Proceedings of 2016 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Harbin, China

FPGA Based Real-time Vehicle Detection System under Complex Background

Jiaojiao Gu, Han Xiao, Wenhao He, Shijun Wang, Xiaonan Wang and Kui Yuan

Institute of Automation
Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing, China
gujiaojiao2012@ia.ac.cn

***Abstract* - In this paper, a real-time vehicle detection system is designed and implemented on an FPGA (Field Programmable Gate Array). The system is composed of an infrared camera and an image acquisition and processing board developed by our research team. An FPGA chip and a DSP chip are embedded in the image board as the major calculation units, which make real-time computation possible. First, edge features of the training samples are extracted according to Naive Bayes Model and then fused with brightness features to construct the detector template on the host computer. Second, real-time vehicle detection experiments are conducted according to the detector template on the image acquisition and processing board. In contrast to conventional methods in which a huge number of negative samples are collected for training, an even distribution is assumed in our method and no negative samples are needed, thus greatly shortening the training time. Experiment results show that the algorithm can detect different shapes and sizes of vehicle objects efficiently.**

Index Terms - vehicle detection, real-time, Naive Bayes, embedded system, infrared image.

I. INTRODUCTION

With the rapid development of technology, missiles and other offensive weapons play an important role in modern military armed conflicts, which pose serious threats to the safety of planes, warships and other high-value fighting platforms. However, on modern high-tech battlefields, the distance between the military camera and hostile object is usually very far away, which results in the object occupying quite a low percentage in the image[1]. Long range imaging is influenced by many factors, such as weather, seasons, day/night and the cloud, which make it almost impossible to extract any gray degree features or texture features from the small object[2-3]. This restriction makes it difficult for humans to distinguish objects from the surrounding environments with naked eyes.

Infrared imaging technology works by receiving targets' infrared radiation passively and has advantages of long range imaging, good concealing ability, high accuracy, strong anti-electromagnetic interference capability, working around the clock, and so on[4]. It has been widely used in civilian and military applications, such as infrared precise guidance, hazard warning and video surveillance.

In recent years, researchers have proposed lots of methods to detect targets in infrared (IR) video sequences, but most of them focused on detect moving objects only and couldn't detect stationary objects at the same time. What's more, some of them emphasize the effectiveness and accuracy of the algorithm proposed in their papers and ignore the real-time performance. However, to rapidly and accurately find out the hostile objects, no matter moving objects or stationary objects, is a more important factor that affects the success of modern high-tech war. So in this paper, we will not only pay attention to the precision, but also the speed.

The hostile objects on the battlefield mainly contain pedestrians, cars, tanks, planes and so on. As pedestrians have stable body temperature, so they remain the same brightness in IR images, which can be detected by their shape features and brightness information[5]. However, the majority of a vehicle body is metal, whose temperature is greatly affected by illumination, surrounding environments' temperature and other factors. Especially when a car is driving, both the engine and the friction between tires and ground will generate heat, so the temperature will raise over time, which leads to different vehicle targets having brightness difference in the IR image.

It is well known that vehicle tracking processes are very computationally intensive. Traditionally, vehicle tracking algorithms have been implemented using software approaches. The software approaches have a large computational delay, which causes low frame rate vehicle tracking[6]. Therefore, in order to detect objects in real-time, it's necessary to develop high-performance embedded software and hardware system. Using an FPGA chip and a DSP chip as the main computing elements, our research team developed an intelligent image acquisition and processing board to realize the real-time computation for complex algorithms.

The Naive Bayes Model is a simple and well-known method for performing supervised learning of a classification problem[7-8]. It is based on a simple result from elementary statistics known as Bayes Theorem[9]. The method is relatively easy to code in both software and hardware, and comparable in performance with other more sophisticated techniques[10]. Therefore, this paper designs a real-time IR vehicle detection system based on the fusion of edge features and brightness features, implements the naive Bayes algorithm on FPGA, which can quickly and effectively detect the vehicle targets under various conditions.

