1. Introduction

Drivers and passengers may be injured by a sudden stop or a high speed collision in a vehicle. To prevent such injuries, public security departments require that all drivers and passengers wear seat belts, which are designed as a buffer and decrease fatal injuries. Seat belt warning systems are also a mandatory requirement, which remind drivers to wear a seat belt.

Current seat belt warning systems consist of simple metal sensors installed in the seat belt buckle. It is easy for drivers to circumvent the sensor tip without wearing a seat belt. In response to the demands of public security departments, seat belt detection methods based on surveillance videos have been proposed. Existing seat belt detection methods are mostly based on edge detection and the Hough transform method. Guo et al. [1] presented a seat belt detection system that consisted of two parts, namely, driver area positioning and seat belt detection. For the driver area, they used a geometric positioning method based on the proportional relationship with the license plate, windshield, and vehicle. This algorithm first uses a linear filter in HSV color space to locate the license plate through horizontal and vertical projections. It then uses an edge detection operator (e.g., Sobel or Canny) to find the top and bottom edge of the windshield. Finally, it determines the position of the driver according to the proportional relationships with the license plate,the windshields, and vehicle. The performance of this method is greatly influenced by the vehicle color. Additionally, the angle of the camera has too much of an effect to the geometrical relationships that determine the driver area. For seat belt detection, they used a method based on edge detection and the Hough transform. The image quality requirement for this method is relatively high.Additionally, a threshold is set according to a specific image set, which is lack of robustness and hard to popularize.

To mitigate for the influences of the illumination, vehicle body drivers clothes and other factors in the seat belt detection process, we propose a seat belt detec-

tion system based on the convolution neural network (CNN) and SVM. CNN is a multi-layer neural network that uses supervised learning [2] to extract global features. Its feature extraction function has a kernel model that implements convolution and pooling in the hidden layer. The network model uses the gradient descent method to minimize the loss function, so that it can reverse adjust the weighting parameters of the networks layers, improving the accuracy of the network using an iterative training process. Multi-scale CNN feature extraction [3] is a from-overall-to-part process, in which the classifier can be fed with the features extracted by multiple stages. The motivation for using features extracted by multiple stages in the classifier is to provide different scales of receptive fields to the classifier. This allows the classifier to use, not only high-level and invariant features with little details, but also low-level features with precise details. Therefore, we chose Multi-scale CNN combining the global and the local features to identify seat belt better. First, we run the vehicle detection and localization processes, which are used to determine the windshield location. Then several candidate seat belt areas are derived to combine with the vehicle and windshield positioning results. We obtain the final seat belt detection using a support vector machine (SVM) classifier.

In this paper, we present a new type of seat belt detection model that calculates multi-scale deep features using a CNN. Unlike the traditional detection method based on Haar-like [4] features, the CNN has recently been successfully applied to image recognition [5, 6, 7, 8, 9, 10]. The deep network automatically extracts multi-layer features from the original image. Typically, numerous features are extracted by such a network, and they are more efficient than the traditional subjective features. Inspired by this, we trained a feature extraction model based on CNN using an image dataset. In this paper, we call these derived features CNN features.

To determine a more accurate seatbelt position, we must combine its neighboring regions and the whole vehicle body to calculate the features. As a result, our seat belt detection model detects the belt area, windshield area, and vehicle body. Therefore, we extract multi-scale CNN features for every image from

three nested and increasingly larger rectangular windows, which enclose the seat belt area, the windshield area, and the vehicle body area.

In addition to the multi-scale CNN features, we further trained a fully connected neural network. Multi-scale CNN features are fed into connection layers, which are trained on labeled image data. Thus, these full connection layers act as a regression, which calculates the detection score of every image region. It is well known that deep neural networks have at least one full connection layer, which can be used to train to a high level of regression accuracy.

We have extensively evaluated our CNN-based seat belt detection model over existing datasets. There is no ready-made seat belt detection database; therefore, we collected many vehicle images from real road images, which were provided by the traffic administration, and set up an experimental database. This database contains belt images and non-belt images, and it includes different backgrounds, illuminations, and vehicle types.

In summary, this paper makes the following contributions.

(1) We propose a new type of seat belt detection method that incorporates multi-scale CNN features extracted from nested windows by a deep neural network on multiple full connection layers. The detection score of each detection window is used to train a SVM classifier.

(2) We designed a complete seat belt detection, which is intuitive, accurate, and robust.

(3) We constructed a seat belt detection database that includes many vehicles and seat belt areas, which can be used for training detection models in follow-up studies.

