Research Article

Intelligent Detection of Vehicle Driving Safety Based on Deep Learning

Deyun Wang

Changde Vocational Technical College, Changde 415000, China

Correspondence should be addressed to Deyun Wang; wangdeyun1964@nuaa.edu.cn

Received 9 March 2022; Accepted 9 May 2022; Published 17 June 2022

Academic Editor: Deepak Kumar Jain

Copyright © 2022 Deyun Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the continuous increase in the number of cars, traffic safety problems are also becoming more and more serious, whether the driver wears a seat belt to protect the driver’s personal safety so that the problem can be solved in the event of a traffic accident. The author puts forward the research status of deep learning and convolutional neural network, as well as its theory and technology, and conducts in-depth analysis and research, a small target detection algorithm Deconv-SSD based on transposed convolution is proposed, driver area localization algorithm Squeeze-YOLO is based on lightweight model, and driver seat belt detection algorithm is based on semantic segmentation. Deconv-SSD achieves fast vehicle detection through depthwise separable convolution and fusion of multiresolution feature maps and then utilizes the salient features of the front windshield; through the method of lightweight feature extraction and the Squeeze-YOLO algorithm, the rapid positioning of the driver area is realized. Fast segmentation of seat belts is based on semantic segmentation algorithm and pruning technology in the positioning area, and by judging the maximum connected domain area after segmentation, the driver’s seat belt detection is realized. Experiments and data analysis are carried out on the proposed algorithm. When the image resolution is consistent with the feature extraction model, the average accuracy of Deconv-SSD is compared with the original SSD algorithm in the PASCALVOC public dataset, from 77.2% to 79.6%. In the self-made seat belt detection dataset, Squeeze-YOLO can reach 73 FPS when the average accuracy is 99.96%, the semantic segmentation algorithm accelerated by pruning achieves an accuracy of 94.87% at a speed of 305 FPS, and the validity of the experiment is verified.

1. Introduction

According to the statistics published in the 2014 Statistical Bulletin of National Economic and Social Development, China’s civilian car ownership hit a record high at the end of 2014, totaling 154.48 million, an increase of 12% over the end of 2013. Among them, there are 83 million civil sedans in China, an increase of 17% over 2013. There were 75.9 million private sedans, an 18% increase from 2013 [1]. Although in 2014, the number of deaths per 10,000 vehicles in China’s road traffic accidents was 5% lower than that at the end of 2013, but from the above data, we can know that, compared with 2013, the number of civilian vehicles in 2014 increased significantly, so actually the number of deaths due to traffic accidents in 2014 increased compared to the number of deaths in 2013, and the growth rate is 9%. Seat belts are a very important passive protection measure, and it can effectively reduce the speed of the vehicle when driving on the road, the rate of casualties due to vehicle collisions, or other traffic accidents. In a traffic accident, the risk of an occupant who does not wear a seat belt correctly is only 0.4 of the occupant who wears a seat belt correctly. Moreover, according to Article 51 of the Regulations of the People’s Republic of China on Human Rights and Safety, during driving, wearing seat belts correctly is a traffic law that drivers must abide by. In the form of regulations, all countries in the world require car manufacturers to install the safety belt reminder (SBR) system, thereby improving the seat belt wearing rate as shown in Figure 1 [2]. In 2009, China also included the seat belt wearing reminder system as a bonus item for car safety and wrote it into the C-NCAP management rules, in 2010, the national standard for the safety belt wearing reminder system was implemented, and the definition, indication signals, and technical...
requirements of the safety belt wearing reminder system were clearly stipulated. However, the wearing rate of seat belts in China is generally low, mainly due to the lack of safety awareness of occupants, and there are various irregularities in evading the seat belt reminder system. Common irregularities include the following: drivers put on the seat belt before getting in the car and sit directly on the seat with the seat belt on after getting in the car, instead of wearing the seat belt correctly on the chest or using a separate buckle, causing the vehicle’s own seat belt reminder system to fail. According to a survey, in Taobao shopping malls, the monthly sales of seat belt buckles exceed 9,000 pieces, and the car supplies section of major supermarkets are also sold [3]. Therefore, study the detection method of whether the driver in the motor vehicle wears the seat belt, and it is of great significance to improve the driver’s awareness of obeying traffic laws. In traditional seat belt detection, the features of seat belts are usually designed manually according to the geometric features in the sample image, but the cost of manual design is relatively high, and the versatility is poor, and the emergence of deep learning (deep learning) solves the above problems. In deep learning, by continuously combining shallow information, a more abstract high-level representation is finally obtained, thereby uncovering potential patterns in the data. The essence of deep learning is an unsupervised pretraining algorithm, features can be automatically learned from samples, which reduces the complexity of manually designing features, and at the same time, the impact of human intervention can be minimized. In recent years, deep learning has gradually become a hot field in machine learning research, and it has been applied in some research fields, but no relevant research has applied it to seat belt detection; therefore, the author’s research based on the deep learning seat belt detection method not only has certain practical value but also has important reference significance [4].

