Chapter 8
Intelligent Vision Processing Technology for Advanced Driver Assistance Systems

Po-Chun Shen, Kuan-Hung Chen, Jui-Sheng Lee, Guan-Yu Chen, Yi-Ting Lin, Bing-Yang Cheng, Guo-An Jian, Hsiu-Cheng Chang, Wei-Ming Lu, and Jiun-In Guo

Abstract Intelligent vision processing technology has a wide range of applications on vehicles. Many of these applications are related to a so-called Advanced Driver Assistance System (ADAS). Collaborated with cameras, Pedestrian and Motorcyclist Detection System (PMD), Lane Departure Warning System (LDWS), Forward Collision Warning System (FCWS), Speed Limit Detection System (SLDS), and Dynamic Local Contrast Enhancement (DLCE) techniques can help drivers notice important events or objects around. This chapter gives an in-depth exploration for these intelligent vision processing technologies from the viewpoints of methodology development, algorithm optimization, and system implementation on embedded platforms. More precisely, this chapter tends to first give a survey and overview for newly appeared state-of-the-art intelligent vision processing technologies for ADAS, and then highlights some significant technologies including PMD, LDWS, FCWS, SLDS, and DLCE developed in System on Chip (SoC) Laboratory, Fong-Chia University, Taiwan, and intelligent Vision System (iVS) Laboratory, National Chiao Tung University, Taiwan. Besides, implementation and verification of the above ADAS technologies will also be presented. In summary, the proposed PMD design achieves 32.5 frame per second (fps) for 720 × 480 (D1) resolution on an AMD A10-7850K processor by using heterogeneous computing. On an automotive-grade Freescale i.MX6 (including 4-core ARM Cortex A9, 1 GB DDR3 RAM, and Linux environment) platform, the proposed LDWS, FCWS, and SLDS designs, respectively, achieve 33 fps, 32 fps, and 30 fps for D1 resolution. Finally, the proposed DLCE system is realized on a TREK-668 platform with an Intel Atom 1.6 GHz processor for real-time requirement of 50 fps at D1 resolution.

P.-C. Shen • J.-S. Lee • G.-Y. Chen • Y.-T. Lin • B.-Y. Cheng • G.-A. Jian
H.-C. Chang • J.-I. Guo
Department of Electronics Engineering, National Chiao Tung University,
Hsinchu City, Taiwan

K.-H. Chen (✉) • W.-M. Lu
Department of Electronics Engineering, Feng Chia University, Taichung City, Taiwan
e-mail: kuanhung@fcu.edu.tw

C.-M. Kyung et al. (eds.), *Smart Sensors and Systems*,
DOI 10.1007/978-3-319-33201-7_8

Keywords ADAS • FCWS • Intelligent vision processing • LDWS • Pedestrian detection

8.1 Introduction

According to the survey report of Gartner in 2015, autonomous vehicles attract the highest expectation among many other popular applications. Autonomous cars adopt sensors such as RAdio Detection And Ranging (RADAR), Light Detection And Ranging (LiDAR), and cameras to understand environment around. Among these sensors, cameras are rather inexpensive in cost consideration and mature in manufacturing aspect. However, we need an elaborated intelligent processing system to analyze the visual contents to construct sensing ability for autonomous cars. In addition, design trend for car safety moves from passive ways to active ones. In the USA, rear view monitoring becomes a standard equipment for new cars. Car manufacturers such as BMW, Lexus, and Infiniti have launched Around View Monitoring (AVM) adoption. In EU, Lane Departure Warning System (LDWS) is already a standard equipment for vehicles. Besides, car manufacturers such as BMW, Audi, Volvo, and Mercedes-Benz provide advanced options, e.g., Adaptive Cruise Control (ACC), Adaptive Front lighting System (AFS), Driver Status Monitoring (DSM), Blind Spot Detection (BSD), and so on, to customers. Moreover, search engine vendor Google also devotes to develop self-driving cars. Google self-driving cars adopt a LiDAR sensor for detecting objects around from a 3D-space viewpoint and millimeter wave RADAR sensors for distant object detection. As for pedestrian and bicycle detection, vision technology is adopted. A car electronics vendor worth mentioning is Mobileye, which provides vision Systems on Chip (SoCs) for safety driving to car manufacturers. Based on solid foundation of ARM and digital signal processor (DSP) experience, TI has also launched TDA2x/3x SoCs for Advanced Driver Assistance Systems (ADAS). These above facts reveal that intelligent vision processing technology for ADAS attracts high attention recently and will probably become critical in the upcoming 5–10 years. Accordingly, this chapter tends to give a survey and overview for newly appeared intelligent vision processing technology for ADAS in Sect. 8.2, and highlights some significant technologies including Pedestrian and Motorcyclist Detection System (PMD), LDWS, Forward Collision Warning System (FCWS), Speed Limit Detection System (SLDS), and Dynamic Local Contrast Enhancement (DLCE) developed in our laboratory in Sect. 8.3. Besides, implementation and verification of the above ADAS technology are also presented in Sect. 8.4. Finally, we end this chapter in Sect. 8.5, i.e., the conclusion.

8.2 Existing ADAS Systems

Machine learning leads the trend to support multiple-object detecting and distinguishing. How to generate good machine learning samples and simplify the complex architecture are still challenges to real-time processing with limit computing resource. Certain unique features are extracted to recognize one kind of object, which reduces the computation complexity with minor sacrifice of detection diversity. With inclement weather against techniques, ADAS systems perform even better. This section reviews the related works of intelligent vision processing technology for ADAS which can be categorized into PMD, LDWS, FCWS, SLDS, and DLCE in the following sub-sections.

8.2.1 Pedestrian and Motorcyclist Detection System

Various techniques have been developed for detecting moving objects commonly seen on the road, including pure pedestrians [1–21], pure vehicles [22–29], and multiple kinds of objects [30, 31]. Enhancing the detection rate and lowering down the false alarm rate are the two main objectives of all existing vision-based objects detection designs. The designs [30, 31] are dedicated on multiple kinds of objects detection, while the remainders focused on single kind of object, either pedestrian or motorcyclist detection. All of these existing designs are verified by using software models only and simulated on personal computers. One of the contributions of this work is to present the experience of implementing a vision-based multiple moving objects detection system on a portable platform. Matching models are required to enable machines to detect objects in images. These matching models can be approximately classified into two types, i.e., with either global features or local features. Features of the interested objects are extracted to train the classifier. Using global features of the objects is beneficial to achieving high detection rate, while using local features of the objects can solve the occlusion problems.

In [1], the key insight is that one may compute finely sampled feature pyramids at a fraction of the cost, without sacrificing performance, i.e., features computed at octave-spaced scale intervals are sufficient to approximate features on a finely sampled pyramid for a broad family of features. Extrapolation is inexpensive as compared to direct feature computation. The work [2] presented a spatialized random forest (SRF) approach, which can encode an unlimited length of high-order local spatial contexts. By spatially random neighbor selection and random histogram-bin partition during the tree construction, the SRF can explore much more complicated and informative local spatial patterns in a randomized manner. In [3], the authors had evaluated their system on a data set specifically for pedestrian detection from a moving vehicle, and they have shown that it is able to outperform other fast detection methods in both speed and accuracy. This is due to: (1) the use of a Coarse to Fine (CtF) procedure for fast image scan; (2) the use of object parts to simulate local deformations; (3) the evaluation of detections with missing resolutions; and (4) the introduction of an additional feature that balances out scores with missing resolutions and gives possibly high scores also to small detections, which are very important in the context of driving assistance. The work [4] proposed a decomposition-based human localization model dealing with this issue in three steps, i.e., a stable upper-body is firstly detected, then a set of bigger bounding boxes are extended, from which the most appropriate instance is distinguished by a discriminative Whole Person Model (WPM). The work [5] presented a method for characterizing tiny images of pedestrians in a surveillance scenario, specifically, for performing head orientation and body orientation estimation, employing arrays of covariance as descriptors, named Weighted ARray of COvariances (WARCO). The design [7] addressed the problem of ascertaining the existence of objects in an image. In the first step, the input image is partitioned into non-overlapping local patches, then the patches are categorized into two classes, namely natural and man-made objects to estimate object candidates. Then, a Bayesian methodology is employed to produce more reliable results by eliminating false positives. To boost the object patch detection performance, they exploit the difference between coarse and fine segmentation results. The design [8] proposed a representation for scenes containing relocatable objects that can cause partial occlusions of people in a camera's field of view. The authors formulated an occluder-centric representation, called a graphical model layer, where a person's motion in the ground plane is defined as a first-order Markov process on activity zones, while image evidence is aggregated in 2D observation regions that are depth-ordered with respect to the occlusion mask of the relocatable object. The work [9] improved on the successful Evolution COnstructed (ECO) features algorithm by employing speciation during evolution to create more diverse and effective ECO features. Speciation allows candidate solutions during evolution to compete within niches rather than against a large population.

The aforementioned literature provide solid foundation for researchers to develop their works. However, there is still a gap to be filled before one can achieve accurate moving objects detection for intelligent automobiles with real-time performance on portable platforms.

8.2.2 Lane Departure Warning System

In the basic system flow of LDWS, there are two main steps of detecting the lanes. One is lane-mark generation, and the other is lane model fitting.

At lane-mark generation stage, most papers, such as [32, 33], and [34, 35], used canny edge detector, which can keep a good performance even in low contrast weather conditions, to extract the lane-mark. However, the computing burden of canny edge detector is more than the one of brightness thresholding. With this reason, the paper [36] used brighter region to extract lane-mark based on Charge-Coupled Device (CCD) camera parameter regulation skill.

Straight line is the most famous and common lane model, which can be detected with Hough transform [33, 34, 36] or Weight Least Square Regression (WLDR) [37], because the lane-mark appears straight near the vehicles. In order to include curves, curve lane models are adopted in some papers. Lindner et al. [32] used the fitting value of Hamacher function to decide whether the line can be added or not. Yoo et al. [35] used the quadratic curve model to represent the lane-mark as shown in Eq. (8.1), where x and y are coordinate values, c_0 is the curvature of the lane, m is the slope, and b is the offset of the lane.

$$\mathrm{y} = \frac{1}{2}c_0x^2 + mx + b \tag{8.1}$$

8.2.3 Forward Collision Warning System

In basic system flow of FCWS, Sun et al. [38] claimed two steps for detecting vehicles, i.e., Hypothesis Generation (HG) and Hypothesis Verification (HV). Generally speaking, the computing time in HV is more than that in HG, so eliminating most of free-driving space in HG by apparent vehicle features is needed. Next, some famous methods of HG and HV are introduced in the following paragraphs.

For shadow feature in HG, Kumar [39] used edge and gray value to capture shadow features. To begin with, free-driving space is extracted by edge segmentation and then they compute the mean and standard deviation to calculate threshold of shadow. The intensity-value difference in the vertical direction is examined to see whether or not there exists a transition from brighter intensity to darker intensity. The candidates are revealed in HG by shadow features.

