applied sciences

MDPI

Article

Seatbelt Detection Algorithm Improved with Lightweight Approach and Attention Mechanism

Liankui Qiu *, Jiankun Rao and Xiangzhe Zhao

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China; jiankun_rao@stu.haust.edu.cn (J.R.); xiangzhe_zhao@stu.haust.edu.cn (X.Z.)
* Correspondence: qiulk@haust.edu.cn

Abstract: Precise and rapid detection of seatbelts is an essential research field for intelligent traffic management. In order to improve the detection precision of seatbelts and speed up algorithm inference velocity, a lightweight seatbelt detection algorithm is proposed. Firstly, by adding the G-ELAN module designed in this paper to the YOLOv7-tiny network, the optimization of construction and reduction of parameters are accomplished, and the ResNet is compressed with the channel pruning approach to decrease computational overheads. Then, the Mish activation function is utilized to replace the Leaky Relu in the neck to enhance the non-linear competence of the network. Finally, the triplet attention module is integrated into the model after pruning to make up for the underlying performance reduction caused by the previous stage and upgrade overall detection precision. The experimental results based on the self-built seatbelt dataset showed that, compared to the initial network, the Mean Average Precision (mAP) achieved by the proposed GM-YOLOv7 was improved by 3.8%, while the volume and the computation amount were lowered by 20% and 24.6%, respectively. Compared with YOLOv3, YOLOX, and YOLOv5, the mAP of GM-YOLOv7 increased by 22.4%, 4.6%, and 4.2%, respectively, and the number of computational operations decreased by 25%, 63%, and 38%, respectively. In addition, the accuracy of the improved RST-Net increased to 98.25%, while the parameter value was reduced by 48% compared to the basic model, effectively improving the detection performance and realizing a lightweight structure.

Keywords: seatbelt detection; YOLOv7; activation function; attention mechanism; channel pruning; lightweight structure

Citation: Qiu, L.; Rao, J.; Zhao, X. Seatbelt Detection Algorithm Improved with Lightweight Approach and Attention Mechanism. *Appl. Sci.* **2024**, *14*, 3346. https://doi.org/10.3390/app14083346

Academic Editor: Stéfano Frizzo Stefenon

Received: 15 March 2024
Revised: 6 April 2024
Accepted: 15 April 2024
Published: 16 April 2024

1. Introduction

In recent times, the number of motor vehicles has increased worldwide, accompanied by an increasing probability of traffic accidents, which are the major cause of accidental fatalities [1]. However, the proper utilization of seatbelts could validly lower the incidence of casualties [2]. Although buckling up seatbelts properly is a mandatory traffic regulation, a considerable portion of individuals exhibit insufficient awareness about traffic security. Consequently, it is imperative for the relevant departments to regulate and monitor the observance of seatbelt usage. At the contemporary stage, the detection of seatbelts is primarily accomplished by manual recognition, which is time-consuming and labor-intensive.

With the popularization of graphic processing technology, automatic seatbelt detection algorithms have received considerable attention. These algorithms can be categorized into diverse classes: one is the traditional image detection approach. Zhang [3] employed a linear recognition method to locate the region of the seatbelt of a driver by extracting and analyzing relevant features between the vehicle's and the driver's areas. The other is the deep-learning-based algorithm. Chen et al. [4] located the windshield area and identified the seatbelt region by utilizing both Support Vector Machines (SVMs) and Convolution Neural Networks (CNNs). Hosam et al. [5] developed an AlexNet-based adaptive weather condition seatbelt detection approach.

Research in recent years has been gradually moving towards optimizing the number of model parameters and improving network performance, making the network more suitable for real-time monitoring systems. Luo et al. [6] combined the Squeeze-and-Excitation (SE) [7] module with the DenseNet [8] to extract seatbelt features and improve detection performance. Hosseini et al. [9] replaced the adaptive average pooling in the Residual Network (ResNet) [10] with Spatial Pyramid Pooling (SPP) and Power Mean Transformation (PMT) to minimize feature loss and improve precision. Both methods initially employed the YOLO series models to detect the windshield area and then utilized the improved CNNs for driver's seatbelt detection. Moreover, Yang et al. [11] employed Faster R-CNN [12] to detect the driver's region accurately, followed by the introduction of AlexNet to identify the seatbelt.