2. Literature Review

As the number of cars in China continues to grow, road safety issues are becoming more and more serious. Muhammad et al. observed that in the event of a car accident, the driver wears the seat belt to protect the life and safety of the driver; however, some drivers have poor safety awareness, when driving without wearing a seat belt [5]. At present, in order to urge drivers to wear seat belts by Wan et al., the traffic police department manually analyzes the monitoring images, and checking the driver’s wearing of seat belts is a low-efficiency and high-cost manual inspection. With the development of computer vision, the detection of seat belts through image recognition has become an inevitable trend [6]. According to Wang et al., considering that image recognition is one of the main researches in the field of computer vision, the image recognition model uses a classifier such as histogram of gradients (HOG) to describe, recognize, and support images by creating a SVM (support vector machine) [7]. However, Xie et al. believe that the robustness of traditional handcrafted features such as HOG is poor, and the feature extraction ability is insufficient in complex application scenarios. With the excellent performance of deep learning in the field of image recognition, the convolutional neural network has been widely used in various fields related to images. In many domains, such as image segmentation, material exploration, and semantic segmentation, convolutional neural network algorithms have become the most accurate algorithms available today. Convolutional neural networks can automatically extract features through training and have stronger feature fitting capabilities [8]. However, according to Xie et al., this indicates that convolutional neural networks rely heavily on convolution counts and the algorithm is slow. As computer technology continues to advance, computing speeds
continue to increase, especially due to improvements in graphics processing units (GPUs), which are relatively slow [9]. Parallel counting solves the problem of slow neural network execution. And deep learning algorithms represented by convolutional neural networks have received new attention. In summary, the author combines computer vision and deep learning techniques, through the research and development of seat belt detection algorithm, the automatic detection of seat belt drivers in traffic monitoring is realized, replacing the inefficient manual detection method, and it has important practical significance for improving drivers’ awareness of safe driving and improving traffic safety. The early target detection algorithm mainly screened the geometric and color features of the target, and there are problems such as large limitations, poor robustness, and low accuracy. In 2020, Nie et al. proposed convolutional neural network and BP backpropagation; however, convolutional neural networks are uninterpretable and belong to correlation models rather than causal models, and at the same time, limited by the performance of computer hardware equipment at that time, it did not receive extensive attention at that time [10]. In 2021, Xun et al. proposed the support vector machine (SVM). The SVM classifier is suitable for small sample data sets and is a linear classifier [11]. When solving nonlinear classification problems, a low-dimensional linear inseparable problem is transformed into a high-dimensional linearly separable problem through a high-dimensional mapping. However, with high-dimensional data, images are not suitable for classification using SVM directly, and abstract features need to be extracted from images before detection. In 2020, Han et al. proposed the local binary pattern (LBP) feature descriptor, the texture features of images can be extracted, and LBP has the advantages of rotation and grayscale invariance [12]. In 2019, French researcher Geng et al. proposed the histogram of directional gradient HOG feature descriptor, which constitutes features by calculating and counting the gradient direction histogram of local areas of the image [13]. However, there is a problem of low efficiency in the form of combining HOG or LBP with SVM. Because when the detection of multiscale objects gradually increases with the parallel computing power of GPUs, convolutional neural networks have regained the attention of researchers. In 2012, the deep learning algorithm AlexNet won the Imagenet championship with an error rate of 16.4%, surpassing the algorithm that obtained an error rate of 28.2% using handcrafted features such as HOG in 2010. With the advent of Alexnet, image recognition using convolutional neural networks has become mainstream. In 2013, ZFNet won the championship with an error rate of 11.7%. In 2014, Google proposed Googlenet and won the ImageNet championship with an error rate of 6.7%, and it is also widely used in object research, from convolutional neural network improvements to image distribution. In the study of Nguyen et al., it was applied to RCNN in 2014 and achieved 53.7% map (average accuracy) on the PASCAL VOC dataset, while the DPM algorithm using hardware such as HOG has only 35.1% accuracy [14].