The rest of the paper is organized as follows. Section II briefly introduces the intelligent image acquisition and processing embedded board developed by our research team. A general framework of the vehicle detection algorithm is

outlined in Section III, which provides a detailed description of the methodologies. In Section IV, experiments are conducted to test the performance of the system and the algorithm. Section V makes conclusions of this paper.

II. Intelligent Image Acquisition And Processing Embedded Board

Figure 1 shows the experimental equipments, which are mainly composed of an infrared camera, a tablet and an intelligent image acquisition and processing board. The band of the IR camera is 800μm ~ 1400μm, the tablet charges the IR camera and controls the platform of the IR camera moving towards up, down, left and right. The image processing board's core computing elements are an FPGA chip and a DSP chip(see Fig. 2). The FPGA used on the board is Altera Cyclone III EP3C40F484, and the DSP on the board is TI's TMS320DM642. The board contains two SRAMs, two analog video input ports and one analog video output port. Analog video signal of PAL format is converted to BT.656 digital signal by the A/D converter TVP5150. Then the digital signal is transferred into the FPGA. The pins of the FPGA are connected with the two SRAMs, where the image data and results could be saved. Besides, the pins of DSP and FPGA are connected, which makes DSP convenient to access the memory resources in the FPGA and the SRAM on the board. The results and images can be output to the PC through the 100-Mbps Ethernet port. In addition, the gigabit Ethernet port can connect the high-speed digital camera to achieve faster image acquisition speed.

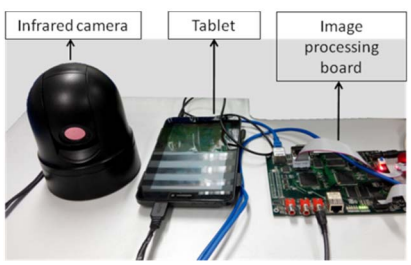

Fig. 1 The experimental equipments.

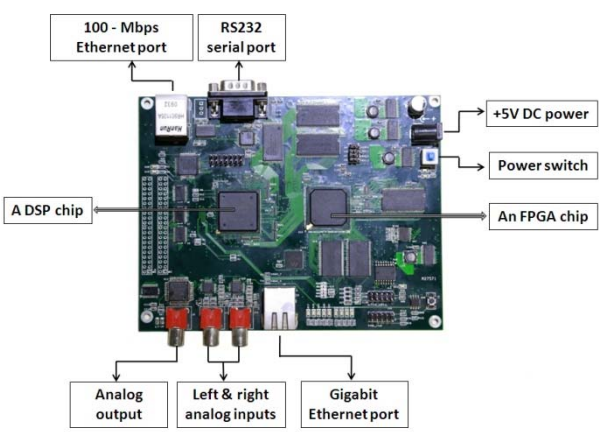

Fig. 2 The intelligent image acquisition and processing board.

III. Real-time Vehicle Detection Algorithm

As different brands and models of vehicles are different in size and shape, moreover, the materials of engines and tires have differences, so different brands and models of vehicle targets vary greatly in the IR image. Here are some possible situations(see Fig. 3): (1) both the engine and tires are brighter than the surrounding environment; (2) both the engine and tires are similar in brightness with the surrounding environment; (3) the engine is brighter than the surrounding environment, while the tires are similar in brightness with the surrounding environment; (4) the tires are brighter than the surrounding environment, while the engine is similar in brightness with the surrounding environment; (5) both the engine and tires are darker than the surrounding environment.

IR imaging target detection system works according to the principle of temperature difference, and there exist vehicle temperature difference between vehicle objects and the background, resulting in the grayscale of vehicle edges change greatly. Therefore, this paper combines the edge features of vehicle objects with Naive Bayes method and implements them on FPGA. To further reduce false positives, brightness features is used as an additional judging criterion. We set edges' pixel number of a vehicle to a certain range. By adding this criterion, false positives are reduced to a reasonable level.

In this paper, the basic idea of the real-time vehicle detection system can be divided into three parts, and the overall implementation scheme is shown in Fig. 4. Firstly, image filtering, edge extraction, image binaryzation and Naive Bayes based object detection are hardware implemented on FPGA, objects' coordinates are also calculated on FPGA and then passed to the SRAM. Secondly, DSP gets the coordinates from the SRAM, marks vehicle objects with rectangles according to the coordinates and then transmits the detection results to PC. Finally, PC displays the detection results and modifies detection parameters of FPGA online, the modified parameters are passed from PC to FPGA through DSP by the 100-Mbps Ethernet interface on the image processing board.