For symmetry feature in HG, Teoh et al. [40] used symmetry characteristic to generate vehicle candidates. Canny operator is adopted to find reliable edge to be the basis of symmetry calculation. They select a pair of proper width and height to calculate symmetry depending on different scanning lines.

For tail-light feature in HG, Fossati et al. [41] used color, shape, area, and position information as a criterion of pairing algorithm. With low-exposure camera, they can take advantage of accurate color information in Hue–Saturation–Value (HSV) space to decide which type the light-object belongs to. After pairing tail-light objects, they estimated the forward vehicles position by using a pinhole model.

For Support Vector Machine (SVM) in HV, Teoh et al. [40] used two-pattern classifier based on a linear SVM to differentiate between vehicles and non-vehicles. In fact, they extract Edge Orientation Histogram (EOH) to be their features of the classifier by quantizing the gradient of each pixel into eight bins. Khairdoost et al. [42] applied lots of methods to SVM in order to raise classifier accuracy. First, Pyramid Histogram of Oriented Gradient (PHOG) is adopted to produce more features for the classifier. Second, they used Principle Component Analysis (PCA) to eliminate redundant PHOG features. Third, genetic algorithm is utilized to find the weighting of PHOG-PCA features.

8.2.4 Speed Limit Detection System

A basic speed limit signs detection flow can be generally divided into three parts, which include speed limit signs detection, to firstly locate the potential candidates for speed limit signs, speed limit signs verification, to verify if the candidates from the previous stage are indeed speed limit signs, and speed limit signs recognition, to classify critical information from the speed limit signs [54, 55].

8.2.4.1 Speed Limit Signs Detection

The goal of this phase is to select the potential signs by locating where they appear. There are two major kinds of speed limit signs, which are circular and rectangular as shown in Fig. 8.1 over the world.

Radial symmetric transform [44, 45] is one of popular shape detection algorithms in the sign location process, which aims to detect the center of n-side regular polygons in gray-scale images through the radial symmetric feature.

8.2.4.2 Speed Limit Signs Verification

In this phase, the candidates from the previous stage are verified whether they are speed limit signs or not by checking their contents.

AdaBoost learning with Haar-like features, a machine learning architecture proposed by Viola–Jones, is adopted [46]. Cascaded geometric detectors are defined such as area, solidity which is the ratio between the number of ROI

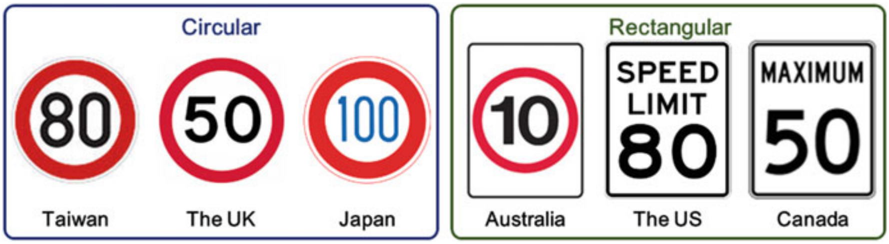


Fig. 8.1 Common types of speed limit signs over the world

background pixels and the total number of ROI pixels, vertices relative positions which detect rotated and non-rotated objects, and dimensional ratios which discard non-symmetric objects and maintain rectangular shape objects [47].

8.2.4.3 Speed Limit Signs Recognition

Here, the speed limit signs are classified to recognize the actual speed limit digits inside the signs. A binary classifier SVM is adopted by the tree structure with rotation invariant features, which is generated by Fourier transformed input image to classify speed limit signs [48]. Digits features are also considered one of efficient manners, e.g., blob, which is defined as a closed region, and breach, which is defined as the open region [49, 50].

8.2.5 Inclement Weather Processing Technology (DLCE)

Currently, there is no one-size-fits-all solution to inclement weathers. Each inclement weather is considered separately. Many de-fog technologies are proposed in recent years, which can be categorized into multi-frame-need/single-frame-need or image enhancement/physical model recovery. Some multi-frame-need methods [56, 57] generated a good result but those are not able to be adopted in dynamic scenes. Thus, single-frame-need methods were developed, with an image enhancement-based idea. Solving fog influence with single frame is accompanied with huge challenges. No matter using cost function and Markov Random Fields [58] or Independent Component Analysis (ICA) [59] has some disadvantages, i.e., unnatural image, and limited applied image. A Dark Channel Prior (DCP) [60, 61] was proposed for de-fogging foggy images in 2009. This assumption obtained wonderful de-fogging results but required high computational complexity and relied on precision of the dark channel computation, that is, the result may fail once the dark channel is wrong. Night is also one of most encountered inclement weathers. Histogram Equalization (HE) [62] is a common way used in image processing for inclement weathers, which is usually applied to image enhancement for single-camera high dynamic range (HDR) processing. However, HE could not reveal comprehensive details and did not cover the consideration for local conditions. Therefore, Adaptive Histogram Equalization (AHE) [63, 64] and Contrast Limited Adaptive Histogram Equalization (CLAHE) [65] were proposed to improve this weakness of HE. A better result is generated by adopting these methods. Since low contrast is a common phenomenon among different inclement weathers, these contrast enhancement algorithms are quite often to be adopted in this field.

8.3 Advanced ADAS System

Vision processing technology has a wide range of applications on vehicles (Fig. 8.2). Collaborated with front cameras, PMD, LDWS, FCWS, Stop-and-Go, Traffic Light Detection (TLD), and SLDS techniques can help drivers notice pedestrians, motorcyclists, vehicles, traffic lights, and speed limit signs in front of the way. With proper vehicle control intervention, FCWS and Stop-and-Go techniques can guarantee drivers further driving safety. Side cameras capture videos for BSD Systems (BSDS). Meanwhile, wide-view video stitching can combine several video sources captured from different cameras around the car and provide a panoramic view with viewing angle up to 360° for drivers. Furthermore, HDR technology helps drivers see clear in scenes with high variation in lighting, e.g., when going in/out tunnels and facing strong lights in opposite direction at night. Inside the cars, driver dangerous behavior detection system can help remind drivers to drive the cars properly. Besides, hand tracking technology help drivers control the in-car equipment by a more convenient and safer way, i.e., hand gesture.

In the following, we introduce more details on several significant intelligent vision processing technologies for ADAS including PMD, LDWS, FCWS, SLDS, and DLCE.

8.3.1 Pedestrian and Motorcyclist Detection System

Machine learning algorithms are widely used in pattern recognition such as face detection and license plate recognition. They own high flexibility and good accuracy in detecting the target objects for specific applications. The classifiers used in

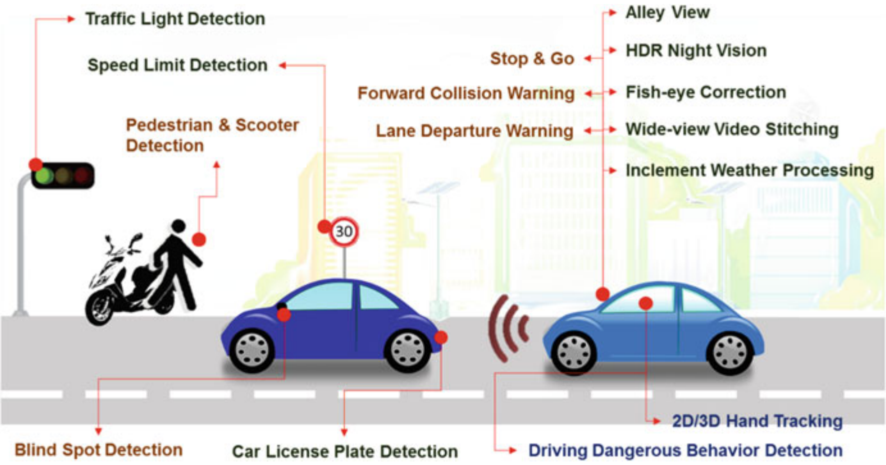

Traffic Light Detection
Speed Limit Detection
Pedestrian & Scooter Detection
30
Stop & Go
Forward Collision Warning
Lane Departure Warning
Alley View
HDR Night Vision
Fish-eye Correction
Wide-view Video Stitching
Inclement Weather Processing
Blind Spot Detection
Car License Plate Detection
2D/3D Hand Tracking
Driving Dangerous Behavior Detection

Fig. 8.2 The application scenario of our intelligent vision processing technology developed in NCTU and FCU, Taiwan

machine learning algorithms are specifically trained based on the collected samples for the target. However, they often generate false alarms as patterns similar to target objects appear in the background, which is also the major design challenge when applying machine learning algorithms on object detection. On the other hand, larger color contrast between the object and the background is helpful for detection. Hence, the positive samples used to train the classifier should be representative. To enhance the robustness of multiple moving object detection, we propose several rules in capturing training samples, and then employ AdaBoost algorithm to detect pedestrians, and people riding motorcycles commonly seen on the road.

Due to the great diversity of clothing and posture, pedestrian detection has always been a challenging task. Besides, information from the background may also influence the detection result. Therefore, we propose to choose pedestrian samples by following the rules below: each sample includes one pedestrian only, let the boundary of the sample close to the pedestrian, and include both samples which include global features and local features. In addition, the training sample selection method for motorcycle detection is somewhat different to that for pedestrian detection because motorcycles move much faster than pedestrian. According to the distance, we present sample selection ways for detecting farther motorcycles and nearer ones, respectively. Considering that farther motorcycles appear to be smaller than the nearer ones, we adopt samples with global features to enhance both the detection accuracy and detection distance. On the contrary, nearer motorcycles appear to be larger in size than the farther ones, we adopt samples with local features to decrease the false detection rate and avoid detection miss due to occlusion. Moreover, we adopted vehicle samples of local features for training the classifier to detect both nearer and farther vehicles because vehicles are rather larger than motorcycles in terms of size.

The performance of the classifier is not always enhanced as the number of samples is increased. The training phase for obtaining a discriminative objects classifier may be time consuming. Therefore, we propose a multi-pass self-correction procedure for effectively achieving the goal. First, we have a classifier trained by using conventional one-pass procedure. We then use this classifier to detect the internal samples from the adopted database and keep the correctly detected samples only for retraining the classifier. After that, we use the classifier to detect more samples beyond the adopted database and keep the correctly detected samples only when retraining the classifier. This method not only can avoid wrecking the feature of the basic classifier, but also can emphasize versatile features of different objects for improving detection performance. To improve the detection performance indicates both increasing the detection accuracy and lowering down the false detection rate. From the performance evaluation, we find that the proposed self-correction procedure not only can save 25 % effective sample selection time but also can lower down 50 % false detection rate due to the effective samples collected. The time saving is evaluated according to the working time saving of an operator who is familiar with the sample selection process including sample capturing and classifier training. More details of the above methods can be found in [30].