3. Methods

The principle of convolutional neural network is similar to that of neural network, and the weights are modified by BP reexpansion. However, the convolutional neural network replaces the traditional neuron with a convolution kernel, and its convolution kernel contains multiple convolution templates, and multichannel convolution is performed on the feature map. By designing the final loss function, find the deviation between the output target value of the convolutional neural network and the true value, and adjust the weights in the convolution kernel by using the chain partial derivative of BP backpropagation, in order to improve the feature fitting ability. A convolutional neural network usually consists of three stages, namely, the convolution process, the pooling layer, and the activation layer. The convolutional layer is the main structure for extracting features from the convolutional neural network. Each convolutional neural network consists of multiple convolutional layers, and each convolutional layer contains multiple convolutional kernels. The number is designed by the developer according to the actual engineering application. Each convolution kernel has multiple convolution templates, and the number of convolution templates is the same as the number of input feature map settings. The structure of the convolutional layer is shown in Figure 2 [15].

With the development of convolutional neural networks, in addition to the traditional multichannel convolution, the convolution process has also been adapted to various types of convolutions, such as transform exchange [16]. The convolution template of variable convolution is no longer limited to rectangles but completes feature extraction in the form of irregular discrete points, in order to achieve the effect of changing the convolution receptive field. The feature extraction ability is improved by using the convolution of the receptive field. The pooling layer is responsible for the downsampling operation in the convolutional neural network, in order to achieve the effect of increasing the receptive field of convolution and reducing the resolution of the input feature map. In convolutional neural networks, low-resolution feature maps carry high semantic information, while high-resolution feature maps carry more object details. The reduction of feature map resolution can be controlled by using the pooling layer, in order to achieve the purpose of extracting different levels of features [17]. The pooling layer is mainly divided into average pooling, max pooling, and global pooling. Average pooling is to average the corresponding feature maps in the pooling layer to output a feature value, the maximum pooling is to take the largest feature value of the corresponding feature map area, and the global pooling directly averages all the values of the feature map to output a value. The activation layer is generally after the convolutional layer, and the activation layer can perform nonlinear mapping on the feature map, increasing the nonlinear fitting ability of the convolutional neural network. Commonly used functions can be divided into ReLU function, sigmoid function, and tanh function. Among them, the ReLU activation function is more commonly used, and the ReLU activation ability increases the nonlinear
absorption capacity of the network. Deepening the convolutional layer will improve the feature extraction ability of the network, so the current design has more convolutional layers. Deeper network structures are prone to gradient disappearance or gradient explosion during training, so convolutional neural networks have introduced batch normalization regularization. Batch normalization normalizes the mean and variance of each dimension of the input feature, as shown in Equation (1). By modifying the output of each layer of the network, the output satisfies the Gaussian distribution to prevent the disappearance of gradient or the phenomenon of gradient explosion [18].

\[ x^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{\text{var}[x^{(k)}]}}. \] (1)

Among them, \( x^{(k)} \) is the input feature, \( E[x^{(k)}] \) is the expectation of the input feature, \( \text{var}[x^{(k)}] \) is the variance of the input feature, and \( x^{(k)} \) is the corrected output feature. Batch normalization regularization performs linear transformation on the output features after forced correction, the parameters of the linear transformation can be adjusted during the training of the convolutional neural network, while satisfying the Gaussian distribution, and the output features retain the original information as much as possible, as shown in the following formula:

\[ y^{(k)} = r^{(k)}x^{(k)} + \beta^{(k)}. \] (2)