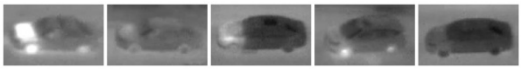

Fig. 3 The possible situations of different vehicles.

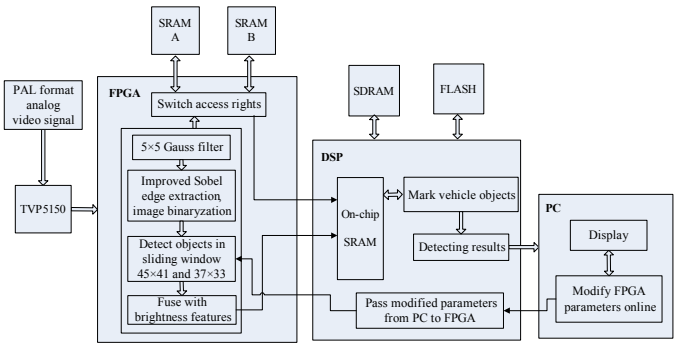

Fig. 4 The overall implementation scheme.

A. Background on Naive Bayes

1) Bayes Theorem

Bayes' rule is an equation for computing conditional probabilities, see (1). A conditional probability is defined as the probability of an event given that another event has occurred. For example, consider two events H and X with $P(X) > 0$. The conditional probability, denoted as $P(H|X)$, is the probability of event H given that event X has occurred. According to the Bayes' rule, $P(H|X)$ is calculated as:

$$P(H|X) = \frac{P(X \mid H)P(H)}{P(X)} \tag{1}$$

$P(X)$ is called the prior probability of H because it is the probability of event X before the occurrence of event H. $P(X|H)$ is called posterior probability because it is the probability of event X *after* the occurrence of event H.

2) Naive Bayes Classification

Naive Bayes is based on the Bayes' rule. Let X be a data sample whose class label is unknown. Let H be some hypothesis, such as the sample X belongs to some type of class C. Then the Naive Bayesian classifier works as follows:

The data sample X is represented by a n-dimensional vector $X = (x_1, x_2, ..., x_n)$ and this has been contributed by n different attributes such as $A_1, A_2, ..., A_n$.

Let us consider that there are m classes, $C_1, C_2, ..., C_m$. We also have a given Sample, X, for which the Naive Bayes classifier would assign the unknown Sample X to the class C_i if and only if

$$P(C_i \mid X) > P(C_j \mid X), \text{ for } 1 \le j \le m,\ j \ne i. \tag{2}$$

This is known as maximum posterior hypothesis. Hence by Bayes Theorem,

$$P(C_i|X) = \frac{p(X \mid C_i)P(C_i)}{P(X)} \tag{3}$$

Apparently $P(X)$ in (3) is a constant value for all classes, so we need to maximize $P(X \mid C_i)\, P(C_i)$. We estimate

$$P(C_i) = T_i / T \tag{4}$$

Where T_i is the total number of training samples of class C_i and T is the total number of training samples.

It would be extremely computationally expensive to compute $P(X \mid C_i)$ when we have datasets with many attributes. In order to reduce this complexity, we make the assumption of class conditional independence. We mean that the values of the attribute are conditionally independent of one another, given the class label. Hence,

$$P(X \mid C_i) = P(x_1 \mid C_i)\, P(x_2 \mid C_i) \ldots P(x_n \mid C_i) \tag{5}$$

Where probabilities $P(x_1 \mid C_i)$, $P(x_2 \mid C_i)$, …, $P(x_n \mid C_i)$ can be estimated from the training samples.

$$P(x_k \mid C_i) = T_{ik} / T_i \tag{6}$$

Where T_{ik} is the total number of training samples of class C_i having the value x_k for A_k and T_i is the total number of training samples belonging to C_i.