Furthermore, we adopt heterogeneous computing with OpenCL for realizing this object detection design on a multicore CPU and GPU platform. By utilizing the techniques of scale parallelizing, stage parallelizing, and dynamic stage scheduling on AdaBoost algorithm, windows load unbalance problem and scale load unbalance problem are solved. Consequently, the proposed object detection design achieves 32.5 frame per second (fps) at 720 × 480 (D1) resolution on an AMD A10-7850K processor.

8.3.2 Lane Departure Warning System and Forward Collision Warning System

Traffic accidents may cause great damage on people's lives and wealth. In Taiwan, there are 273,449 traffic accidents (i.e., the sum of A1 traffic accidents, which causes people die within 24 h, and A2 ones, which causes people injury or die beyond 24 h) in 2013. About 20.4 % of A1 traffic accidents happened due to fatigue driving or drivers' inattention. Therefore, it is inevitable to develop some driver assistance functions to help reminding the drivers to be aware of the dangerous driving conditions. Among the ADAS functions, LDWS and FCWS are two major technologies.

In real driving environments, there are lots of problems, which may cause ADAS functions to fail, such as inclement weathers and complicated scenes. To conquer the problem of inclement weather, dynamic threshold, which combines local threshold with global threshold and change the threshold automatically, is proposed to avoid illumination variation. To reduce the effect of different scenes, multiple frame approval, which accumulates the frequency of the desired objects based on the position of the frame in order to eliminate static objects and the false alarm caused by windshield wiper, is adopted when the vehicle speed is over 40 km/h. With the aforementioned solutions, we are able to overcome the design challenges of the two popular ADAS functions, i.e., LDWS and FCWS, to be applied in real world driving environments, which is one of the major contributions of this chapter.

Hence, we propose the design, verification, and vision radar system integration of two popular ADAS functions including LDWS and FCWS. A dynamic threshold method (including local threshold and global threshold) is adopted to improve the accuracy of lane detection and vehicle detection in various weather conditions. Multiple frame approval is adopted to conquer the effect of some tough scenes, such as static objects, signs appearing temporarily, or interference of the windshield wiper. The proposed system is implemented on automotive-grade Freescale i.MX6 (including 4-core ARM Cortex A9, 1 GB DDR3 RAM, and Linux environment) with a USB webcam to capture the video. Under the D1 resolution, the performance of the proposed LDWS achieves 33 fps, while the performance of the proposed FCWS achieves 32 fps, and the performance of integrated application achieves 22 fps.

8.3.3 Speed Limit Detection System

In recent years, car cam recorders have become more and more popular. Thanks to it, the footages of car accidents, dangerous driving, or other critical road-side events can be entirely recorded, which can help justify who is the perpetrator and be the evidence for illegal driving. Furthermore, the information of traffic signs is able to be extracted from footages. To realize multiple-country SLDS on embedded systems, we simplify the proposed algorithms computing resource with template databases and digital features.

Currently, color-based algorithms are adopted in most of the SLDSs along with training samples for machine learning algorithms. This approach needs to not merely adjust training samples based on different cameras, but also take long-period training time causing it less practical chance. A low complexity shape-based speed limit sign locating algorithm, adaptive threshold and multiple-country-suitable speed-limit-digit recognition algorithm are proposed. Our multiple-country SLDS maintains good detection rate under inclement weathers. The proposed algorithm reaches 150 fps on the Intel i7-2600 3.4 GHz CPU desktop with D1 resolution on desktop and 30 fps with D1 resolution on Freescale i.MX 6 platform.

8.3.4 Inclement Weather Processing Technology (DLCE)

Many key functions in ADAS were proposed and expected to execute with normal weather conditions in the recent years. Capturing a low contrast image and shooting a color-faded image might cause failure of these systems. The technologies proffered to deal with inclement weather [56–68] are framed narrowly, which are not one-size-fits-all. The contrast of vision decrease exists at nights, foggy days, cloudy days, and rainy days with our observations. Thus, exploiting the idea of AHE, we propose a so-called Dynamic Local Contrast Enhancement (DLCE) technique, which can strengthen the image quality in the most inclement weather conditions, improve unnatural over-enhancement image quality, and reduce noise existed in the image. DLCE technique is designed and implemented on an embedded platform to verify its correctness and robustness. Without specific hardware and software optimizations, the proposed DLCE system is realized on TREK-668 platform with ATOM 1.6 GHz in real-time for both requirements of 120 fps 352×288 (CIF) resolution and 50 fps D1 resolution.

8.4 Implementation Issues

Both detection performance and real-time implementation on embedded system are our important targets. Through using OpenCL programming language, a heterogeneous system architecture can speed up the processing performance of PMD. Those who tend to detect targets with unique appearances, i.e., LDWS, FCWS and SLDS, can well recognize the targets by analyzing specific features, e.g., shadows, tail lights, and shape. Furthermore, these detection systems keep good precision rate in various weather by adopting the extended idea of DLCE. The implementation of proposed PMD, LDWS, FCWS, SLDS, and DLCE is illustrated in this section as follows.

8.4.1 Pedestrian and Motorcyclist Detection System

AdaBoost algorithm has been widely used in face recognition. Due to its simplicity and regularity, it has also been adopted to detect other targets such as pedestrians, motorcyclists, and vehicles. Haar-like features are usually utilized alone with AdaBoost algorithm. Besides, developers find that a skill called integral imaging can reduce the redundant computation of Haar-like features and therefore can accelerate the detection speed. Moreover, AdaBoost classifier contains multiple weak classifiers in a cascade manner. Only when a candidate passes all weak classifiers, it is recognized as a targeted object.

Before a superior AdaBoost classifier can be obtained, one should collect enough positive samples and negative samples and train the classifier. The selection of training samples affects the detection performance of the resulting classifier obviously. By adopting the sample selection rules and multi-pass self-correction procedure presented in Sect. 8.3.1, we can train a superior AdaBoost classifier efficiently. After that, we can utilize OpenCV to implement the AdaBoost classifier for PMD.

Although OpenCV contains many useful subroutines, the version performs only 4.76 fps on the Panda board for 320×240 (QVGA) video format. On the other hand, we have also implemented the proposed design as a C model to improve the real-time performance. The C version can achieve 30.3 fps for QVGA video format on the Panda board, i.e., 6.37 times of performance speedup is obtained. Our method achieves a detection rate of 91.8 % with only 3.3 % false alarm rate for multiple objects detection. The detection performance of our method is obviously better than that of existing methods.

In addition, we also adopt heterogeneous computing with OpenCL for realizing this PMD design on a multicore CPU and GPU platform. Figure 8.3 illustrates the flow chart of the proposed heterogeneous computing system for PMD. Scaling up of detecting windows for different sizes of pedestrians is required. The big scaling factors have less number of windows than small scaling factors have. Besides, the amount of windows for detecting near objects is much less than that for detecting far objects. Meanwhile, using GPU to process small amount of windows is inefficiency due to the induced memory latency. Hence, we propose to execute the far-distance PM detection in parallel on GPU and finish the near-distance PM detection in parallel on CPU to maximize the resource utilization of both CPU and GPU cores. Besides, the non-PM windows are rejected at the earlier stages in

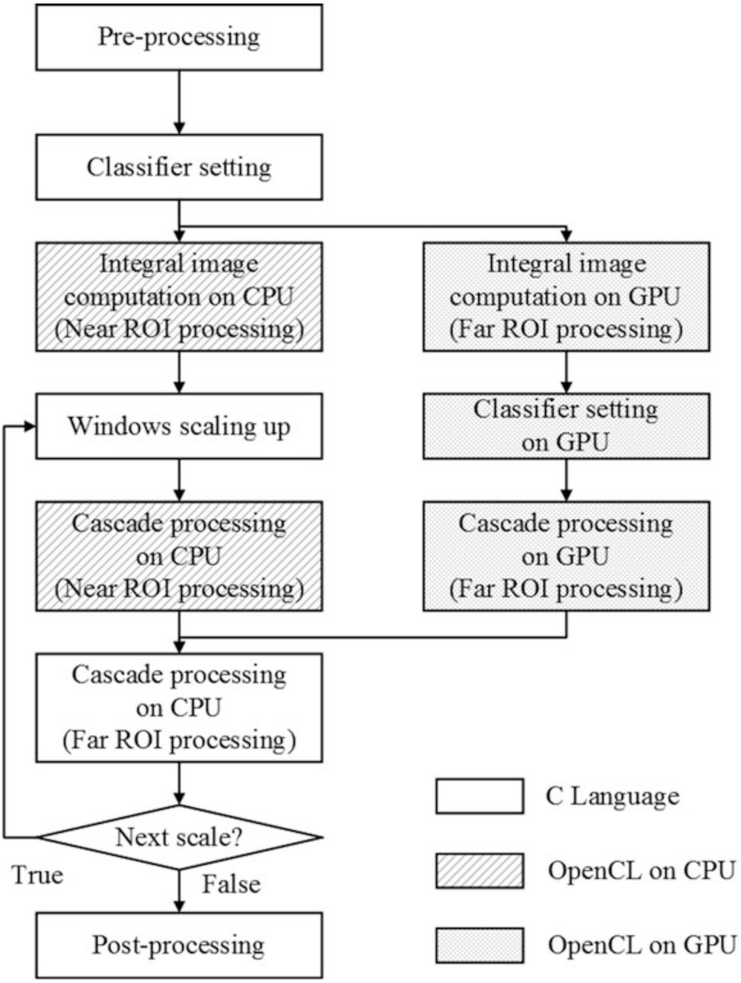

Pre-processing
Classifier setting
Integral image computation on CPU (Near ROI processing)
Integral image computation on GPU (Far ROI processing)
Windows scaling up
Classifier setting on GPU
Cascade processing on CPU (Near ROI processing)
Cascade processing on GPU (Far ROI processing)
Cascade processing on CPU (Far ROI processing)
Next scale?
True
False
Post-processing
C Language
OpenCL on CPU
OpenCL on GPU

Fig. 8.3 The flowchart of proposed heterogeneous computing system for Pedestrian and Motorcyclist Detection System (PMD)

the cascade process of AdaBoost algorithm, and only few windows go through all the stages. Over 95 % of candidate windows are dropped after the first five stages and only 0.03 % of candidate windows finish the whole cascade process. Because the parallelism of the rest candidate windows is not suitable for GPU parallel processing, GPU passes the rest of candidate windows to CPU to complete the whole cascade process in the proposed system. Meanwhile, GPU can process the next scale of pedestrian detection. Moreover, a dynamic stage scheduling is proposed to keep both the CPU and GPU busy in different situations. Once the processing time on GPU is less than that on CPU, more stages then are allocated for GPU in the next scale computation and vice versa. By utilizing these techniques of scale parallelizing, stage parallelizing, and dynamic stage scheduling on AdaBoost algorithm, windows load unbalance problem and scale load unbalance problem are solved. Consequently, the proposed PMD design can achieve 32.5 fps at D1 resolution on an AMD A10-7850K processor.