2. Related work

Seat belt detection system typically comprises three parts: vehicle detection, windshield detection, and seat belt detection. Currently, vehicle and windshield detection algorithms are mostly based on edge detection algorithms [11]. They first detect the horizontal and vertical edges of a vehicle image. Then, they

remove the non-vehicle edges, and calculate the symmetry of the vertical edge. Finally, they extract the target region, and get the position of the vehicle.

For windshield detection, the result is greatly influenced by the color of the vehicle body and lighting conditions. In [12], the windshield detection algorithm based on the HIS color model and genetic algorithms has been proposed. This algorithm requires many iterations, and it often does not satisfies real-time requirements. In [13], the authors proposed a vehicle detection method based on the HSV color model, which successfully detects windshields on dark-colored vehicle bodies. However, it does not perform well for light-colored vehicles and strong light conditions. The Adaboost algorithm was used in [14] to measure the angular point positioning of the windshield. Adaboost [15, 16] is an improved algorithm based on Boosting [17]. This algorithm is very fast and is extensively used in object detection and recognition. First, features are extracted from a large number of windshield samples. These weak classifiers are combined to derive a strong classifier. Finally, the strong classifier is used to position the windshield regions.

There is a limited amount of research related to seat belt detection [1]. Most investigations directly used a line detection method based on the Hough transform. This method is sensitively influenced by vehicle factors such as the steering wheel and the drivers clothing. Then seat belt detection system based on a SVM [18, 19, 20] is designed, using a driver positioning method similar to that in [1]. Some belt related parameters are extracted from the driver region to compose an eigenvector for training the SVM classifier in the seat belt detection process. In this paper, we propose a seat belt detection method that combines an SVM classifier and multi-scale CNN features [21, 22]. This method is robust in different background environments. [23] CNNs have recently been successfully applied to many visual recognition tasks, including image classification [8, 24], object detection [6, 25], and scene parsing [5, 26, 23]. Donahue et al [27] noted that features extracted from Krizhevskys CNN trained on the ImageNet dataset [28] can be repurposed for generic tasks. Razavian et al. [29] extended those results, and concluded that deep learning using a

CNN can be a strong candidate for any visual recognition task. However, CNN features have not been applied to seat belt detection. We propose a simple but very effective neural network that makes the CNN features more applicable for seat belt detection.

3. Seat belt detection based on CNN and SVM

Deep learning uses a large number of simple neurons, each of which receives the output of the lower level neuron. According to the non-linear relationship between the input and the output, the low-level features are combined into a higher level abstract representation, which determines the distributed features of the observation data. Thus, a multi-layer abstract representation is formed using a bottom-up study. Multi-layer feature learning is an automatic process without human interventions. According to the learned network structure, the deep learning process maps the inputs to various feature levels, and uses a classifier or matching algorithm to classify or identify the output of the top layer.

We designed and implemented a seat belt detection system based on CNN. By processing the images captured by road surveillance cameras, we can distinguish whether the driver is wearing a seat belt. After the captured vehicle images are sent to the coarse vehicle positioning module, several candidate vehicle areas are found in the images. We then use the coarse windshield positioning module to find several candidate windshield regions. These results are fed into the seat belt detection module to find a number of seat belt detection areas. Finally, we calculate the accurate detection result of the seat belt through post-processing using an SVM classifier. Figure 1 contains a structural diagram of the system.

3.1. Deep learning

We consider a system $S = (S_1, \ldots, S_n)$, which is an n-layer structure. I is the input to the system, and O is the output. It can be expressed as $I \rightarrow$

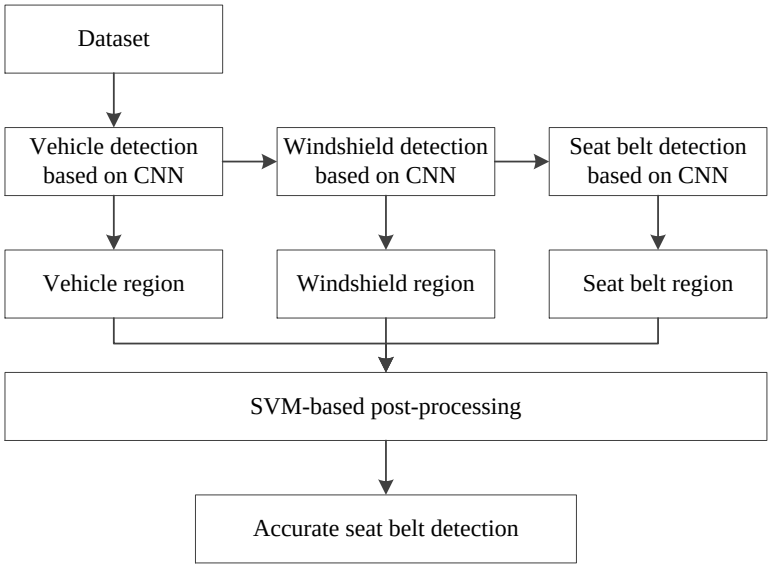

Figure 1. Seat belt detection system based on convolution neural network (CNN).