Among them, \( y^{(k)} \) is the output after linear transformation, and \( r^{(k)} \) and \( \beta^{(k)} \) are determined by BP backpropagation during the training process, and the initialization method of the weights of each convolutional layer will also affect the training time and training effect of the network. As the depth of the network increases, the also inconsistency increases during the convolution process. There are three ways to initialize the convolution weights, random number initialization, Havel initialization, and MSRA initialization for Gaussian distribution. In deeper convolutional neural networks, initializing the convolution weights with random numbers is prone to gradient dispersion or gradient explosion. For example, Gaussian divides the random number by the mean 0, and the difference 1 is used for initialization, as the number of layers further increases in the forward representation process, the opposite pattern law, finally, it approaches 0, resulting in all outputs being completely equal, and the overall network structure loses its vitality. Javier initialization is an initialization method proposed by Xavier in 2010. Javier initialization associates the initialization of the weight with the number of neurons in the layer where the weight is located, and the number of neurons should be inversely proportional to the value of the weight initialization. Havel started to suffer from gradient dispersion issues when convolutional neural networks used the ReLU activation function. When the input voltage of the ReLU function is less than 0, truncation occurs, so the weight difference still disappears [19].

MSRA initialization is based on the Havel initialization that the number of original neurons in this layer is only half of the real number; therefore, the truncation of the input less than 0 by the ReLU activation function is prevented. MSRA initialization can be expressed as shown in formula (3), where \( W \) represents the weight, \( G \) represents the Gaussian distribution, and \( n \) represents the number of network neurons.

\[ W = G \left[ 0, \sqrt{\frac{2}{n}} \right] \] (3)

The convolutional neural network expands the computational layers such as the convolution kernel and the pooling kernel into a sparse matrix, and convert convolution and pooling computations into matrix multiplication computations with feature maps. Using the characteristics of multiple data streams for one instruction of GPU, the multiplication of two matrices is performed in parallel, and each thread calculates the multiplication of the corresponding elements in the two matrices. The computing power of each thread of the GPU is weaker than that of the CPU, but the amount of computation required by each thread of the convolutional neural network is not large, and the high degree of parallelism of GPU is very suitable for convolutional neural network computing. If the CPU is used for computing, although each thread has strong computing power, it will take more time to calculate the convolutional neural network serially [20]. Usually the deep learning open source framework initializes the weights by the CPU and expands the weights and feature
maps into sparse matrices; the prepared feature maps and weights are then sent to the GPU for parallel computing acceleration. When the GPU calculation is complete, the weights are transferred back to the CPU’s memory. The GPU and the CPU complete the data interaction between the GPU memory and the CPU memory through the PCIE interface of the motherboard. The convolutional neural network algorithm designed by the author uses NVIDIA’s GPU and CUDA environment for training and development, and the process of using GPU to train the network structure is shown in Figure 3.

At present, the deep learning open source frameworks convert convolution calculations into matrix calculations, and when GPU acceleration of matrix multiplication is performed, the matrix acceleration library Cublas developed by CUDA is generally used. Using Cudnn in addition to Cublas will also significantly increase the training speed of convolutional neural networks, and Cudnn is a computational acceleration library for deep learning developed by NVIDIA. By comparing the convolutional layers in the convolutional neural network, the optimization of common layers such as pooling layers and fully connected layers achieves the acceleration effect. Convolutional neural networks can be used as nonlinear classifiers when using the cross entropy loss function, and its classic application scenario as a classifier is handwritten digit recognition. In the field of handwritten digit recognition, a common algorithm is the Lenet-5 convolutional neural network proposed by Y. LeCun in the 20th century. Taking handwritten digit recognition as an example, the convolutional neural network is improved by means of continuous asymmetric convolution; introduce the application of convolutional neural network in image classification. The principle of continuous asymmetric convolution is similar to the matrix low-rank decomposition, with \(1 \times n\) and \(n \times 1\) as a set of feature extraction modules, this set of feature extraction modules has only \(2 \times n\) parameters, and an \(n \times n\) convolution kernel requires \(n \times 2\) parameters, and continuous asymmetric convolution can reduce the amount of parameters while achieving the same feature fitting effect. Continuous asymmetric convolution has a certain limitation on feature map resolution, when the resolution of the feature map is too high, the continuous asymmetric convolution structure is used, which is easy to cause serious information loss, when the feature map resolution is the smallest, the acceleration effect of the continuous asymmetric convolution model is not obvious. It is generally assumed that continuous asymmetric convolutional models are designed for feature maps with resolutions between \(10 \times 10\) and \(30 \times 30\). Because in the handwritten digit recognition task, the resolution of the input image is low, and the size of each feature map of the convolutional neural network is distributed from \(10 \times 10\) to \(30 \times 30\), so it is suitable to use a continuous asymmetric convolution structure. Combined with the actual needs of handwritten digit recognition resolution, a convolutional neural network structure is designed, and its feature extraction network structure parameters are shown in Table 1.