In order to classify an unknown sample X, $P(X \mid C_i)\, P(C_i)$ is evaluated for each Class C_i. Sample X is then assigned to the class C_i if and only if

$$P(X \mid C_i)\, P(C_i) > P(X \mid C_j)\, P(C_j), \text{ for } 1 \le j \le m,\ j \ne i. \tag{7}$$

B. Image pre-processing

Firstly, we conduct Gaussian filtering for the IR image through a 5×5 Gaussian filter (see TABLE I) on the FPGA. According to the characteristics of FPGA pipeline structure, we split the Gaussian filter template into two directions: the horizontal template, see TABLE II (1), and the vertical template, see TABLE II (2). We conduct vertical filtering first and then apply horizontal filtering to the vertical filtering result.

Secondly, we extract edges on the FPGA by an improved Sobel operator. The improved Sobel operator not only contains the traditional Sobel operator, but also adds another two direction operator, 45° direction and 135° direction, as shown in TABLE III. We calculate the four direction operators' values of each pixel, and choose the maximum one as the pixel's edge grey-level value, thereby we get the edge pixel stream. On this basis, we make binaryzation to the edge pixel stream and get the binarized edge stream.

C. Implementation of Naive Bayes on FPGA

Firstly, as the binary pixel stream is generated by the aforementioned operation, a sliding window for vehicle object's edge detection is opened on the binary image. All the vehicle objects are detected within this window, and therefore it is called "the top window". After that, two detectors are applied in the top window to detect vehicles of different sizes. In our experiment, the sizes of the detectors for each object at each scale are 45×41 and 37×33, see Fig. 5.

Secondly, We take 300 vehicle object IR images as training samples, some of them are shown in Fig. 6. The training samples are adjusted to two kinds of uniform size, 55×90 and 45×70.

In order to gain more robustness against scale variations and shape variations, only the left part of vehicle bodies are used for training, whose sizes are 55×45 and 45×35, the corresponding binary images are shown in Fig. 7.

The probability distribution obtained from these 300 images is shown in Fig.8, in which brighter areas have higher probabilities of containing white pixels.

TABLE I
5×5 GAUSSIAN FILTER TEMPLATE

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

TABLE II
THE HORIZONTAL AND VERTICAL GAUSSIAN FILTER TEMPLATE

1	4	6	4	1

(1)

1
4
6
4
1

(2)

TABLE III

THE IMPROVED SOBEL OPERATOR

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

0	1	2
-1	0	1
-2	-1	0

2	1	0
1	0	-1
0	-1	-2

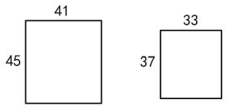

Fig. 5 Two vehicle detectors.

To reduce space complexity on FPGA further, we cut off part of Fig 8, thus we get the final probability distribution, whose sizes are 45×41 and 37×33 (Fig. 9).

Thirdly, in this paper, to facilitate the fixed-point number calculations on the FPGA, we turn the probabilities into fractional numbers with a unified denominator 128 and only use the numerators to construct the probability matrices. Each element in these two matrices represents the probability of a white pixel appearing in the corresponding position. On this basis, the corresponding decimals in the probability distribution matrices(PDMs) of Fig.9 are turned to positive integers, see (8).

$$Q(X \mid C_i) = \text{round}(128 * P(X \mid C_i)) \tag{8}$$

By subtracting each element from 128, the two matrices can be turned into complementary matrices in which each element represents the probability of a black pixel appearing in that position.

Judging whether a vehicle exists in the detector window is a two-class classification problem. For the positive class (vehicles), the PDMs of white pixels are calculated according to (8).

For the negative class(background), we assume that the probability of a pixel being white or black is equal, resulting in two PDMs in which every element is 64.

Naive Bayes assumes that each pixel is independent. Although this assumption is not the truth, the performance of a Naive Bayes classifier is surprisingly good because it is actually using a super ellipsoid to enwrap the positive class. On this basis, the product of many elements in the PDMs has to be computed to get the "likelihood", as shown in (5) of Section III *A*.