$S_1 \to S_2 \to \cdots \to S_n \to O$. If the output O equals the input I, the information contained in O is the same as the original input, which indicates that there was no information lost when passing through each layer (S_i). That is, O is another representation of the input, I (the original information). In brief, the essence of deep learning is that the input is equal to the output for any layer of an n-layer neural network. Ideally, no human intervention would occur during the learning process, such that that the network can automatically learn object features. Given an input I, when we pass through an n-layer system S, we adjust the system parameters so that the output equals the input. This means that O is still I. Finally, we get a series of level features for I that correspond to $S_1, \ldots, S_n$

The above concept is based on the assumption that the output is strictly equal to the input. Equal has two meanings; the output equals to the input in an abstract sense and not absolutely, and with respect to the extent of the constraints. For example, it is a complete equal without any ambiguity or an

equal with appropriate loose conditions. The absolute sense of equal is too strict; therefore, if the difference between the input and the output is as small as possible, we can slightly relax these restrictions.

3.2. CNN feature extraction

A CNN is a multi-layer neural network. Each layer is composed of multiple two dimensional planes, and each plane is made up of several independent neurons. The network includes some simple and complex neurons, denoted as S-neurons and C-neurons respectively. S-neurons aggregate together to make up the S-plane, and S-planes aggregate together to build the S-layer, which is denoted as U_s. The C-neuron, C-plane, and C-layer (U_c) are the same. Every intermediate level of the network is concatenated by the S-layer and C-layer; however, the input layer contains only one layer, and it directly receives the two dimensional visual patterns. The feature extraction steps for the samples are embedded into the interconnected structure of the CNN model.

In a CNN, the input connections between S-neuron are variable, and the others are fixed. We use $u_{sl}\left(k_l, n\right)$ to represent the output of an S-neuron on the l level of the k_l S-plane, whereas $u_{cl}\left(k_l, n\right)$ represents the output of the C-neuron of the k_l C-plane. n is a two-dimensional coordinate that represents the location of the field in the input layer. In the first stage, the receptive field has a small area, which then increases in level l. The output of the S-neuron is

$$u_{sl}(k,n) = r_l(k)\Phi\left\{\frac{1+\sum_{k_{l-1}}^{K_{l-1}}\sum_{v\in A_l} a_l(v,k_{l-1},k)u_{cl-1}(k_{l-1},n+v)}{1+\dfrac{r_l(k)}{r_l(k)+1}b_l(k)u_{vl}(n)}-1\right\} \quad (1)$$

$$\Phi\left(x\right) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (2)$$

where $a_l\left(v, k_{l-1}, k\right)$ and $b_l\left(k\right)$ are the connection coefficients of the excitatory input and the inhibitory input, respectively. $r_l\left(k\right)$ is a constant that controls the

selectivity of the feature extraction. A larger value corresponds to less tolerance to noise and feature distortions. $\Phi(x)$ is an nonlinear function. v is a vector that represents the relative position of the former neuron in the receptive field, n. A_l determines the size of the feature extraction of the S neuron, which represents the receptive field of n. Thus, the sum of v also contains all the neurons of the designated area, and the sum of k_{l-1} contains all the sub-planes of the former level. Therefore, the sum term in the numerator is sometimes called the excited term, and is the sum of the product. The neurons are input into the receptive field, and the outputs are multiplied by their corresponding weights. $u_{vl}(n)$ is an assumed inhibitory neuron V, which is located in the S-plane and can be used to show the inhibitory effect of the network. The output of the V-neuron is

$$u_{vl}(n) = (\sum_{k_{l-1}}^{K_{l-1}} \sum_{v \in A_l} c_l(v)(u_{cl-1}(k_{l-1}, n+v))^2)^{\frac{1}{2}} \tag{3}$$

where $c_l(v)$ contains the weights between the V-neurons.