Based on the proposed network structure, comparative experiments are carried out using the MNIST handwritten digit recognition dataset. The MNIST dataset includes 10 digits from 0 to 9 and is a widely used standard dataset in the field of handwritten character recognition, it contains 60,000 images in the training set and 10,000 images in the test set, and the resolution of each image is \(28 \times 28\). In the hardware platform used by the author, the CPU is Intel(R) Core i5-4570, and the GPU is NVIDIA(R) GTX 1050TI. Utilize Caffe deep learning open source framework and CUDA parallel computing library. The designed continuous asymmetric convolution structure is compared with Lenet-5 on the MNIST dataset, and the experimental results are shown in Table 2 [21].

The comparative analysis of the experimental results shows that, the continuous asymmetric convolution structure proposed by the author compares with Lenet-5, the recognition accuracy and recognition speed have been improved, and it is also verified that the same network structure is used in the case of CUDA and Cudnn, and with the help of the parallel computing power of the GPU, the acceleration effect can be achieved.

Convolutional neural networks can achieve object detection tasks when using both regression and classification loss functions, and at the same time, the position and category of the object to be detected in the image are obtained. This paper mainly introduces YOLO and SSD two target detection algorithms and their applications. YOLO performs target detection in a regression way, and fast target detection can be achieved with a small amount of parameters and a small amount of calculation. In 2016, Joseph Redmon and others proposed the target detection algorithm YOLOV1 and then improved YOLOV2 and YOLOV3 on the basis of YOLOV1. The YOLOV1 network structure is shown in Figure 4.

YOLOV1 divides the feature map of the last layer into \(7 \times 7\) grids, and each grid assumes two candidate boxes. The candidate frame to be regressed is selected by judging the degree of overlap between the two candidate frames and the real value. The values to be regressed for each candidate box are the deviation relative to the upper left corner of each grid and the width and height relative to the image. In addition to the regression location information, the information of the target category and its confidence is also obtained through the cross-entropy loss function. The feature map of the last layer of YOLOV1 is output in the form of fully connected feature vector, and when the number of fully connected parameters is too large, it can be replaced by \(1 \times 1\) convolution [22]. On the basis of YOLOV1, YOLOV2 adds batch normalization regularization to the convolutional layer of feature extraction, introduces the concept of anchor in the loss function, improves the network to a fully convolutional structure, and increases the input image resolution. YOLOV2 introduces the anchor mechanism in the loss function, an anchor is a candidate frame with no center point and only an aspect ratio, and its aspect ratio is obtained by clustering the objects to be detected in each category in the training set. However, the introduction of the anchor mechanism did not improve the accuracy of YOLOV2 too much, and the improvement of the accuracy of YOLOV2 mainly depends on improving the resolution of the feature map and adding batch normalization regularization. YOLOV2
still only performs classification and regression on the last layer of feature maps, and in order to further improve its accuracy, YOLOV3 performs multiscale regression on the basis of YOLOV2. YOLOV3 regresses anchors on multiple feature maps of different resolutions, thereby improving the detection ability of multiscale targets and achieving the effect of improving accuracy. Similar to YOLOV3, in 2016, Liu et al. proposed the target detection algorithm SSD. SSD takes candidate boxes of different sizes for feature maps of six layers with different resolutions. The six-layer feature map is trained by the loss function to obtain the class confidence and the position of the candidate box, and the deviation between the real value and the SSD network structure is shown in Figure 5 [23].

SSD designs anchors with different numbers and sizes for feature maps of different resolutions, and the size of each anchor is determined by the size of the feature map. The length and width of the anchor in the SSD are the values calculated according to the resolution of the feature map of each layer, and the center coordinate is the center point that maps the feature map back to the original image. SSD outputs two convolutional layers for the feature map of each resolution and uses the two convolutional layers to output the location and category information of the feature map of the resolution.