8.4.2 Lane Departure Warning System

As shown in Fig. 8.4, Conditional Dynamic Threshold (CDT) is adopted to extract brighter region as lane-marks. Line-thinning speeds up the Hough transform procedure, then line collection reduces lots of lines not belonging to lane-marks. Our algorithm can solve the problems of detecting outer lanes and avoiding the effect of windshield wiper by the occurrence frequency in finite state machine.

CDT is applied at the first stage, that is, more conditions are added into the equation to conquer the effect of the different weathers, such as day, nightfall, or night. The weather condition is based on the sky region as shown in Fig. 8.5.

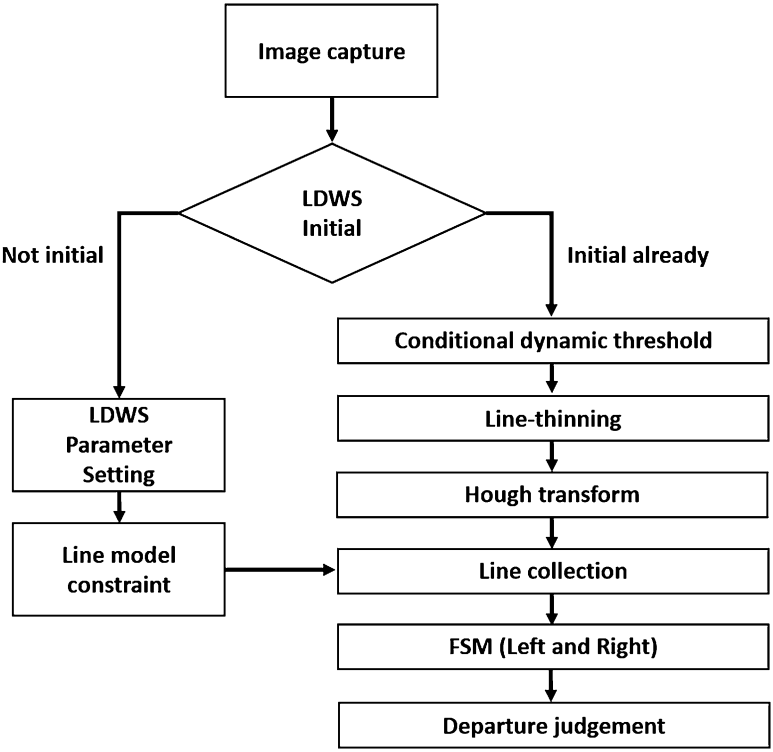


Fig. 8.4 The flowchart of proposed Lane Departure Warning System (LDWS)

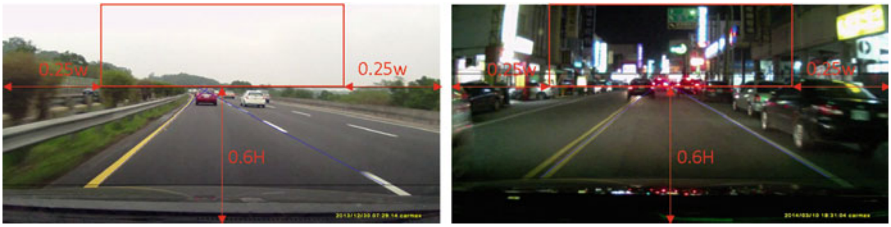


Fig. 8.5 Sky region of judging weather condition

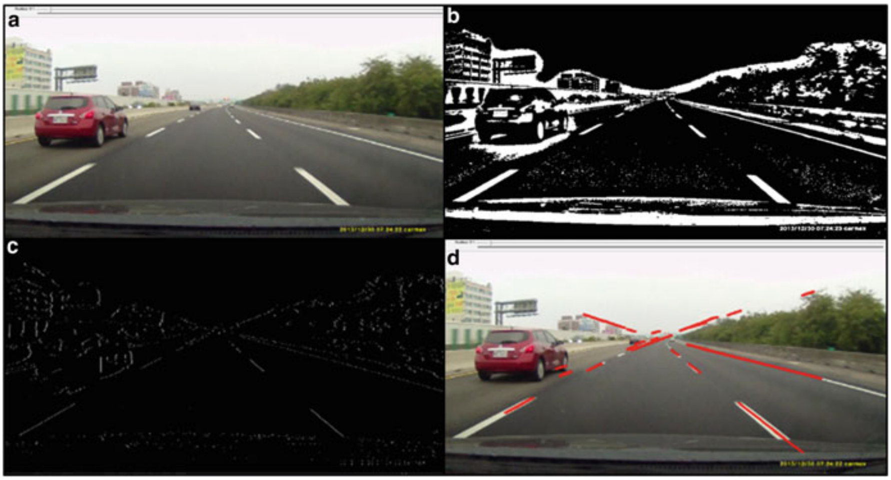


Fig. 8.6 Line detection result images: (**a**) original image, (**b**) after CDT, (**c**) after line-thinning, and (**d**) after Hough transform

After thresholding of CDT, line-thinning is executed to speed up Hough transform, as shown in Fig. 8.6.

The line model is composed of the middle point of the vehicle in x coordinate, and the vanishing point in y coordinate. With our observation, we find that the distance from the point to the line, belonging to the lane, will be always less than the threshold, 50 pixels in D1 resolution, and the constraint is shown in Eq. (8.2).

$$\begin{aligned} &p(x_0, y_0) \ \ and \ \ L : ax + by + c = 0 \\ &D(p, L) = \frac{|a \cdot x_0 + b \cdot y_0 + c|}{\sqrt{a^2 + b^2}} < 50 \end{aligned} \tag{8.2}$$

This system is integrated with the FCWS in order to enhance driving safety, whose results are shown at the end of Sect. 8.4.3.

8.4.3 Forward Collision Warning System

As shown in Fig. 8.7, weather judgment is applied and benefits dynamic threshold decision to extract brighter region as tail-light features and darker region as shadow features. We use vertical and horizontal edge to verify features in spatial domain. The occurrence frequency reduces the false detections caused by different scenes or windshield wiper in temporal domain since the changes in each frame of different scenes and windshield wiper are stronger than the one of lane is.

Vehicle model based on certain position takes advantage of judging whether the vehicle size is correct or not. The proposed vehicle model expresses the situation, that is, the farther the vehicle is, the smaller it will be, and vice versa, as shown in Eq. (8.5), which is deduced from pinhole camera extended equation, Eq. (8.3), and an equation from [43], i.e., Eq. (8.4), by eliminating variable D. For parameter meaning, F_c is camera focal, and y_h is the vanishing point. We assume that the

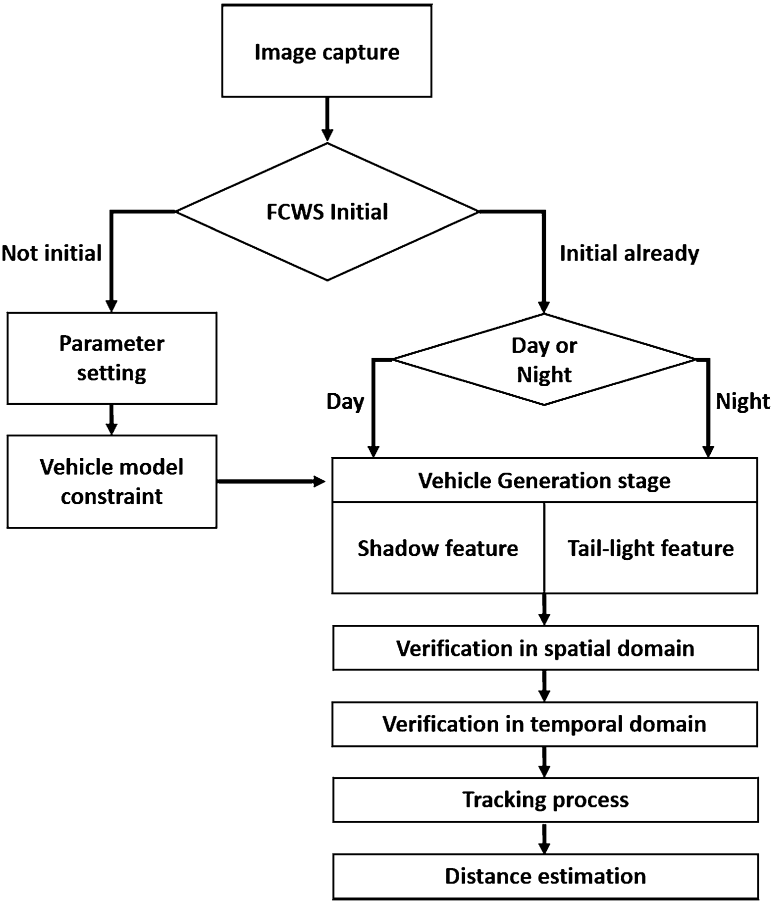


Fig. 8.7 The flowchart of proposed Forward Collision Warning System (FCWS)

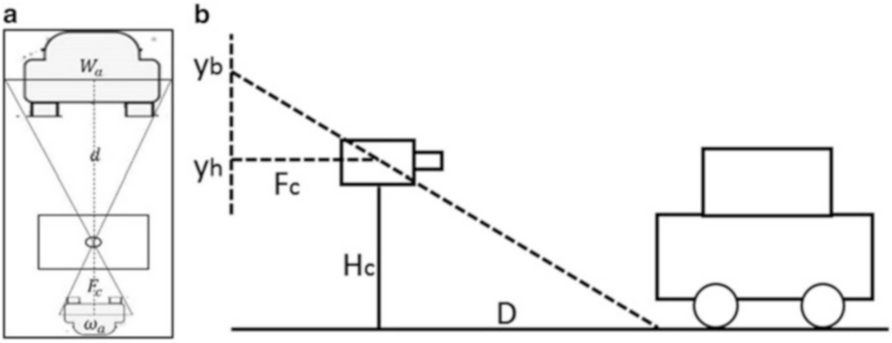


Fig. 8.8 Distance estimation: (**a**) pinhole model and (**b**) equation from [43]

camera height (H_c) is 1.5 m, and the average width of vehicles (W_a) is 2 m. The bottom width of the vehicle must be in the range of generated vehicle model on the corresponding image y-axis as demonstrated in Fig. 8.8.

$$\mathrm{D} = \mathrm{F_c} \cdot \frac{W_a}{\omega_a} \tag{8.3}$$

$$\mathrm{D} = \frac{\mathrm{F_c} \cdot H_c}{y_b - y_h} \tag{8.4}$$

$$\omega_a = (y_b - y_h)^* \frac{W_a}{H_c} \rightarrow \omega_a = (y_b - y_h)^* 1.3 \tag{8.5}$$

Light objects can be obtained by Connecting Component Labels (CCL) in the image after dynamic threshold. Rule-pairing is to find tail-light features with four rules, i.e., horizontal position, area, motion, and vehicle model. If two of the light objects satisfy the four rules, the tail-light feature is formed.