The output of the C-neuron is

$$u_{cl}(k_l, n) = \varphi\left[\frac{1 + \sum_{k_{l-1}=1}^{K_l} j_l(k_l, k_{l-1}) \sum_{v \in D_l} d_l(v)\, u_{sl}(k_l, n+v)}{1 + V_{sl}(n)} - 1\right] \tag{4}$$

$$\varphi(x) = \begin{cases} \dfrac{x}{\beta + x} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{5}$$

where β is a constant. K_l is the number of sub-planes (S) in the l level. D_l is the receptive field of the C-neuron. Therefore, it corresponds with the feature size. $d_l(v)$ is the weight of the fixed excited connection, and it is a monotonic decreasing function of $|v|$. If the k_l sub-plane of the S neuron has received signals from the k_{l-1} sub-plane, $j_l(k_l, k_{l-1}) = 1$, otherwise it is 0.

4. Multi-scale detection model

Multi-scale theory has been increasingly applied to image processing and analysis. Analyzing an object at different image scales increases our understanding of its semantic information. In the real world, object features at different scales can independently exist within a certain space scope. An object has different forms when it is observed at different scales. For example, when observing an object from a distance, we can see large scale features such as contours and shapes. By taking a closer look, we can see the fine structures of the object such as its composition and texture. Thus, it is important to choose an appropriate observation scale for target recognition and understanding.

In computer vision, image representation based on pixel space (pixel scale) can only be applied to some data level processes. We must also extract image features at an appropriate scale. Because the images contain variously sized objects, it is impossible to predefine an optimal scale for analyzing an image. Therefore, it must consider the image content at multiple scales.

As shown in Figure 2, we built a multi-scale feature extraction model that contains three CNNs. Each CNN model consists of eight layers, including five convolution layers and three full connection layers. Features are automatically extracted from each image using three nested and increasingly large rectangular windows (the seat belt region, the windshield region, and the vehicle region). The three features extracted by CNNs are sent to two full connection layers, and the output of the second full connection is sent to the output layer. Finally, we use the linear SVM classifier to classify all the sub-blocks.

We use the CNN to extract features from each of the images. As shown in Fig. 2, we first select candidates for the seat belt region [Figure 2(a)]. In this region, we can directly observe the seat belt information. We extract features for the seat belt region using the CNN model. This vector is called Feature $\boldsymbol{A}$.

If we detect the seat belt region by directly extracting its features, we will have many wrong detection regions. Therefore, we extract a second feature vector from a rectangular neighbourhood to improve the accuracy. This neighbour-

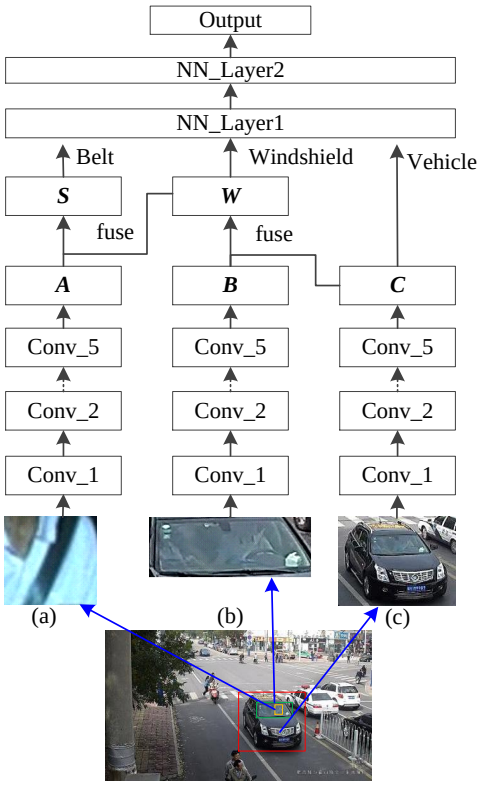

Figure 2. Multi-scale detection model.

hood is the bounding box of the seat belt region and its immediate neighbouring regions, i.e., the windshield region [Figure 2(b)]. This vector is called Feature $\boldsymbol{B}$.

Similarly, using the detected windshield and seat belt regions, we detect the vehicle region [Figure 2(c)]. The vehicle region features extracted by the CNN are called Feature $\boldsymbol{C}$.