4. Experiments and Discussions

Relevant literature on driver wearing seat belt detection has appeared since 2013. At present, the literature mainly realizes seat belt detection through the idea of image classification. Whether a driver wears a seat belt is essentially a classification problem, and seat belt detection can be completed through a classification algorithm. A higher detection accuracy can be achieved when the driver wears a seat belt with obvious features. When training an image classifier, the classifier realizes the classification of images by learning the feature difference between positive and negative samples. In practical applications, the driver’s driving scene is complex, and whether the driver wears a seat belt cannot be the main difference between positive and negative samples. The main difference between positive and negative samples is not seat belts, which are more common, such as driver’s clothing, interior decoration, and driving posture. These are more distinct features, and it is easy to provoke whether

---

**Table 1: Network structure parameters.**

<table>
<thead>
<tr>
<th>Type of layer</th>
<th>Size</th>
<th>Number of neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolutional layer</td>
<td>1 × 3</td>
<td>32</td>
</tr>
<tr>
<td>Convolutional layer</td>
<td>3 × 1</td>
<td>32</td>
</tr>
<tr>
<td>Convolutional layer</td>
<td>1 × 3</td>
<td>64</td>
</tr>
<tr>
<td>Convolutional layer</td>
<td>3 × 1</td>
<td>64</td>
</tr>
<tr>
<td>Dropout (p = 0.2)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Max pooling layer</td>
<td>2 × 2</td>
<td>-</td>
</tr>
<tr>
<td>Convolutional layer</td>
<td>3 × 3</td>
<td>128</td>
</tr>
<tr>
<td>Convolutional layer</td>
<td>3 × 3</td>
<td>128</td>
</tr>
<tr>
<td>Dropout (p = 0.4)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Max pooling layer 2 × 2</td>
<td>2 × 2</td>
<td>-</td>
</tr>
</tbody>
</table>

---

Figure 3: GPU training process of convolutional neural network.
the driver wears a seat belt or not, which cannot be the main distinguishing feature of the positive and negative samples. If the seat belt detection is realized by means of target detection, since the seat belt belongs to the target of irregular geometric shape, the target detection boxes are all rectangular boxes, so a large number of nonseat belt areas will be introduced in the training set labeling, and whether the driver wears seat belts still cannot be the main difference in the training set. Semantic segmentation performs pixel-level classification of images, it can realize the detection of irregularly shaped targets, so it is necessary to expand the segmentation results and connect multiple connected domains and increase the robustness of the algorithm. Finally, the maximum connected domain area after expansion is calculated, and set the maximum bounded area to determine whether the driver is wearing a seat belt. The algorithm flow is shown in Figure 6.

When using the semantic segmentation network structure for feature extraction, the edge accuracy mainly depends on the decoding part, and it uses transposed convolution to achieve upsampling to restore the resolution of low-resolution feature maps with high semantic information. The higher the number of transposed convolutional layers, the higher the edge quality. Feature maps with higher

<table>
<thead>
<tr>
<th>Network type</th>
<th>Test speed</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous asymmetric convolution+CUDA</td>
<td>0.0048 s</td>
<td>99.5%</td>
</tr>
<tr>
<td>Lenet-5+CUDA</td>
<td>0.2322 s</td>
<td>99.1%</td>
</tr>
<tr>
<td>Lenet-5+CPU</td>
<td>0.4688 s</td>
<td>99.0%</td>
</tr>
</tbody>
</table>

Table 2: MNIST dataset comparison experiment.

![Figure 4: YOLOV1 network structure.](image)