Four 5-min videos are picked up to show our integrated achievements (i.e., LDWS and FCWS) in this chapter. Forward vehicles within the range between 5 and 40 m at day, and between 5 and 30 m at night, and lateral vehicles within the range between 10 and 30 m at both day and night should be detected in our specification. We sample each frame per second in order to reduce the counting process of detection rate and false alarm, defined in Eq. (8.7). As shown in Table 8.1, we can achieve 90.15 % detection rate and 5.96 % false alarm rate averagely, and each result of the video sequence would be displayed in Fig. 8.9.

$$\begin{aligned} &\begin{cases} vehicles(real\ \ answer) \\ non-vehicles(real\ \ answer) \end{cases} \begin{matrix} \begin{cases} vehicles(our\ \ judgement) & a \\ non-vehicles(our\ \ judgement) & b \end{cases} \\ \begin{cases} vehicles(our\ \ judgement) & c \\ non-vehicles(our\ \ judgement) & d \end{cases} \end{matrix} \\ &Detection\ \ rate = \frac{a}{a+b},\quad False\ \ alarm\ \ rate = \frac{c}{a+c} \end{aligned} \tag{8.6}$$

Table 8.1 The integrated system experiment results

No	Weather	Scene	Detection rate	False alarm
1	Day	Highway	99.62 % (0265/0266)	001.48 % (004/0269)
2	Day	Highway	96.88 % (0249/0257)	002.35 % (006/0255)
3	Day	City	98.49 % (0196/0199)	009.25 % (020/0216)
4	Night	Highway	76.07 % (0248/0326)	006.06 % (016/0264)
5	Night	City	80.85 % (0114/0141)	016.17 % (022/0136)
Day			98.33 % (0710/0722)	004.05 % (030/0400)
Night			77.51 % (0362/0467)	005.26 % (038/0722)
Average			90.15 % (1072/1189)	005.96 % (068/1140)

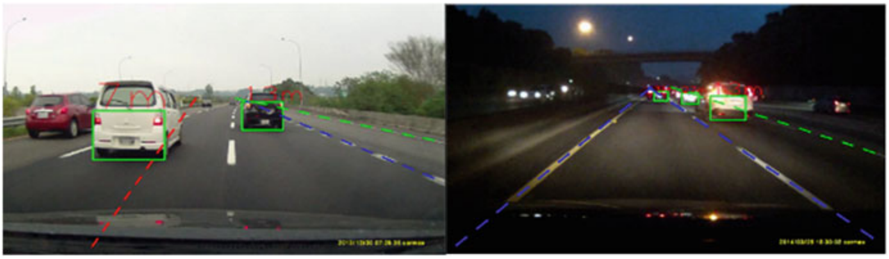


Fig. 8.9 The integrated system experiment results

On Freescale i.MX6 with Logitech C920 webcam input, LDWS achieves 33 fps at D1 resolution, FCWS achieves 32 fps at D1 resolution, and the integrated system reaches 22 fps at D1 resolution.

8.4.4 Speed Limit Detection System

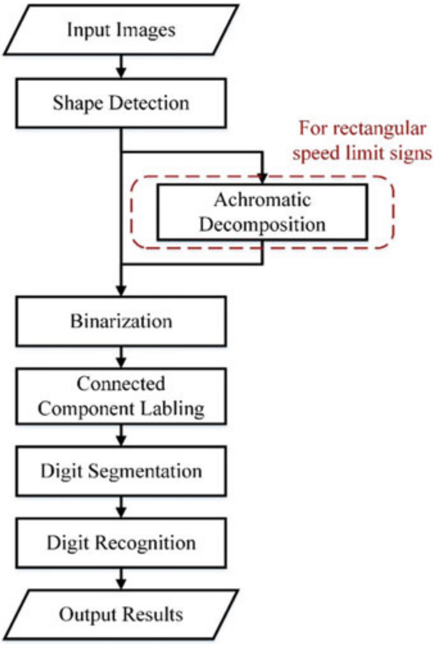
8.4.4.1 Shape Detection

Figure 8.10 shows the flowchart of proposed SLDS, which starts from shape detection after getting image. The voting process is based on the gradient of each pixel. The direction of gradient can form a vote. The vote generated from each pixel follows the symmetric axes, which cause the highest in the center of the shapes.

Sobel operator is used to generate horizontal and vertical gradients, where each selected pixel is represented with its absolute magnitude, and the gradient vector is denoted as **g**(p). The direction of **g**(p) can be formulated with the horizontal gradient G_x and the vertical gradient G_y into an angle as shown in Eq. (8.7).

$$\mathbf{g}(\mathrm{p}) = \tan^{-1}\frac{G_y}{G_x} \tag{8.7}$$

Fig. 8.10 Proposed speed limit detection algorithm



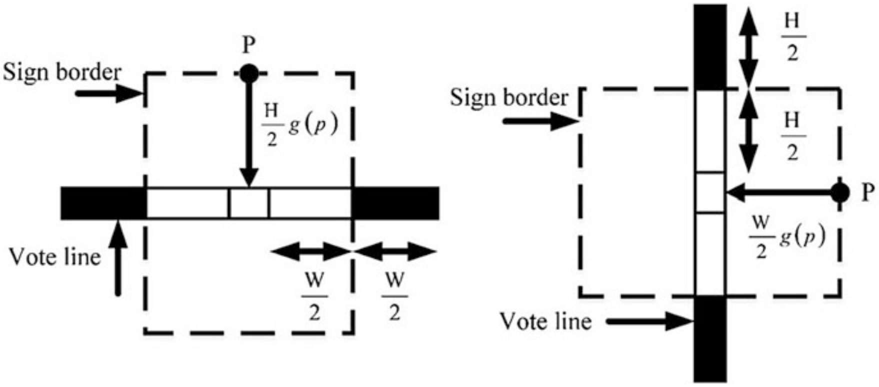
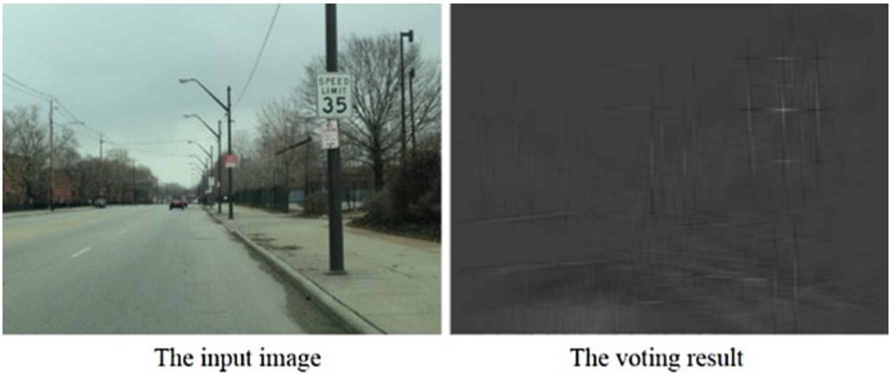


Fig. 8.11 The voting process for both horizontal and vertical vote

The voting image is firstly initialized to zero, and accumulates vote-number by horizontal and vertical voting line, which is generated by each pixel with positive and negative votes as illustrated in Fig. 8.11.

The centers of sign candidate will receive higher vote-number as shown in Fig. 8.12, and then several sign candidates are picked up with designed constraints for rest judging processes.



The input image

The voting result

Fig. 8.12 The result after the voting process

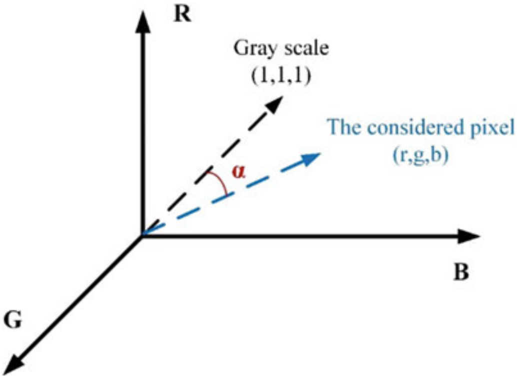


Fig. 8.13 The schematic of the RGB model and the angle α

8.4.4.2 Achromatic Decomposition

We use the inner product between (1,1,1), where gray scale is along, and each considered pixel to check the angle α between these two vectors to apply the decomposition in RGB domain as illustrated in Fig. 8.13, where each considered pixel in vector form of (r,g,b). The cosine function of α, which is equal to the inner product is shown in Eq. (8.8).

$$\cos\alpha = \frac{(1,1,1)\cdot(r,g,b)}{|(1,1,1)|\times|(r,g,b)|} = \frac{r+g+b}{\sqrt{3}\times\sqrt{r^2+g^2+b^2}} \tag{8.8}$$

8.4.4.3 Binarization and Digit Segmentation

Otsu threshold is adopted in days and adaptive threshold is adopted in nights. Using the fact that the speed limit of rectangular speed limit signs is two-digit, the pairing rules sizes and positions are proposed as below:

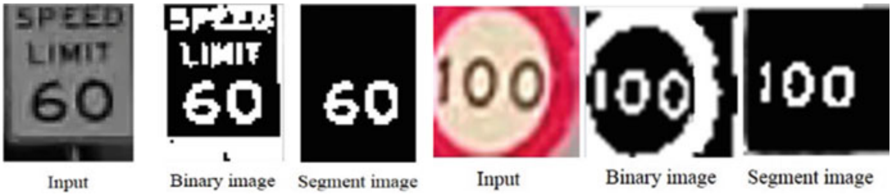
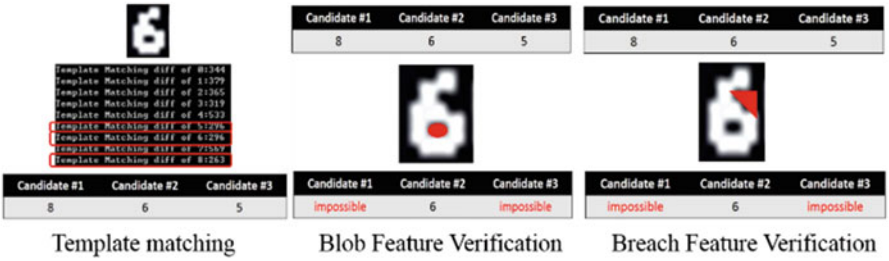


Fig. 8.14 The example of digit segmentation results



```
Template Matching diff of 0:344
Template Matching diff of 1:379
Template Matching diff of 2:365
Template Matching diff of 3:319
Template Matching diff of 4:533
Template Matching diff of 5:296
Template Matching diff of 6:296
Template Matching diff of 7:569
Template Matching diff of 8:263
```

Candidate #1 Candidate #2 Candidate #3
8 6 5
Template matching
Candidate #1 Candidate #2 Candidate #3
8 6 5
Candidate #1 Candidate #2 Candidate #3
impossible 6 impossible
Blob Feature Verification
Candidate #1 Candidate #2 Candidate #3
8 6 5
Candidate #1 Candidate #2 Candidate #3
impossible 6 impossible
Breach Feature Verification

Fig. 8.15 The example of digit recognition results

1. The areas of the digit candidates should be similar;
2. The positions of the digit candidates should be close enough;
3. The density of the pixels inside the digit candidates should be similar.