We use Feature $\boldsymbol{C}$ to train the vehicle detection model. We combine $\boldsymbol{B}$ and $\boldsymbol{C}$ to get a new feature $\boldsymbol{W}$, which is used to train the windshield detection model. $\boldsymbol{W}$ is defined as

$$\boldsymbol{W} = \nu\boldsymbol{B} + \mu\boldsymbol{C} \tag{6}$$

where ν and μ represent the confidences of $\boldsymbol{B}$ and $\boldsymbol{C}$. The characteristics of human observations mean that the attentions of the objects are different. For example, if we want to detect an object in the image, the object region is set as the target region. A position that is further from the target region has a smaller weight. Thus we set $\nu = 0.7$ and $\mu = 0.3$.

We use the combination of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ (Feature $\boldsymbol{S}$) to train the seat belt detection model. $\boldsymbol{S}$ is defined as

$$\boldsymbol{S} = \alpha \boldsymbol{A} + \beta \boldsymbol{B} + \gamma \boldsymbol{C} \tag{7}$$

where $\alpha = 0.6$, $\beta = 0.3$, $\gamma = 0.1$, which represent the confidences of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$.

5. SVM-based post-processing

In the previous section, we described the algorithm that derives the coarse positions of the vehicle region, windshield region, and seat belt region using the CNN. We calculate their detection scores, and then combine them with the positional relationships among these vehicle components to build the feature vector. Finally, we use the SVM algorithm for post-reprocessing, and eliminate incorrect regions.

SVMs is a classical machine learning method based on statistical learning theory. They were first proposed to solve classification problems. The core idea is to find an optimal hyperplane for classifying different features. The goal is to maximize the classification distance between different types of training samples, and to achieve the best classification.

In this paper, we use three detections (D_1, D_2, D_3) based on the CNN detection algorithm. D_1 represents the vehicle detection result, D_2 is the windshield detection result, and D_3 is the seat belt detection result. $(B : s) \in D_i$ is a 5-dimension feature vector, where $B = (x_1, y_1; x_2, y_2)$. (x_1, y_1) represents the coordinates of the upper left corner of the detection box, and (x_2, y_2) are the coordinates of the bottom right corner. s represents the feature score of each

part. Coordinate B is normalized by the length and width of the candidate picture, which satisfies $(x_1, y_1; x_2, y_2) \in [0, 1]$. As shown in Figure 3, we build a 15-dimension feature vector (D_1, D_2, D_3) for the seat belt region. For the windshield region, we build a 10-dimension feature vector (D_1, D_2). We then use a linear SVM to classify the feature vectors of the seat belt region, the windshield region, and the vehicle region. The training data are the labeled values (D_1, D_2, D_3), which are detected by the CNN algorithm.

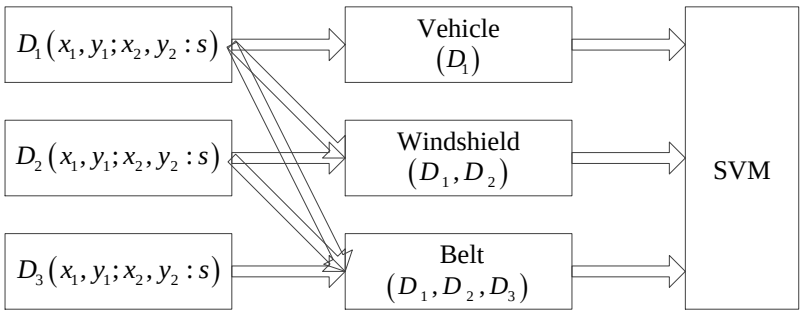

Figure 3. Support vector machine (SVM)-based post-processing.

6. Experimental results

6.1. Dataset

We built two kinds of dataset, a dynamic database and a static database. For the dynamic database, we collected vehicle images from different road environments using road surveillance cameras in conjunction with the traffic administration department. These data included three kinds of vehicles (large, medium, and small). We built a database of vehicles that contained 10000 images, which we used to train the vehicle detection model and the windshield model. We selected 4000 images with good illumination conditions as training samples, which contained 2000 belt images and 2000 non-belt images.

Additionally, we installed cameras at several specific crossings to collect vehicle data for the static database. We selected 200 images of different roads

as test samples, which contain different illuminations and three kinds of vehicles. There were 100 belt images, and 100 non-belt images.

6.2. Vehicle and Windshield Detection

We ran three experiments, including vehicle detection, windshield detection, and seat belt detection. For the vehicle and windshield detection experiments, we evaluated the results using the detected rate (CIR) and undetected rate (MIR), defined as

$$CIR = \frac{N_r}{N_t} \quad and \quad MIR = \frac{N_m}{N_t} \tag{8}$$

where N_r is the number of correctly detected images for the target region. N_m is the number of undetected images. N_t is the total number of tested images, which satisfies

$$N_t = N_r + N_m \tag{9}$$

For the vehicle detection experiment, we selected 6000 vehicle images to train the model. Then, we selected 2000 vehicle images as test images, and randomly divided them into 10 sets. Some examples results are shown in Figure 4(a). The average detection rate was **93.3%**, and the average undetected rate was **6.7%**.