Despite the optimizations of these algorithms being helpful to enhance network detection performance, there are still a number of shortcomings: The DenseNet with SE struggles to effectively extract crucial features when the input picture is vague. The construction of ResNet is redundant and the utilization of PMT raises the inference memory overhead and computational complexity. Additionally, Faster R-CNN possesses an intricate structure, consumes considerable computational resources, and is impractical to run on portable or mobile facilities. Therefore, there is still ample scope for advancement in the above networks.

This paper aims to tackle the challenges existing in the current seatbelt detection algorithms by proposing an automatic approach (GM-YOLOv7 and RST-Net). Due to the dearth of public driver's seatbelt datasets, this paper constructs a dataset that covers various automobile categories and locations. The major contributions of this paper include the following three aspects:

1. A lightweight Ghost module [13] is introduced to the YOLOv7-tiny [14] for efficient feature extraction while improving the Efficient Layer Aggregation Network (ELAN) in the backbone to decrease the number of model parameters, computing operations, and overall size. Moreover, the Mish [15] activation function is applied to substitute the Leaky Relu in the neck for feature aggregation, upgrading the network's non-linear capability and detection performance.
2. The lightweight deep learning algorithm is proposed to detect seatbelts by employing the channel pruning technique on the basis of ResNet, which effectively reduces the network's size and minimizes the run-time memory footprint. Moreover, a novel attention mechanism, Triplet Attention (TA) [16], is employed to improve the importance of the seatbelt feature and restrain irrelevant information. The TA module realizes more detailed attention to feature information in a cross-latitude, interactive way, improving network robustness.
3. Finally, a considerable number of experiments verify the validity of the proposed algorithm. The RST-Net has better detection behavior when compared to the well-known approaches. In addition, the Parameters (Params) and Giga Floating-Point Operations (GFLOPs) of GM-YOLOv7 and RST-Net are substantially lower compared to the baseline and other improved networks.

The rest of this paper is arranged as follows: Section 2 introduces the related work. Section 3 provides a detailed elaboration of the approaches proposed in this paper. The experimental setup, results and analyses are depicted in Section 4. Finally, the conclusions and future work are summarized in Section 5.

2. Related Studies

2.1. Object Detection Algorithm

Before the proposal of R-CNN [17], classical object detection methods struggled to effectively deal with various situations by applying sliding windows on images for classification. By combining selective techniques and CNNs, R-CNN achieved a remarkable enhancement in detection accuracy. Since then, object detection algorithms have gained the interest of lots of researchers, leading to the emergency of numerous prevalent networks.

At present, the algorithm of object detection can be categorized into two classes—two-stage and single-stage—depending on whether extra region proposal modules are necessary. The former is to generate regional recommendations, such as Fast R-CNN [18] and Mask R-CNN [19]. These algorithms produce numerous candidate regions that potentially include targets of interest at the first stage, followed by regressing the terminal boundary anchor box at the second stage. Although two-stage networks obtain excellent detection accuracy, their inference demands substantial computing operations and expenses, resulting in a relatively slower detection speed.

In single-stage networks, representative ones include RetinaNet [20]; the YOLO series, ranging from YOLOv1 [21] to YOLOv7; and the Single Shot MultiBox Detector (SSD) [22], and they treat target detection as a regression issue. The core theory of single-stage detection networks is to transport the input into the algorithm and immediately return to the objective's boundary, anchor box position, and classification. Single-stage networks often possess faster running speeds and shorter inference times than two-stage networks due to their plain constructions. Among them, the YOLO series algorithms have been widely utilized in practical applications.

2.2. Lightweight Technology

Recently, CNNs have demonstrated extraordinary capabilities in various missions, including image recognition, target detection, voice synthesis, and semantic segmentation. However, competitive performance is chiefly achieved at the expense of deepening the convolutional layers, increasing the number of Floating-Point Operations (FLOPs) and model size. For instance, a standard VGG16 has more than 130 million parameters and requires more than 10 GFLOPs. How to deploy networks more efficiently and conveniently without substantially corrupting performance has become a current research hotspot.

In recent years, a variety of approaches have been proposed to address the above-mentioned challenges, such as model pruning [23], neural architecture design [24], and model quantization [25].