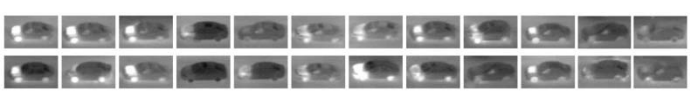

Fig. 6 A part of the training samples.

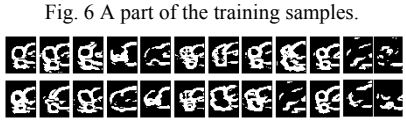

Fig. 7 The corresponding binary images of Fig. 6.

Fig. 8 The probability distribution obtained from 300 images.

Fig. 9 The final probability distribution.

However, this will require lots of hardware resources and be time-consuming in the FPGA. To reduce space complexity and computation complexity on FPGA further, we switch the multiplication to addition by performing logarithmic operation with base 2. Then we multiply it by 32 and round the result to get the logarithmic probability matrix (LPM) with fixed-point numbers, see (9)

$$R(X \mid C_i) = \text{round}(32 * \log_2 (Q(X \mid C_i))) \tag{9}$$

We record the LPM for white pixels in the 45×41 detector as LPMW(45×41). Correspondingly, the LPM for black pixels in the 45×41 detector is recorded as LPMB(45×41). Also, there are LPMW(37×33) and LPMB(37×33), which can be calculated from (8) and (9).

For example, if a position is a white pixel and the probability of a white pixel appearing in this position is 0.7007, as 0.7007 = 0.7007 * 128 / 128 = 89.6896 / 128, round(89.6896) =90, and round(32*$\log_2$(90))=208, so 0.7007 is switched to 208 in LPMW.

For the negative class, as every element in the two PDMs is 64, and round(32*$\log_2$(64)) = 192, so every element in LPM(45×41) and LPM(37×33) is 192.

According to the rule of logarithmic operations, we can get the following equation, as shown in (10) .

$$\begin{aligned} &\log(Q(X \mid C_i)\, Q(C_i)) \\ &= \log(Q(x_1 \mid C_i)\, Q(x_2 \mid C_i) \ldots Q(x_n \mid C_i) Q(C_i)) \\ &= \sum_{k=1}^{n} \log(Q(x_k \mid C_i)) + \log(Q(C_i)) \end{aligned} \tag{10}$$

Based on (5) (7) (8) (9) (10), we find that maximizing $P(X \mid C_i)\, P(C_i)$ equals maximizing $\sum_{k=1}^{n} R(x_k \mid C_i)) + R(C_i)$. Hence, sample X is then assigned to the class C_i if and only if

$$\sum_{k=1}^{n} R(x_k \mid C_i)) + R(C_i) > \sum_{k=1}^{n} R(x_k \mid C_j) + R(C_j) ,$$

for $1 \le j \le m,\ j \ne i$. (11)

Finally, for each white pixel in the detector window, the element in the corresponding position of LPMW is chosen; for each black pixel in the detector window, the element in the corresponding position of LPMB is chosen.

By adding these elements, the likelihood of the content being a vehicle is computed, which we record as $L_{vehicle}$. Meanwhile, the likelihood of the content being background $L_{background}$ is a constant value for each detector window, because an even distribution of the pixel values is assumed for the negative class. The criteria for judging can then be written as (12) .

$$IsVehicle = \begin{cases} 1, & L_{vehicle} \ge L_{background} + K \\ 0, & L_{vehicle} < L_{background} + K \end{cases} \tag{12}$$

Where K is an adjustable parameter representing the ratio of the prior probability of negative class to the prior probability of positive class.

To further reduce false positives, brightness features is used as an additional judging criterion. We set edge's pixel

number of a vehicle to a certain range. By adding this criterion, false positives are reduced to a reasonable level.

IV. Experiments And Results

A. *The Improved Sobel Operator.*

The contrast between the traditional Sobel operator and the improved Sobel operator is shown in Fig. 10.

The first column of the figure is the original image, the second column is the extracted edge of the traditional Sobel operator and its corresponding binary image, the third column is the extracted edge of the improved Sobel operator and its corresponding binary image.

From the figure, we see the traditional Sobel operator is just capable of extract distinct edges but incapable of extracting the blurry edges and the oblique edges, while the Sobel operator proposed in this paper can extract all the edges effectively.