For comparison, the results of the vehicle detection method based on the Adaboost algorithm are shown in Figure 4(b). For the same dataset, the average detection rate was **90.6%**, and the average undetected rate was **9.4%**.

Our algorithm performed better than the Adaboost algorithm, with a smaller detection error and undetected rate. Thus, the vehicle detection window is more suitable for the subsequent windshield detection.

The results of the windshield detection experiments are shown in Figure 5. We selected 6000 windshield images to train the model, and 2000 images for testing, which were randomly split into 10 subsets. We combined the vehicle region features and windshield region features, and input them into the windshield detection system based on deep learning. The average detection rate was

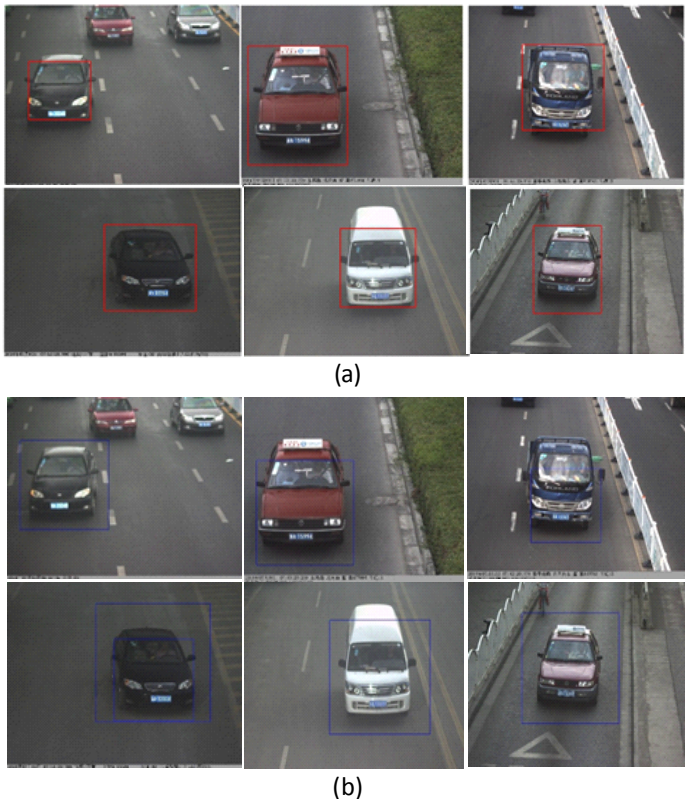

Figure 4. Vehicle detection examples using (a) the deep learning algorithm and (b) the Adaboost algorithm.

93.6%, and the average undetected rate was **6.4%**.

6.3. Seat Belt Detection

For the seat belt detection experiments, we evaluated the system using the *CIR*, false positive rate (*WIR*), and *MIR*, defined as

$$CIR = \frac{N_r}{N_t}, WIR = \frac{N_w}{N_t} \quad and \quad MIR = \frac{N_m}{N_t} \tag{10}$$

where N_r is the number of correctly detected images for the target region. N_w is the number of non-belt images that were incorrectly detected as being

Figure 5. Windshield detection examples using the deep learning algorithm.

belt images. N_m is the number of the belt images that were incorrectly detected as non-belt images. N_t is the total number of test images, which satisfies

$$N_t = N_r + N_w + N_m \tag{11}$$

We selected 4000 seat belt region images to train the model, including 2000 belt images and 2000 non-belt images. We chose 2000 seat belt regions as test images, and split them into 10 subsets. Each subset contained 100 belt images and 100 non-belt images. Combining the seat belt features, vehicle features, and windshield features, we input them into the seat belt detection system based on deep learning. Some example results are shown in Figure 6. The average detection rate was **92.1%**, the average false positive rate was **6.4%**, and the average undetected rate was **2.5%**.

Finally, we applied the seat belt detection method to the static database. The images have a different background to the dynamic database, and were only used to test the method. The training model was the same as before (i.e., from the dynamic database). We selected 200 images with different illuminations and three kinds of vehicles. There were 100 belt images and 100 non-belt images. Table 1 compares the results for the static database using the deep learning and Adaboost algorithms.