Model pruning is used to remove insignificant connections or channels within pre-trained neural networks in the light of a variety of evaluation standards to lower the network's volume and computing operations without decreasing algorithm performance [26]. Several strategies delete entire neurons or channels to better utilize hardware devices, such as Graphics Processing Units (GPUs). Ref. [27] proposed a Structured Sparsity Learning (SSL) approach to improve the construction of CNNs. The other technique primarily focuses on pruning single neuron parameters, generating a sparse and irregular network. Although the parameters are relatively small, such techniques can merely obtain an acceleration of inference under customized matrix libraries. Ref. [28] proposed a technique for eliminating redundant weights by leveraging the second derivative matrix, necessitating software libraries tailored to support the pruned network.

Neural architecture design primarily centers around designing compact computing modules for CNNs to reduce network computation without compromising performance. The Xception [29] architecture employs separable convolution operation to improve the usage of model parameters efficiently. MobileNet [30], on the other hand, designs a lightweight deep neural module by splitting a vanilla convolution into a deepwise convolution and a pointwise convolution. This approach preserves the network's original feature representation capability while significantly lowering its computational complexity. ShuffleNet [31] addresses the problem of feature information interaction between channel groups by utilizing channel shuffle operations, enabling input and output channels to be correlated.

Model quantization refers to representing weights or activations in CNNs with discrete values to lower memory overheads and achieve faster computation. The process of quantization transforms parameters from high-precision floating-point numbers to low-precision fixed-point numbers, such as 16 bit, 8 bit, and 4 bit. Binary and ternary techniques [32,33] considerably accelerate the speed of models by quantifying real-valued weights into bi-

nary or ternary weights. Tensor decomposition methods [34] compress the model size or improve computation through utilizing redundancy and low-rank characteristics of the weights.

2.3. Attention Mechanism

The attention mechanism possesses the capability to autonomously shift attention to the most crucial regions within an image, neglect inconsequent components, and strengthen the network's discriminatory performance. The attention mechanism predominantly comprises channel attention, spatial attention, and a hybrid combined category: channel and spatial attention.

Channel attention emphasizes salient feature maps from the channel dimensionality through allocating diverse weights. Hu et al. [7] proposed the SE module, which gathers global information and captures non-linear characteristics between channels to enhance their representation capacity. Moreover, the spatial attention captures spatial feature maps by offering a recalibration function, which assigns different weights to various positions to acquire distinguished feature information within a spatial dimension. However, these mechanisms, such as Gather-Excite (GE) [35], require substantial computational overheads and are challenging to deploy in mobile devices.

The Convolutional Block Attention Module (CBAM) [36], a representative hybrid attention, is proposed by Woo et al. and utilizes feature maps from both channel and spatial domains to suppress irrelevant information and upgrade the capability of feature extraction. Yang et al. [37] proposed a Simple Attention Module (SimAM), which emphasized the magnitude of adjusting attention weights that change during the learning process. Hou et al. [38] proposed the Coordinate Attention (CA) mechanism by incorporating positional features into channel dimensions. This allows the improved model to concentrate on substantial significant areas with less computing overheads.

3. Methodologies

The objective of this study is to develop a deep Convolution Neural Network with fewer parameters and computing operations while maintaining outstanding anti-interference capability and superior robustness to realize rapid and precise identification. In this section, we outline our approach and provide a comprehensive description of each portion of the proposed model.

3.1. Method Overview

The flowchart of the proposed algorithm is shown in Figure 1. The algorithm begins with the precise positioning and cropping of the driver region from the input image using the GM-YOLOv7 network. Subsequently, the algorithm conducts the classification of the driver's seatbelt by employing RST-Net.

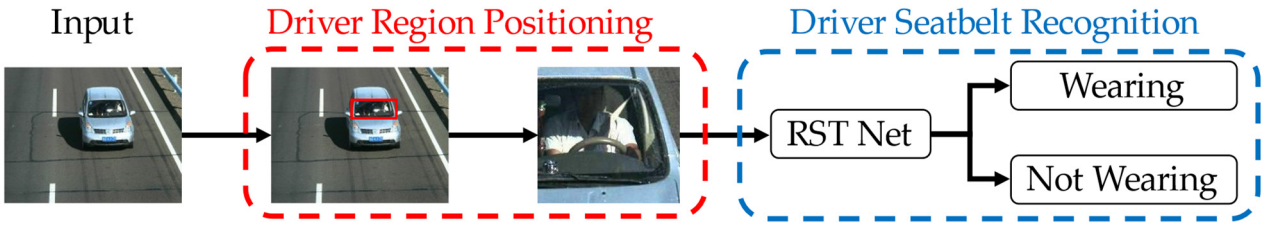

Figure 1. Flowchart of the proposed algorithm.

