonlinear image representation using divisive normalization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[224] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

[225] Kevin P. Murphy. *Machine learning : a probabilistic perspective.* Adaptive computation and machine learning. Cambridge, Mass. : MIT Press, 2012., 2012.

[226] Online. http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/.

[227] D. Erhan, Y. Bengio, A. Courville, P.A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning?. *Journal of Machine Learning Research*, 11:625–660, 2010.

[228] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014.

[229] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pages 1096–1103, 2008.

[230] BRUNO A. OLSHAUSEN and DAVID J. FIELDt. Sparse coding with an overcomplete basis set: A strategy employed by v1 ? 1996.

[231] Honglak Lee, Chaitanya Ekanadham, and Andrew Y. Ng. Sparse deep belief net model for visual area v2. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 873–880. Curran Associates, Inc., 2008.

[232] Chihang Zhao, Yongsheng Gao, Jie He, and Jie Lian. Recognition of driving postures by multiwavelet transform and multilayer perceptron classifier. *Engineering Applications of Artificial Intelligence*, 25(8):1677 – 1686, 2012.

[233] ChihangH. Zhao, BailingL. Zhang, XiaozhengZ. Zhang, SanqiangQ. Zhao, and HanxiX. Li. Recognition of driving postures by combined features and random subspace ensemble of multilayer perceptron classifiers. *Neural Computing and Applications*, 22(1):175–184, 2013.

[234] Chihang Zhao, Bailing Zhang, and Jie He. Vision-based classification of driving postures by efficient feature extraction and bayesian approach. *Journal of Intelligent and Robotic Systems*, 72(3-4):483–495, 2013.

[235] DavidG. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[236] Deep convolutional neural networks for large-scale speech tasks. *Neural Networks*, 64(0):39 – 48, 2015. Special Issue on.

[237] H. Cecotti and A. Graser. Convolutional neural networks for p300 detection with application to brain-computer interfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):433–445, March 2011.

[238] Xueyun Chen, Shiming Xiang, Cheng-Lin Liu, and Chun-Hong Pan. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *Geoscience and Remote Sensing Letters, IEEE*, 11(10):1797–1801, Oct 2014.

[239] Jialue Fan, Wei Xu, Ying Wu, and Yihong Gong. Human tracking using convolutional neural networks. *Neural Networks, IEEE Transactions on*, 21(10):1610–1623, Oct 2010.

[240] Cha Zhang and Zhengyou Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 1036–1041, March 2014.

[241] Sijin Li, Zhi-Qiang Liu, and A.B. Chan. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 488–495, June 2014.

[242] Zhanpeng Zhang, Ping Luo, ChenChange Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014*, volume 8694 of *Lecture Notes in Computer Science*, pages 94–108. Springer International Publishing, 2014.

[243] Shiqi Yu, Daoliang Tan, and Tieniu Tan. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 441–444, 2006.

[244] W. Kusakunniran, Qiang Wu, Hongdong Li, and Jian Zhang. Multiple views gait recognition using view transformation model based on optimized gait energy image. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1058–1064, Sept 2009.

[245] W. Kusakunniran, Qiang Wu, Jian Zhang, and Hongdong Li. Support vector regression for multi-view gait recognition based on local motion feature selection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 974–981, June 2010.

[246] Khalid Bashir, Tao Xiang, and Shaogang Gong. Cross-view gait recognition using correlation strength. In *Proceedings of the British Machine Vision Conference*, pages 109.1–109.11. BMVA Press, 2010.

[247] R. Wagner, M. Thom, R. Schweiger, G. Palm, and A. Rothermel. Learning convolutional neural networks from few samples. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–7, Aug 2013.

[248] Honglak Lee, Roger B. Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 609–616, 2009.

[249] Quoc V. Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Y. Ng. Ica with reconstruction cost for efficient overcomplete feature learning. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1017–1025. Curran Associates, Inc., 2011.

[250] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1717–1724, June 2014.

[251] Maodi Hu, Yunhong Wang, Zhaoxiang Zhang, De Zhang, and J.J. Little. Incremental learning for video-based gait recognition with lbp flow. *Cybernetics, IEEE Transactions on*, 43(1):77–89, Feb 2013.

[252] M. Jeevan, N. Jain, M. Hanmandlu, and G. Chetty. Gait recognition based on gait pal and pal entropy image. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 4195–4199, Sept 2013.

[253] K. Bashir, Tao Xiang, and Shaogang Gong. Gait recognition using gait entropy image. In *Crime Detection and Prevention (ICDP 2009), 3rd International Conference on*, pages 1–6, Dec 2009.

[254] Y. Makihara, H. Mannami, A. Tsuji, M.A. Hossain, K. Sugiura, A. Mori, and Y. Yagi. The ou-isir gait database comprising the treadmill dataset. *IPSJ Trans. on Computer Vision and Applications*, 4:53–62, April 2012.