The pairing steps of circular speed limit signs are similar but looser to one of the rectangular speed limit signs because two-digit and three-digit speed limits are included in circular speed limit signs (Fig. 8.14).

8.4.4.4 Digit Recognition

The extracted digits are firstly classified with the built-in different-font templates and selected out three possible numbers. After it, the blob, the closed region inside the digit, and breach features are applied to verify the final number.

After calculating the Sum of Absolute Difference (SAD) between the target digit candidate and the built-in templates, we select three possible digit numbers with less matching difference. With possible digit candidates, we adopt union row, which gathers several rows as a union row, to detect the blob feature, and then verify the digit. The pixel value of a union row is the union value of all rows in the union row. Then, for each union row, we count the number of lines in white pixels. A blob is formed only if the number of lines is the sequence of "1, 2, . . ., 2, 1". Similarly, we count the number of pixels where the white pixel first appears from both the right and the left in each column to half of width of digit candidates to detect breach, where there are a series number of pixels that are larger than half of digit height (Fig. 8.15).

Table 8.2 The accuracy for two different major types of signs

	Rectangular	Circular
Total video frames count	3482	2832
Total speed limit sign count	24	25
Detected signs	23	24
Detection accuracy (%)	95.83	96.00
Total detected signs frames count	81	80
Total detected signs and correctly classified frames	78	77
Recognition accuracy (%)	96.30	96.25
Overall accuracy (*detection accuracy* * *recognition accuracy*) (%)	92.28	92.40

Table 8.3 The details of rectangular speed limit detection

Video sequence number	1	2	3	4	5	6	7
Weather	Day	Day	Day	Night	Night	Rain	Rain
Number of signs	4	4	3	4	2	2	3
Detected signs	4	4	3	3	2	2	3
Missed signs	0	0	0	1	0	0	0
Number of frames with sign detection	15	17	10	9	7	9	12
Number of correct speed limit recognition	15	16	10	8	6	9	12
Number of wrong speed limit recognition	0	1	0	1	1	0	0

Table 8.4 The details of circular speed limit detection

Video sequence number	1	2	3	4	5	6	7
Weather	Day	Day	Day	Day	Night	Night	Rain
Number of signs	4	6	5	5	2	2	1
Detected signs	4	6	5	5	1	2	1
Missed signs	0	0	0	0	1	0	0
Number of frames with sign detection	12	20	15	16	4	9	4
Number of correct speed limit recognition	12	19	15	15	4	9	3
Number of wrong speed limit recognition	0	1	0	1	0	0	1

Freescale i.MX6 is chosen as the target embedded platform and the proposed algorithm is first designed in C++ with Visual Studio platform on an Intel i7-2600 3.40 GHz CPU desktop running Windows 7 with 8 GB memory. It reaches 150 fps at D1 resolution in average for both rectangular and circular speed limit signs on desktop and 30 fps at D1 resolution on Freescale i.MX6. The accuracy for two different major types of signs is listed in Table 8.2. Tables 8.3 and 8.4 also show the accuracy under different weather conditions for different types of signs.

As listed in Table 8.5, the proposed system provides high accuracy and efficient performance in applications of speed limit detection. It can reach real-time implementation on embedded systems with low computing resource, support different types of multiple-country speed limit signs, and support different digit fonts thanks to adopting the blob and breach features. Figure 8.16 shows SLDS results in various weather conditions and countries.

Table 8.5 The comparisons among other works and our system

	[51]	[52]	[46]	[53]	Our system	
CPU	2.13 GHz dual-core laptop	1.67 GHz Intel Atom 230 and a NVIDA GeForce 9400M GSGPU	2.16 GHz dual-core laptop	2.13 GHz dual-core laptop	Intel® Core™ i7-2600 CPU 3.4 GHz	Freescale i. MX6 with 4-core 1 GHz Cortex-A9 CPU
Accuracy (%)	90	88	96.25	90.9	92.3	
Video resolution	640 × 480	640 × 480	700 × 400	Image only	720 × 480	
Frame rate on PC (fps)	20	33	25	7.7 (130 ms)	150	30
Real-time on embedded system	X	O	X	X	O	
Supporting all types of speed limit sign	O	X	X	X	O	
Supporting different fonts of speed limit signs	X	X	X	X	O	

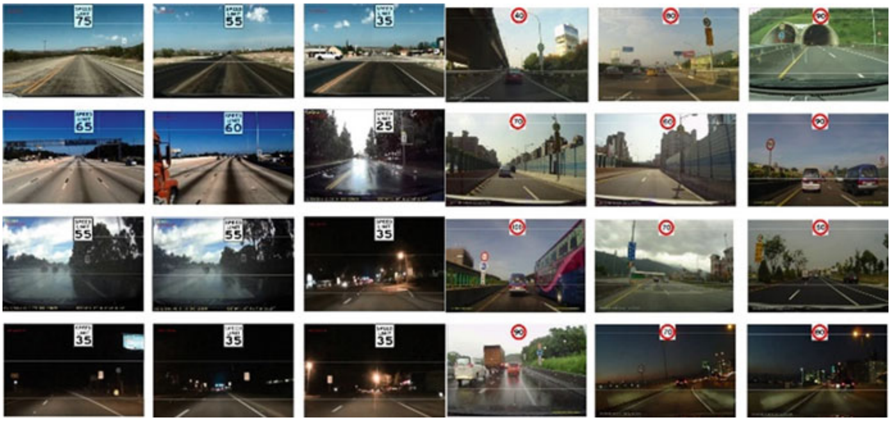


Fig. 8.16 The overall results for speed limit signs detection

8.4.5 *Inclement Weather Processing Technology (DLCE)*

In the proposed DLCE system, a contrast of a pre-defined block in the image is calculated and is used to limit the height of the histogram dynamically. Figure 8.17 shows the flowchart of the proposed system.

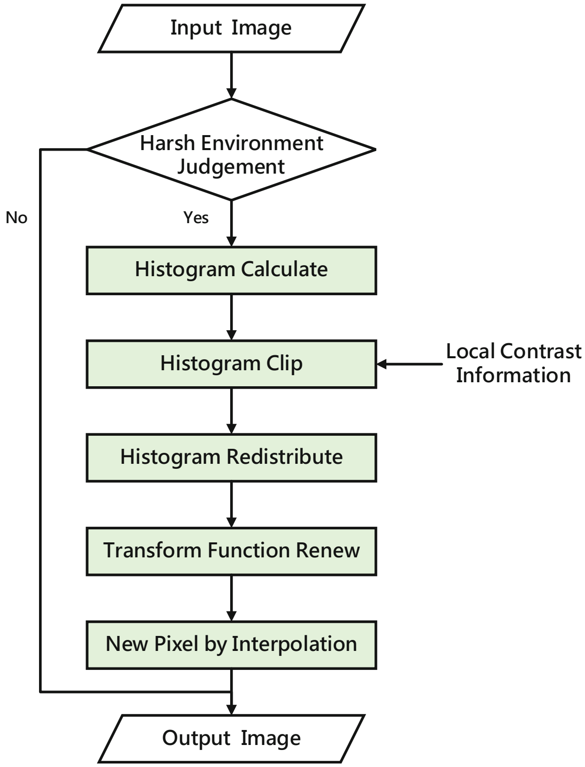


Fig. 8.17 Flowchart of the proposed Dynamic Local Contrast Enhancement (DLCE) method

Two main goals at harsh environment judgment phase are: (1) distinguishing harsh environment, and (2) obtaining the local contract information, as shown in Fig. 8.18. The luminance difference of pixel D(x,y) is calculated by the following equation:

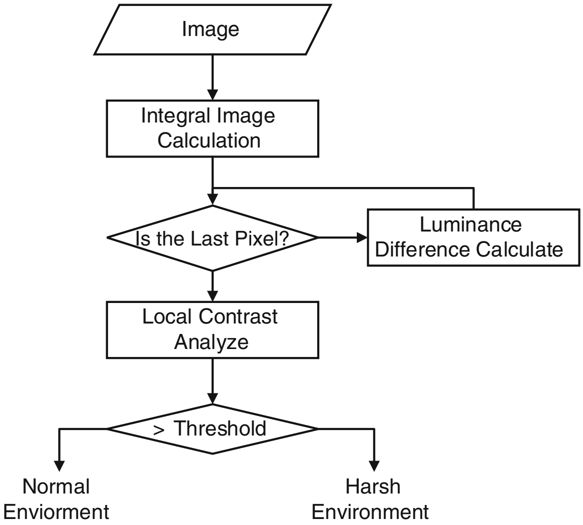
$$\mathrm{D}(x,y) = |f(x,y) - A(x,y)| \tag{8.9}$$

where f(x,y) is luminance of pixel (x,y) and A(x,y) is the average of luminance in block W × W. Eventually, the contrast with image size M × N is generated as Eq. (8.10).

$$\text{Contrast} = \frac{\sum_{x=0}^{M}\sum_{y=0}^{N} D(x,y)}{M \times N} \tag{8.10}$$

The luminance difference of pixel determines the value at which the histogram is clipped according to the following equations:

Fig. 8.18 Flowchart of the harsh environment detection



$$\mathrm{clip} = (1-\beta)*H_{avg} + \alpha*H_{avg} \tag{8.11}$$

$$\beta = \left(\frac{\sum_{x=0}^{M}\sum_{y=0}^{N} D(x,y)}{M \times N}\right)/255 \tag{8.12}$$

β, H_{avg}, and α represent dynamic control parameter, average of histogram, and intensity parameter to bridle image adjustment, respectively. Mapping function Re (x) updates depend on β after clipped pixel is distributed, which benefits these techniques to fit the current scene. (Note: CDF is cumulative distribution function.)

$$\mathrm{Re}(x) = \left(H'_{max} - H'_{min}\right) \times CDF(x) + H'_{min} \tag{8.13}$$

$$H'_{max} = H_{max} - \beta*(H_{max} - H_{min}) \tag{8.14}$$

$$H'_{min} = H_{min} + \beta*(H_{max} - H_{min}) \tag{8.15}$$

The proposed method has been implemented on TREK-668 embedded platform with 1.6 GHz Intel Atom N2600 CPU and tested under various inclement weather conditions. The performance of the prototype is able to achieve 50 fps at D1 video input.