![Figure 5: SSD network structure.](image)
downsampling ratios contain higher semantic information, but their resolutions are lower. The essence of the transposed convolution is to calculate the parameters through the interpolation adjusted by the training. With only a small number of transposed convolutions, directly interpolating low-resolution feature maps back to feature maps of the original image resolution will result in errors in edge details. If multistep interpolation is used, the sampling rate of each interpolation is low, and at the same time, the feature information of the same resolution feature map in the downsampling stage can be fused to obtain more detailed edge accuracy. Too many transposed convolutions will increase the amount of computation, resulting in slower detection speed, which does not meet real-time requirements. In order to reduce the influence of edge accuracy on the detection results, the author determines whether the driver wears a seat belt by judging the maximum connected domain area after semantic segmentation. In this way, the accuracy of the edge of the semantic segmentation result will not affect the accuracy of seat belt detection. In order to enhance the feature extraction capability of the network, multilayer convolution is used in the encoding part. At the same time, to speed up the calculation, only the single-layer transposed convolution design is retained in the decoding part. The edge quality is sacrificed to a certain extent, but the accuracy of seat belt detection is not affected. The mainstream innovations in Deeplab and Pspnet are the current mainstream improvement methods of semantic segmentation algorithms. Both aspp and psp depend on the front-end feature extraction model to have a strong feature extraction capability and a large amount of parameters to be effective. If the front-end feature extraction capability is strengthened, the speed of the algorithm will be greatly reduced; therefore, the semantic segmentation algorithm under this improved method requires a large amount of computation, which does not meet the real-time application requirements of driver seat belt detection. The purpose of multiresolution feature fusion algorithms like aspp and psp is to enhance the feature extraction capability of multiscale objects. However, the difficulty of seat belt detection is not that the size of the seat belt is small, but that the characteristics of the seat belt itself are not obvious. The author does not pay attention to the edge accuracy of seat belt segmentation; therefore, the author did not use the current mainstream improvement techniques for semantic segmentation algorithms; instead, a simple algorithm structure similar to 32-fold upsampling single-layer transposed convolution is directly used, and the network structure is shown in Figure 7.

The number of convolution kernels in the transposed convolution layer depends on the number of categories. Since the author performs seat belt semantic segmentation, it is only divided into background class and seat belt class, so the transposed convolution layer has two convolution kernels. The two output feature maps are of the same resolution as the input image, and whether the pixel is a seat belt area is determined by the confidence of the same pixel position in the two feature maps. Although a single-layer transposed convolution is used, there are still a large number of convolutional layers in the front-end coding part of the semantic segmentation network, and its calculation speed is still slow. In order to meet the real-time detection requirements, it also needs to be accelerated design. Although seat belt detection is a binary classification task, the seat belt features are not obvious in complex environments; therefore, in the feature extraction part of the

![Figure 6: Semantic segmentation detection seat belt algorithm flowchart.](image-url)
seat belt semantic segmentation algorithm, no lightweight design is used. Semantic segmentation is a fully convolutional network structure, that is, all weights are convolutional layers and do not contain fully connected layers. Although the number of weights of convolutional layers is smaller than that of fully connected layers in other networks, the computational cost of convolutional layers is higher than that of fully connected layers. The reason why the multipass convolution calculation of the convolution layer is slow is that there are multiple convolution templates in each convolution kernel. The convolution template performs convolution calculation on the feature map, respectively, and traverses the whole image, so the calculation speed of the convolution layer is slow. When the convolution template convolves the whole image, its weight is unchanged, and the weight parameters are shared, so the amount of parameters is small. The author’s main task to speed up the network computing speed is to accelerate the convolutional layer. The weight values of some convolution kernels in the convolutional neural network are close to zero. When the convolution calculation is performed, the part with the weight value closer to zero has less influence on the final result, but this part of the weight participates in the calculation and takes up the calculation time. If it is not involved in the calculation, the calculation speed of the convolutional neural network will be accelerated, and at the same time, it does not affect its feature extraction effect. The researchers propose to use pruning method to accelerate the processing of convolutional neural network; pruning is to delete the redundant convolutional weights that have less influence on the network feature extraction ability. Reducing the number of parameters in the network can not only speed up the inference speed of the network but also improve the generalization ability of the network, achieve the effect of regularization, and prevent overfitting. If only the weights whose weights are close to zero are deleted, although the calculation of the weights will be reduced, the memory access lines will become lower due to the sparseness of the weights, which is not conducive to the optimization of the cache. Some researchers have proposed an accelerated way to delete the entire convolution kernel, except that a certain convolution kernel is compared with a certain weight in the deleted convolution kernel, directly delete the entire convolution kernel, which can be in the existing convolutional neural network computing framework, and do not make any changes to the frame to achieve a direct acceleration effect. Directly delete the convolution kernel, and it is also convenient to use the existing acceleration library after pruning, and the pruned convolutional neural network is further accelerated.

In the front-end feature extraction node of the pruned semantic segmentation network structure, the adhesive layer parameters are shown in Table 3.