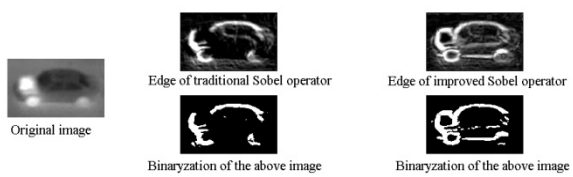

Fig. 10 Results of the improved Sobel operator.

B. *Implementation of Naive Bayes on FPGA.*

After offline training with the 300 binary images in Fig.7, the maximum, average and minimum of $L_{vehicle}$, $L_{background}$ and edges' pixel number of a vehicle for each of the 2 detectors are given in Table IV. Except for the negative class , we also list another two situations of the detector window - all black and all white.

From the table, we find:

First, in the all white situation, both meeting the requirements of the edge's pixel number alone and meeting the requirements of $L_{vehicle}$ and $L_{all\ white}$ criterion alone can separate the vehicle from all white picture for both the 45 x 41 and 37 x 33 detector at the same time.

TABLE IV
THE PARAMETERS FOR EACH DETECTOR

Size of detector(pixels)	45×41	37×33
max($L_{vehicle}$)	381929	253091
min($L_{vehicle}$)	344883	235791
average($L_{vehicle}$)	372243	245938
$L_{background}$	**354240**	**234432**
$L_{all\ black}$	371366	238435
$L_{all\ white}$	278834	196955
min($L_{vehicle}$) - $L_{background}$	-9357	1359
max (edge's pixel number) of the positive class	875	671
min (edge's pixel number) of the positive class	260	222
average (edge's pixel number) of the positive class	619	484
edge's pixel number of the negative class	922.5	610.5
edge's pixel number of the all black situation	0	0
edge's pixel number of the all white situation	0	0

Second, in the all black situation, only if meeting the requirements of the $L_{vehicle}$ and $L_{all\ black}$ criterion together with meeting the requirements the edge's pixel number can separate the vehicle from all black picture for both the 45 x 41 and 37 x 33 detector at the same time.

Third, in the negative class(background) situation, only if meeting the requirements of the $L_{vehicle}$ and $L_{background}$ criterion together with meeting the requirements the edge's pixel number can separate the vehicle from background for both the 45 x 41 and 37 x 33 detector at the same time.

To sum up, only if meeting the requirements of the $L_{vehicle}$ and $L_{background}$ criterion together with meeting the requirements the edge's pixel number can separate the vehicle from background, all white situation and all black situation for both the 45 x 41 and 37 x 33 detector at the same time. We set the thresholds for each detector according to outdoor experiments and list them in Table V.

TABLE V
THE THRESHOLDS FOR EACH DETECTOR

Size of detector(pixels)	45×41	37×33
Ratio of the Prior Probability (K)	12700	2478
Threshold of vehicle edge's pixel number	400	480

C. *The real-time* vehicle detection *system.*

Outdoor experiment results of real-time vehicle detection are shown in Fig. 11 and Fig. 12. The detected vehicles are marked with rectangles, and the numbers in the upper right corners of the images represent the num of detected vehicle objects in the image.

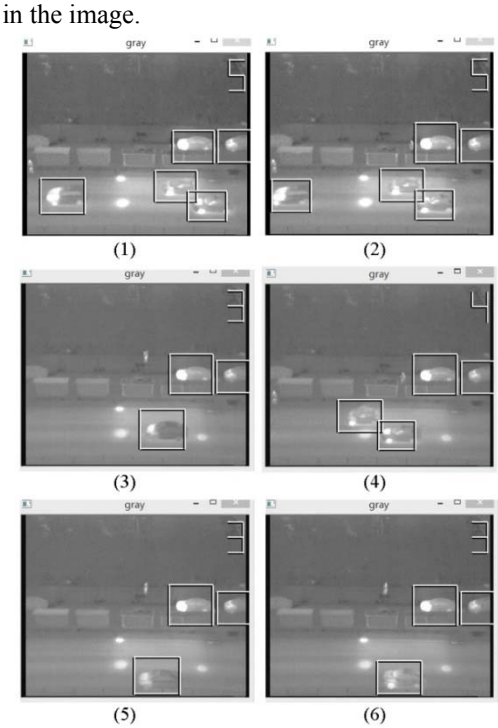

Fig. 11 The results of real-time vehicle detection.