Table 1 shows that our algorithm performed better than the Adaboost al-

Figure 6. Seat belt detection examples using the deep learning algorithm.

gorithm, with a higher detection rate, and lower false positive and undetected rates. Furthermore, according to the results of our cross validation, the detection rate of our algorithm only declined slightly in complex backgrounds and illuminations, showing that it is robust to environmental conditions.

6.4. SVM Post-processing for Seat Belt Detection

The previous subsections described the results of the coarse positioning of the vehicle, windshield, and seat belt candidate regions. However, when we need a high detection rate we also get a lot of false detections. We can use the positional relationship of the vehicle components combined with the detection scores to build a feature vector. Then, we use the SVM algorithm for post-processing.

We ran the experiments using different combinations. The results are shown in Figure 7. The curves represent the detection results. The further the curve is from the coordinate axes, the better the result. $'WD + DD + SD'$ represents a

Table 1. Comparison of Seat Belt Detection Results (%)

	CIR	*WIR*	*MIR*
Detection based on deep learning algorithm	80	12	8
Detection based on Adaboost algorithm	73	15	12

15-dimension feature vector consisting of the outputs of the vehicle detector, the windshield detector, and the seat belt detector. This vector was used to train and test the SVM classifier. $'DD + SD'$ represents the 10-dimension feature vector from the windshield and seat belt detectors, and $'SD'$ represents the 5-dimension feature vector from the seat belt detector.

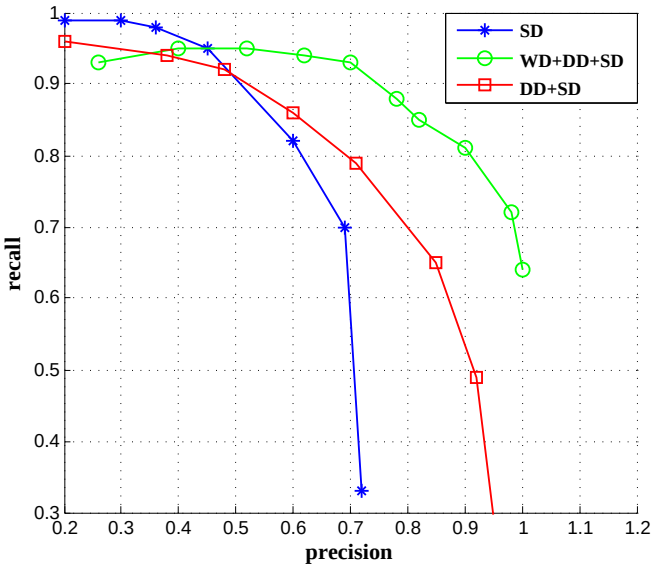

Figure 7. Windshield detection examples using the deep learning algorithm.

Table 2. Comparison of Seat Belt Detection Results (%)

	CIR	WIR	MIR
Deep learning + SVM-based post-processing	87	9	4
Deep learning	80	12	8

Figure 7 shows that the SVM post-processing step is most effective when using all the results from the three detectors to form a 15-dimension feature. Thus, we should use the relative positions of these vehicle components combined with the detection scores to build the feature vector for the SVM post-processing. This significantly reduced the false positive and undetected rates.

Therefore, we chose this combination for the SVM post-processing step.

Table 2 shows the results based on deep learning with and without the SVM-based post-processing. The SVM post-processing step significantly improved the performance.

7. Conclusions

Seat belt detection in intelligent transportation systems is an important research topic. We propose an efficient seat belt detection system. As for the detection part of our algorithm, we extract features using a multi-scale deep neural network. This derives features that are more suitable for training the detection model. In the classification part, we use SVM post-processing, which improves the robustness of the seat belt detection system. Our method significantly reduced the false positive and undetected rates.

In the future, we will collect more vehicle images from real road environments to further expand the training set. This will refine the vehicle classification standards, and increase the complexity of the background environment. We will also attempt to further improve the real-time performance. We believe that the advantages of our method in terms of its real-time performance make it a feasible approach for practical applications.

Acknowledgements

This work is partially supported by National Nature Science Foundation of China (61105076,61272393, 61322201,61432019), Anhui Province Nature Science Foundation (1408085MKL76),key projects of Anhui Province science and technology plan (15czz02074).

References

[1] H. Guo, H. Lin, S. Zhang, S. Li, Image-based seat belt detection, in: Vehicular Electronics and Safety, IEEE, 2011, pp. 161–164.

[2] L. Zhang, Y. Gao, Y. Xia, K. Lu, J. Shen, R. Ji, Representative discovery of structure cues for weakly-supervised image segmentation, IEEE Transactions on Multimedia 16 (2) (2014) 470–479.