Figures 8.19 and 8.20 present the quality of DLCE and Table 8.6 proves the statistics compared with Global Histogram Equalization (GHE), CLAHE [65], and DCP [60]. The average of higher contrast values, calculated by Eq. (8.10), as in Table 8.6 proves that DLCE method provides videos with clearer scene. Besides, local contrast is also improved when adopting the proposed method.

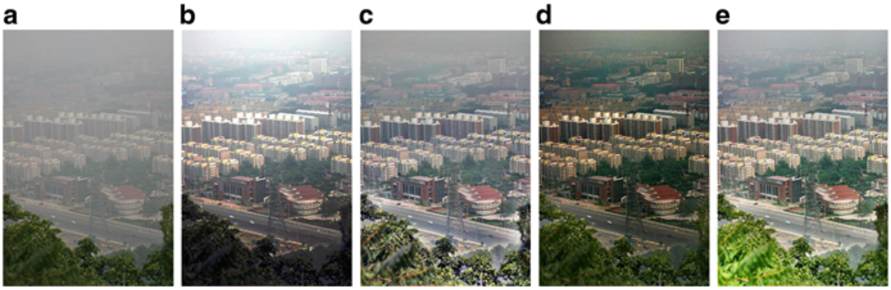


Fig. 8.19 Foggy day results with different methods: (**a**) original, (**b**) Global Histogram Equalization (GHE), (**c**) Contrast Limited Adaptive Histogram Equalization (CLAHE), (**d**) Dark Channel Prior (DCP), and (**e**) DLCE

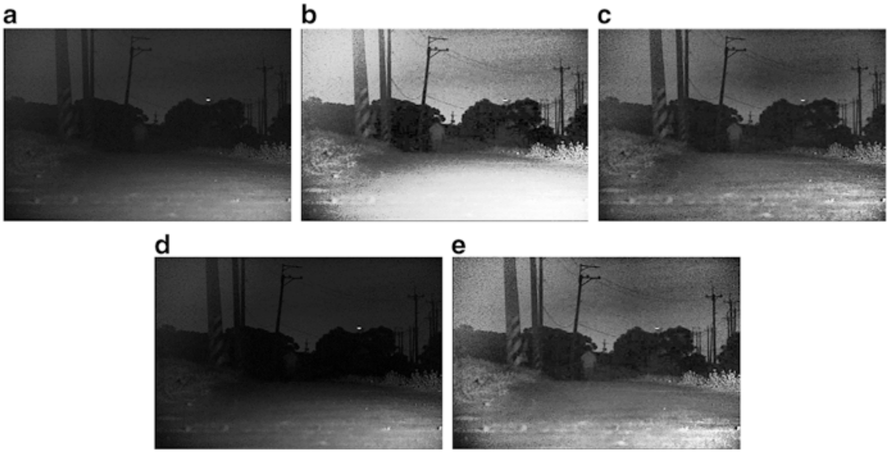


Fig. 8.20 Night results with different methods: (**a**) original, (**b**) GHE, (**c**) CLAHE, (**d**) DCP, and (**e**) DLCE

Table 8.6 Contrast values with various methods

	Original	GHE	CLAHE [65]	DCP [60]	DLCE
Fog day	9	18	25	19	24
Night	4	23	15	6	17

DLCE benefits ADAS in gaining better image quality at inclement weather conditions. More pedestrians are detected, better lane detection is generated, and farther scene can be seen while DLCE is applied, as demonstrated in Fig. 8.21 and Table 8.7.

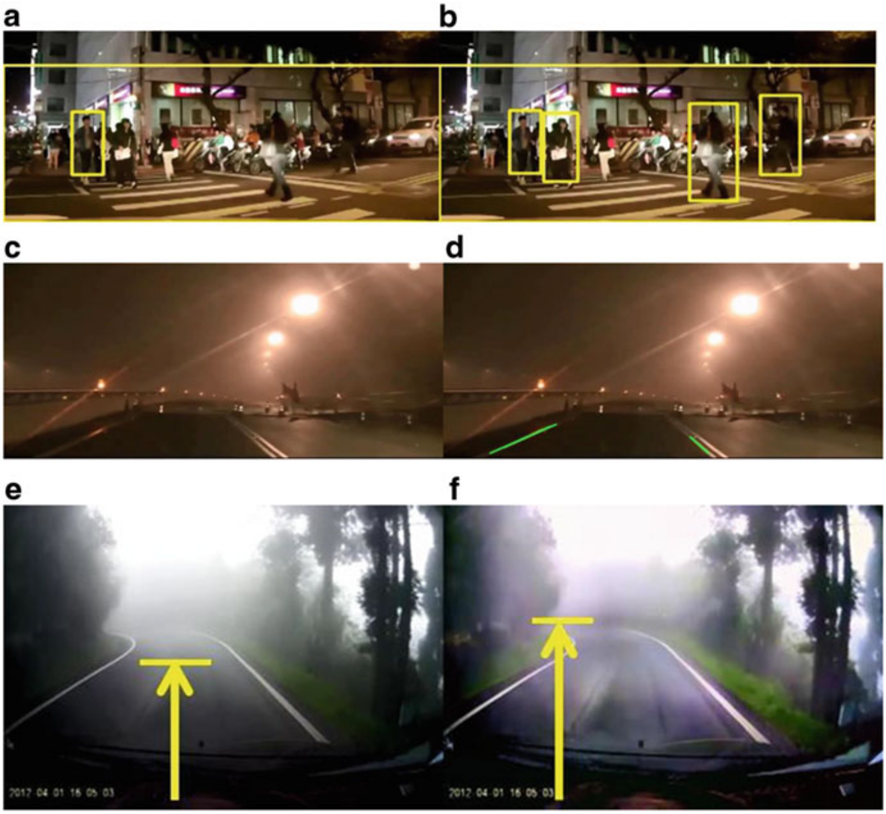


Fig. 8.21 ADAS results with adopting DLCE (**a**–**b**) PD, (**c**–**d**) LDWS, (**e**–**f**) car recorder, and (**b**, **d**, **f**) DLCE is adopted

Table 8.7 Increasing rate after adopting DLCE

	Number of detected lane LDWS	Number of detected lane LDWS + DLCE	Increasing rate of detected lane (%)
Rainy day	533	834	56.47
Foggy day	1100	1710	55.45
Night	912	1098	20.39

8.5 Conclusion

ADAS becomes quite important in these years for smart vehicle development. Moreover, having an autonomous car is not a far dream human beings immerse in. Vision-based object detection is an intuitive detection method similar to human visual perception, which is much low-cost compared with other detection methods such as RADAR or LiDAR. However, current vision-based objects detection methods still suffer from several challenges such as high false alarm rate and unstable detection rate which limit their value in practical applications. In addition, luminance variation and weather change induce tougher challenges to vision-based object detection. Therefore, more research and development effort are necessary in this area.

In this chapter, we have introduced a set of rules in selecting proper training samples and presented a multi-pass self-correction training procedure to achieve an effective multiple moving objects detection system with high accuracy and low false alarm rates. Besides, FCWS and LDWS are proposed and implemented on Freescale i.MX6 with a monocular camera, which provides safety information to drivers. The dynamic threshold method conquers the problems resulted from different weather conditions, and the multiple frame approval reduces the effect of different scenes and windshield wiper. Moreover, the proposed LDWS and FCWS algorithms are integrated together in order to produce comprehensive information for day and night highway driving, which detects not only the forward vehicle, but also the potential cut-in vehicles. The implementation shows that the integrated system can achieve real-time processing for video input with D1 resolution. Furthermore, speed limit signs, which are significant traffic signs beside the road, are regarded as the primary targets to recognize. A detection system is designed to not only work under different weather conditions, but also achieve real-time processing performance based on single webcam. With more and more demands on the driving safety, future works can focus on using the blob and breach features on recognition of licenses plates and supporting other traffic signs based on the shape detection which can make good use of videos from car cam recorders. In addition, DLCE method can help ADAS system obtain better results at inclement weather conditions. The significant improvements of DLCE are proved in terms of detection accuracy of LDWS and pedestrian detection system. The proposed method has been implemented on TREK-668 embedded platform with 1.6 GHz Intel Atom N2600 CPU and tested with many inclement weather conditions. The performance of the prototype is able to achieve 50 fps at D1 video input.

Currently, these works can extend the functions of car cam recorders, making it actively ensure the safety of drivers. Although there are lots of challenges ahead, the dream of autonomous car will eventually come true by continuous efforts on enhancing existing functions and enlarging the ability of detecting object.

References

Pedestrian, Motorcyclist, and Vehicle Detection System (PMD)

1. Dollar P, Appel R, Belongie S, Perona P (2014) Fast feature pyramids for object detection. IEEE Trans Pattern Anal Mach Intell 36(8):1532–1545
2. Ni B, Yan S, Wang M, Kassim AA, Qi T (2013) High-order local spatial context modeling by spatialized random forest. IEEE Trans Image Process 22(2):739–751