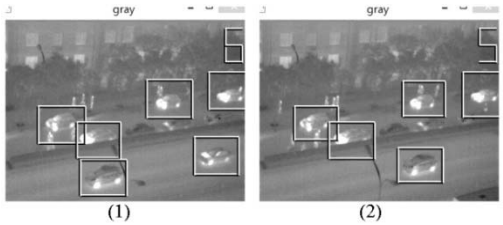

Fig. 12 The results of real-time vehicle detection.

From the two figures, we see:

First, although the detectors are intended for vehicles whose left lateral sides face to the camera(Fig. 11), they can simultaneously detect vehicles whose left lateral sides have an oblique angle with the camera (Fig. 12), which indicates that the system has good robustness and generalization performance.

Second, the detectors can simultaneously detect vehicles in another 6 different situations: vehicles in different sizes and shapes; both moving and stationary vehicles; both the engine and tires are brighter than the surrounding environment, see the bottom of Fig. 11(1); both the engine and tires are similar in brightness with the surrounding environment, see the bottom of Fig. 11(5); the engine is brighter than the surrounding environment, while the tires are similar in brightness with the surrounding environment, see Fig. 11(3); the tires are brighter than the surrounding environment, while the engine is similar in brightness with the surrounding environment, see the middle of Fig. 11(4) and the bottom of Fig. 12(2).

Third, the detectors can detect vehicles which are partially occluded by the billboards at the bus stop(Fig. 11) or the isolation belts (Fig. 12). Moreover, although the background is complex and the round manhole covers on the road as well as the windows of the buildings look similar to vehicles' engines and tires, still, the detectors can distinguish vehicles from pedestrians, trees, round manhole covers, windows of the buildings and other interferences correctly.

V. Conclusion

This paper designed and implemented a real-time vehicle object detection system on a low-power embedded image acquisition and processing board. Under a clock frequency of 50MHz, this designed system meets the real-time requirement for infrared image resolution (360×288) at 25 frame-per-second(fps). Experiment results show that the method is of good computation complexity, robustness and generalization performance, and can correctly and efficiently detect vehicle objects in different situations.

Acknowledgment

The work is supported by National Natural Science Foundation of China (No.61273360). Besides, the authors would like to thank the authors of references and the anonymous reviewers for improving the quality of this paper.

References

[1] J. Gu, Z. Wang, H. Song, et al, "A real-time small immobile object recognition system using wavelet moment invariants", IEEE International Conference on Mechatronics and Automation, 2015.
[2] C. Li, "Automatic Detection Method Of Infrared Ship Target Under Complex Sea-Sky Background" [D], Hunan Normal University, 2012.
[3] C. Wei, "Research on Infrared Small Target Detection and Tracking Algorithm" [D], Harbin Institute of Technology, 2009.
[4] J. Sun, "Research on Algorithm of Infrared Small Target Detection and Tracking in Image Sequence" [D]. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, 2014.
[5] H. Xiao, H. Song, et al, "Real-time shape and pedestrian detection with FPGA", IEEE International Conference on Mechatronics and Automation IEEE, 2015.
[6] P. Li, Z. Wang, H. Song, et al, "FPGA-Based Hardware Implementation of Image Processing Algorithms for Real-Time Vehicle Detection Applications"[J], Dissertations & Theses - Gradworks, 2012.
[7] K. Toni, "An Optimised Naive-Bayes Detection System" [D], Oklahoma State University, 2005.
[8] M. Cui, "Paper Categorization Using Naive Bayes" [D], University of Victoria, 2007.
[9] T. Bayes, "An essay towards solving a problem in the doctrine of chances", Philosophical Transactions of the Royal Society of London 1763,53:370-418.
[10] P. Liu, "Discriminative training of naive Bayes classifiers for natural language call routing" [D], York University, 2004.