[3] H. Zhang, X. Shang, H. Luan, Y. Yang, T. S. Chua, Learning features from large-scale, noisy and social image-tag collection, in: ACM International Conference on Multimedia, 2015, pp. 1079–1082.

[4] K. Y. Park, S. Y. Hwang, An improved haar-like feature for efficient object detection, Pattern Recognition Letters 42 (1) (2014) 148–153.

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

[6] S. Zhan, Q. Q. Tao, X. H. Li, Face detection using representation learning, Neurocomputing 187 (2015) 19–26.

[7] B. Hariharan, P. Arbelez, R. Girshick, J. Malik, Simultaneous detection and segmentation, Lecture Notes in Computer Science 8695 (2014) 297–312.

[8] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems 25 (2) (2012) 1097–1105.

[9] L. Zhang, Y. Gao, R. Hong, Y. Hu, R. Ji, Q. Dai, Probabilistic skimlets fusion for summarizing multiple consumer landmark videos, IEEE Transactions on Multimedia 17 (1) (2015) 40–49.

[10] M. Wang, X. Liu, X. Wu, Visual classification by l1-hypergraph modeling, IEEE Transactions on Knowledge Data Engineering 27 (9) (2015) 2564–2574.

[11] G. Y. Song, K. Y. Lee, J. W. Lee, Vehicle detection by edge-based candidate generation and appearance-based classification, in: Intelligent Vehicles Symposium, IEEE, 2008, pp. 428–433.

[12] B. Sun, S. Li, Moving cast shadow detection of vehicle using combined color models, in: Pattern Recognition, IEEE, 2010, pp. 1–5.

[13] W. Li, J. Lu, Y. Li, Y. Zhang, J. Wang, H. Li, Seatbelt detection based on cascade adaboost classifier, in: International Congress on Image and Signal Processing, Vol. 2, IEEE, 2013, pp. 783–787.

[14] A. Khammari, F. Nashashibi, Y. Abramson, C. Laurgeau, Vehicle detection combining gradient analysis and adaboost classification, in: Intelligent Transportation Systems Conference, IEEE, 2005, pp. 66–71.

[15] H. Grabner, C. Beleznai, H. Bischof, Improving adaboost detection rate by wobble and mean shift, in: Proceedings of Computer Vision Winter Workshop, Vol. 5, 2010, pp. 23–32.

[16] Y. Gao, F. Gao, Edited adaboost by weighted knn, Neurocomputing 73 (16-18) (2010) 3079–3088.

[17] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, D. G. Lowe, A boosted particle filter: Multitarget detection and tracking, in: Computer Vision, Springer, 2004, pp. 24–39.

[18] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, D. Haussler, Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics, 16 (10) (2000) 906–914.

[19] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, The Journal of Machine Learning Research, 2 (2002) 45–66.

[20] S. Tong, E. Chang, Support vector machine active learning for image retrieval, in: Proceedings of the ninth ACM international conference on Multimedia, ACM, 2001, pp. 107–118.

[21] G. Li, Y. Yu, Visual saliency based on multiscale deep features, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015.

[22] Q. Guo, F. Wang, J. Lei, D. Tu, G. Li, Convolutional feature learning and hybrid cnn-hmm for scene number recognition, Neurocomputing 184 (2015) 78–90.

[23] L. Zhang, Y. Yang, M. Wang, R. Hong, L. Nie, X. Li, Detecting densely distributed graph patterns for fine-grained image categorization., IEEE Transactions on Image Processing 25 (2) (2015) 553–565.

[24] C. Luo, B. Ni, S. Yan, M. Wang, Image classification by selective regularized subspace learning, IEEE Transactions on Multimedia 18 (1) (2016) 40–50.

[25] M. Wang, Y. Gao, K. Lu, Y. Rui, View-based discriminative probabilistic modeling for 3d object retrieval and recognition., IEEE Transactions on Image Processing 22 (4) (2013) 1395–1406.

[26] M. Wang, W. Li, D. Liu, B. Ni, J. Shen, S. Yan, Facilitating image search with a scalable and compact semantic mapping, IEEE Transactions on Cybernetics 45 (2015) 1561–1574.

[27] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, in: International Conference on Machine Learning, IEEE, 2013.

[28] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. FeiFei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.

[29] A. S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: an astounding baseline for recognition, in: Computer Vision and Pattern Recognition Workshops, IEEE, 2014, pp. 512–519.