3. Pedersoli M, Gonzalez J, Hu X, Roca X (2014) Toward real-time pedestrian detection based on a deformable template model. IEEE Trans Intell Transp Syst 15(1):355–364
4. Bing S, Su S, Li S, Yun C, Ji R (2013) Decomposed human localization in personal photo albums. Proc. visual communications and image processing (VCIP), p 1–6
5. Tosato D, Spera M, Cristani M, Murino V (2013) Characterizing humans on Riemannian manifolds. IEEE Trans Pattern Anal Mach Intell 35(8):1972–1984
6. Yang Y, Ramanan D (2013) Articulated human detection with flexible mixtures of parts. IEEE Trans Pattern Anal Mach Intell 35(12):2878–2890
7. Choi J, Jung C, Lee J, Kim C (2014) Determining the existence of objects in an image and its application to image thumbnailing. IEEE Trans Signal Process Lett 21(8):957–961
8. Ablavsky V, Sclaroff S (2011) Layered graphical models for tracking partially occluded objects. IEEE Trans Pattern Anal Mach Intell 33(9):1758–1775
9. Lillywhite K, Lee D-J, Tippetts B (2012) Improving evolution-constructed features using speciation for general object detection. Proc. IEEE workshop applications of computer vision (WACV), p 441–446.
10. Satpathy A, Jiang X, Eng H-L (2014) Human detection by quadratic classification on subspace of extended histogram of gradients. IEEE Trans Image Process 23(1):287–297
11. Farhadi M, Motamedi SA, Sharifian S (2011) Efficient human detection based on parallel implementation of gradient and texture feature extraction methods. Proc. machine vision and image processing (MVIP), p 1–5
12. Enzweiler M, Gavrila DM (2011) A multilevel mixture-of-experts framework for pedestrian classification. IEEE Trans Image Process 20(10):2967–2979
13. Wang X, Wang M, Li W (2014) Scene-specific pedestrian detection for static video surveillance. IEEE Trans Pattern Anal Mach Intell 36(2):361–374
14. Prest A, Schmid C, Ferrari V (2012) Weakly supervised learning of interactions between humans and objects. IEEE Trans Pattern Anal Mach Intell 34(3):601–614
15. Yao BZ, Nie BX, Liu Z, Zhu S-C (2014) Animated pose templates for modeling and detecting human actions. IEEE Trans Pattern Anal Mach Intell 36(3):436–452
16. Wu J, Hu D (2014) Learning effective event models to recognize a large number of human actions. IEEE Trans Multimedia 16(1):147–158
17. Yao B, Li F-F (2012) Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. IEEE Trans Pattern Anal Mach Intell 34 (9):1691–1703
18. Farhadi A, Sadeghi MA (2013) Phrasal recognition. IEEE Trans Pattern Anal Mach Intell 35 (12):2854–2865
19. Rakate GR, Borhade SR, Jadhav PS, Shah MS (2012) Advanced pedestrian detection system using combination of Haar-like features, Adaboost algorithm and Edgelet-Shapelet. IEEE International Computational Intelligence and Computing Research Conf. (ICCIC), Coimbatore, Dec 2012, p 1–5
20. Gressmann M, Palm G, Lohlein O (2011) Surround view pedestrian detection using heterogeneous classifier cascades. IEEE International Intelligent Transportation Systems Conf. (ITSC), Washington, DC, Oct 2011, p 1317–1324
21. Prioletti A, Mogelmose A, Grisleri P, Trivedi MM, Broggi A, Moeslund TB (2013) Part-based pedestrian detection and feature-based tracking for driver assistance: real-time, robust algorithms, and evaluation. IEEE Int Intell Transp Syst 14(3):1346–1359
22. Li Y, Li B, Bin T, Yao Q (2013) Vehicle detection based on the and- or graph for congested traffic conditions. IEEE Trans Intell Transp Syst 14(2):984–993
23. Lv Y, Yao B, Wang Y, Zhu S-C (2012) Reconfigurable templates for robust vehicle detection and classification. Proc. IEEE workshop applications of computer vision (WACV), p 321–328
24. Enzweiler M, Hummel M, Pfeiffer D, Franke U (2012) Efficient Stixel-based object recognition. Proc. IEEE conf. intelligent vehicles symposium (IV), p 1066–1071

25. Niknejad HT, Takeuchi A, Mita S, McAllester D (2012) On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. IEEE Trans Intell Transp Syst 13(2):748–758
26. Feris R, Datta A, Pankanti S, Sun M-T (2013) Boosting object detection performance in crowded surveillance videos. Proc. IEEE workshop applications of computer vision (WACV), p 427–432
27. Sivaraman S, Trivedi MM (2013) A review of recent developments in vision-based vehicle detection. Proc. IEEE conf. intelligent vehicles symposium (IV), p 310–315
28. Chunpeng W, Lijuan D, Jun M, Faming F, Xuebin W (2009) Detection of front-view vehicle with occlusions using AdaBoost. IEEE International Engineering and Computer Science, ICIECS 2009, Wuhan, 19–20 Dec 2009, p 1–4
29. Bobo D, Wei L, Pengyu F, Chunyang Y, Xuezhi W, Huai Y (2009) Real-time on-road vehicle and motorcycle detection using a single camera. IEEE international industrial technology, Gippsland, Feb 2009, VIC, p 1–6
30. Chen K-H, Ju T-F, Lu W-M, Guo J-I (2014) Vision-based multiple moving objects detection for intelligent automobiles. Int J Electr Eng 21(6):201–213
31. Choi MJ, Torralba A, Willsky AS (2012) A tree-based context model for object recognition. IEEE Trans Pattern Anal Mach Intell 34(2):240–252

Lane Departure Warning System

32. Lindner P, Blokzyl S, Wanielik G, Scheunert U (2010) Applying multi-level processing for robust geometric lane feature extraction. Proc. 2010 I.E. conference on multisensor fusion and integration for intelligent systems (MFI), Sept 2010, p 248–254
33. Lin Q, Han Y, Hahn H (2010) Real-time lane detection based on extended edge-linking algorithm. Proc. 2010 2nd international conference on computer research and development, May 2010, p 725–730
34. Chang CY, Lin CH (2012) An efficient method for lane-mark extraction in complex condition. Proc. international conference on intelligence and computing and international conference on autonomic and trusted computing, Sept 2012, p 330–336
35. Yoo H, Yang U, Sohn K (2013) Gradient-enhancing conversion for illumination-robust lane detection. IEEE Trans Intell Transp Syst 14(3):1083–1094
36. Ge PS, Guo L, Xu GK, Zhang RH, Zhang T (2012) A real-time lane detection algorithm based on intelligent CCD parameters regulation. Publishing in Discrete Dynamics in Nature and Society, Discret Dyn Nat Soc p 1–16
37. Huo CL, Yu YH, Sun TY (2012) Lane departure warning system based on dynamic vanishing point adjustment. Proc. of 2012 I.E. 1st conference on consumer electronics, p 25–28

Forward Collision Warning System

38. Sun Z, Bebis G, Miller R (2006) On-road vehicle detection: a review. IEEE Trans Pattern Anal Mach Intell 28(5):694–711
39. Kumar U (2013) Vehicle detection in monocular night-time gray-level videos. Proc. 28th international conference in image and vision computing, New Zealand, Nov 2013, p 214–219
40. Teoh S, Brunl T (2012) Symmetry-based monocular vehicle detection system. Mach Vis Appl 23:831–842

41. Fossati A, Schönmann P, Fua P (2011) Real-time vehicle tracking for driving assistance. Mach Vis Appl 22(2):439–448
42. Khairdoost N, Monadjemi SA, Jamshidi K (2013) Front and rear vehicle detection using hypothesis generation and verification. Signal Image Process 4(4):31–50
43. Park K-Y, Hwang S-Y (2014) Robust range estimation with a monocular camera for vision-based forward collision warning system. Sci World J 2014:1–9

Speed Limit Detection System

44. Barnes N, Loy G (2006) Real-time regular polygonal sign detection. Springer Tracts Adv Robot 25:55–66
45. Loy G, Bames N (2004) Fast shape-based road sign detection for a driver assistance system. Proc. IEEE/RSL international conference on intelligent robots and systems, 28 Sept–2 Oct 2004
46. Keller CG, Sprunk C, Bahlmann C, Giebel J, Baratoff G (2008) Real-time recognition of U.S. speed signs. Proc. 2008 I.E. intelligent vehicles symposium, The Netherlands, 4–6 June 2008
47. Abukhait J, Zyout I, Mansour AM (2013) Speed sign recognition using shape-based features. Int J Comput Appl 84(15):32–37
48. Liu W, Lv J, Gao H, Duan B, Yuan H, Zhao H (2011) An efficient real-time speed limit signs recognition based on rotation invariant feature. Proc. 2011 I.E. intelligent vehicles symposium (IV), Baden-Baden, 5–9 June 2011
49. Lin H-H (1998) Recognition of printed digits of low resolution. Thesis, Institute of Computer and Information Science, National Chiao Tung University
50. Sharma R, Jain A, Sharma R, Wadhwa J (2013) Character and digit recognition aided by mathematical morphology. Int J Comput Technol Appl 4(5):828–832
51. Moutarde F, Bargeton A, Herbin A, Chanussot L (2007) Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular traffic signs recognition system. Proc. 2007 I.E. intelligent vehicles symposium, Istanbul, 13–15 June 2007
52. Glavtchev V, Muyan-Özçelik P, Ota JM, Owens JD (2011) Feature-based speed limit sign detection using a graphics processing unit. Proc. 2011 I.E. intelligent vehicles symposium (IV), Baden-Baden, 5–9 June 2011
53. Torresen J, Bakke JW, Sekanina L (2004) Efficient recognition of speed limit signs. Proc. 2004 I.E. intelligent transportation systems conference, Washington, DC, 3–6 Oct 2004
54. Chen L, Li Q, Li M, Mao Q (2011) Traffic sign detection and recognition for intelligent vehicle. Proc. 2011 I.E. intelligent vehicles symposium (IV), Baden-Baden, Germany, 5–9 June 2011
55. Zaklouta F, Stanciulescu B (2014) Real-time traffic sign recognition in three stages. Robot Auton Syst 62(1):16–24

Inclement Weather Processing Technology (DLCE)

56. Schechner YY, Narasimhan SG, Nayar SK (2001) Instant dehazing of images using polarization. Proc IEEE Conf Comput Vis Pattern Recognit 1:325–332
57. Shwartz S, Namer E, Schechner Y (2006) Blind haze separation. Proc IEEE Conf Comput Vis Pattern Recognit 2:1984–1991

58. Tan RT (2008) Visibility in bad weather from a single image. Proc. IEEE conf. on computer vision and pattern recognition, CVPR 2008, June 2008, p 1–8
59. Fattal R, 2008 (2008) Single image dehazing. ACM Trans Graph 27(3):1–9
60. He K, Sun J, Tang X (2009) Single image haze removal using dark channel prior. Proc. IEEE conf. on computer vision and pattern recognition, CVPR 2009, June 2009, p 1956–1963
61. He K, Sun J, Tang X (2013) Guided image filtering. IEEE Trans Pattern Anal Mach Intell 35 (6):1397–1409
62. Park G-H, Cho H-H, Choi M-R (2008) A contrast enhancement method using dynamic range separate histogram equalization. IEEE Trans Consum Electron 54(4):1981–1987
63. Zhiming W, Jianhua T (2006) A fast implementation of adaptive histogram equalization. 2006 8th international conference on signal processing, vol. 2
64. Pizer SM, Amburn EP, John DA, Robert C, Ari G, Trey G, Bart Ter Haar R, John BZ (1987) Adaptive histogram equalization and its variations. Proc Comput Vis Graph Image Process 39 (3):355–368
65. Zuiderveld K (1994) Contrast limited adaptive histogram equalization. Publishing in Graphics gems. In: Graphics gems IV. p 474–485
66. Narasimhan SG, Nayar SK (2003) Contrast restoration of weather degraded images. IEEE Trans Pattern Anal Mach Intell 25:713–724
67. Xu H, Guo J, Liu Q, Ye L (2012) Fast image dehazing using improved dark channel prior. Proc. IEEE conf. on information science and technology, ICIST, March 2012, p 663–667
68. Lv X, Chen W, Shen I (2010) Real-time dehazing for image and video. Proc. Pacific conf. on computer graphics and applications, PG, Sept 2010, p 62–69