<table>
<thead>
<tr>
<th>Convolutional layer</th>
<th>Convolution kernel size</th>
<th>Number of convolution kernels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1-1</td>
<td>3 × 3</td>
<td>64</td>
</tr>
<tr>
<td>Conv1-2</td>
<td>3 × 3</td>
<td>22</td>
</tr>
<tr>
<td>Conv2-1</td>
<td>3 × 3</td>
<td>49</td>
</tr>
<tr>
<td>Conv2-2</td>
<td>3 × 3</td>
<td>62</td>
</tr>
<tr>
<td>Conv3-1</td>
<td>3 × 3</td>
<td>110</td>
</tr>
<tr>
<td>Conv3-2</td>
<td>3 × 3</td>
<td>118</td>
</tr>
<tr>
<td>Conv3-3</td>
<td>3 × 3</td>
<td>141</td>
</tr>
<tr>
<td>Conv4-1</td>
<td>3 × 3</td>
<td>233</td>
</tr>
<tr>
<td>Conv4-2</td>
<td>3 × 3</td>
<td>256</td>
</tr>
<tr>
<td>Conv4-3</td>
<td>3 × 3</td>
<td>302</td>
</tr>
<tr>
<td>Conv5-1</td>
<td>3 × 3</td>
<td>398</td>
</tr>
<tr>
<td>Conv5-2</td>
<td>3 × 3</td>
<td>379</td>
</tr>
<tr>
<td>Conv5-3</td>
<td>3 × 3</td>
<td>512</td>
</tr>
</tbody>
</table>

This paragraph proposes the use of semantic segmentation model to segment seat belts, which provides a new idea for the field of seat belt detection. In the network structure, 32 times upsampling FCN is used to complete driver seat belt segmentation, and the driver’s seat belt detection is realized by judging the maximum connected domain area of the segmentation result, which effectively solves the problem that the seat belt is not the main difference between positive and negative samples. In order to improve the running speed of the seat belt semantic segmentation algorithm, in this section, channel pruning is performed on the semantic segmentation model by means of pruning and compression, and redundant convolution kernels are removed. The results show the semantic segmentation detection seat belt proposed by the author achieves a high detection accuracy while achieving high detection accuracy and can meet real-time detection requirements.
Combined with deep learning technology, the author proposes a driver seat belt detection algorithm based on convolutional neural network and designs and implements the system; the Deconv-SSD algorithm was proposed based on multiscale feature fusion and depthwise separable convolution, which improves the accuracy of small target detection. Deconv-SSD fuses feature maps of different resolutions and uses depthwise separable convolution and channel rearrangement as the minimum unit for feature extraction. Comparative experiments are carried out on the PASCAL-VOC dataset, and with the same feature output model and input image of the same resolution as the original SSD algorithm, the average accuracy is improved from 77.2% to 79.6%. The Squeeze-YOLO algorithm was designed by combining the SqueezeNet network structure and the YOLO loss function, and they can effectively detect the front windshield area of the vehicle quickly. Squeeze-YOLO is based on the prior knowledge that the size of the windshield is large and the features are more obvious, when no small target detection is required, fast driver area localization using YOLOV1 loss function. By conducting comparative experiments on the self-made windshield detection dataset, the experimental results show the driver area localization algorithm proposed by the author can reach a speed of 73 frames per second with an accuracy of 99%. The fast detection of seat belts is realized by using semantic segmentation and channel pruning algorithm. The semantic segmentation algorithm directly uses 32 times upsampling deconvolution in the decoding part, ignoring edge accuracy, and only considers the area of the connected domain after segmentation. After judging semantic segmentation based on the threshold, the author implements the safe area detection based on the maximum connected domain area. The semantic segmentation algorithm performs channel pruning in the coding segment to remove unnecessary convolution kernels to achieve algorithm acceleration. Comparative experiments are carried out on the self-made seat belt detection data set, and the speed of the seat belt detection algorithm proposed by the author can reach 305 FPS when the accuracy rate is 94%. Bit weights are faster. Due to time reasons, the author did not integrate the accelerated model of Tensorrt into the interface system but only conducted a comparative experiment with the accelerated model. In addition, the author uses the XILINX open source IP core in the FPGA, and if you use Verilog or Vivado hls to replace it, you may get better performance by making a dedicated IP core according to the algorithm requirements.

Data Availability
The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest
The author declares no conflicts of interest.

References