3.2. GM-YOLOv7 Algorithm Model

The GM-YOLOv7 network, as shown in Figure 2, consists of the following three portions: Backbone, Neck, and Head. Several lightweight G-ELAN modules, which extract the feature information of various scales from the input image through a series of cheap operations, exist in the Backbone. The Neck is built upon the SPPCSPC the module made

from the aggregation and fusion of the contextual feature information from diverse scales of the former section and improves the ELAN block by introducing more powerful activation functions. The Head is comprised of three detection heads. Such detection heads collect feature information of diverse dimensions from the Neck section, respectively, eventually generating the position, class, and degree of confidence of various objectives.

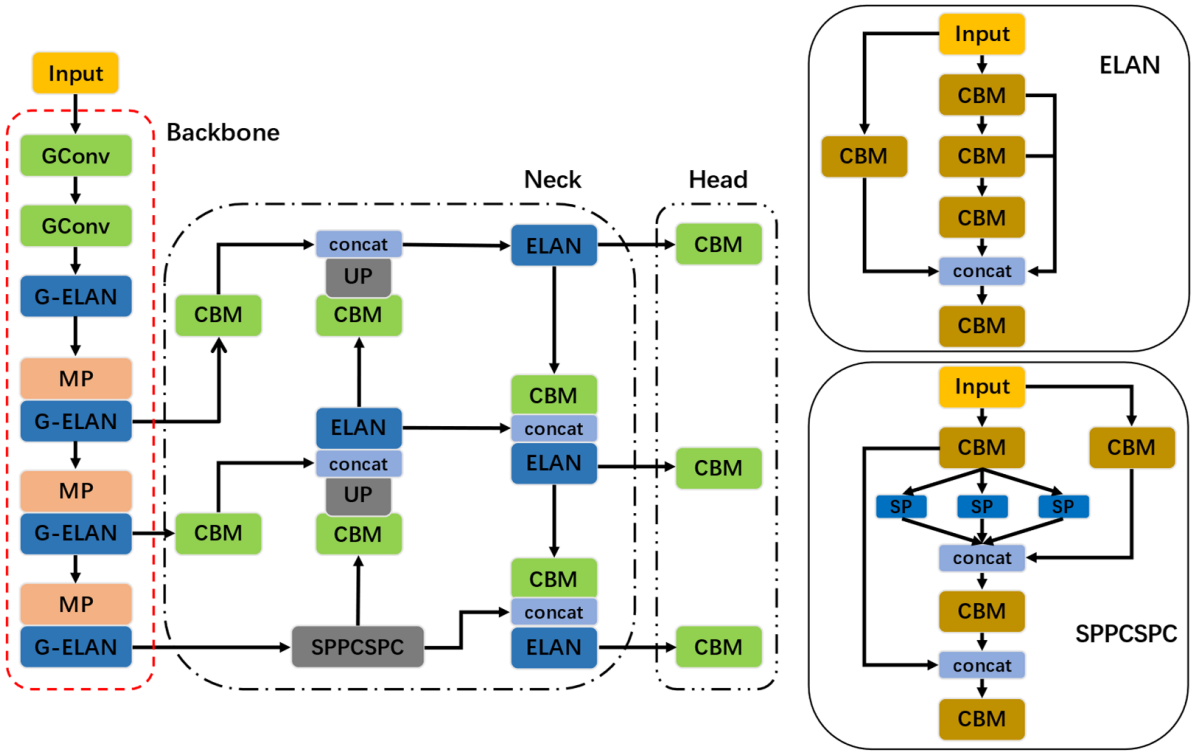

Figure 2. The overall architecture of GM-YOLOv7.

3.2.1. Slim Backbone Network Based on the Ghost Module

The Ghost module, as shown in Figure 3a, is a lightweight convolutional neural network proposed by Huawei Technology in 2020 that offers significant optimizations in terms of parameter and computational complexity. This module transforms traditional convolution into two distinct steps. In the initial phase, the minority vanilla convolutions are employed to generate intrinsic feature maps. The subsequent phase utilizes a variety of inexpensive linear operations to produce more Ghost features. Eventually, the output of the Ghost module, a new and enriched feature representation, is formed by concatenating the intrinsic features and Ghost features.

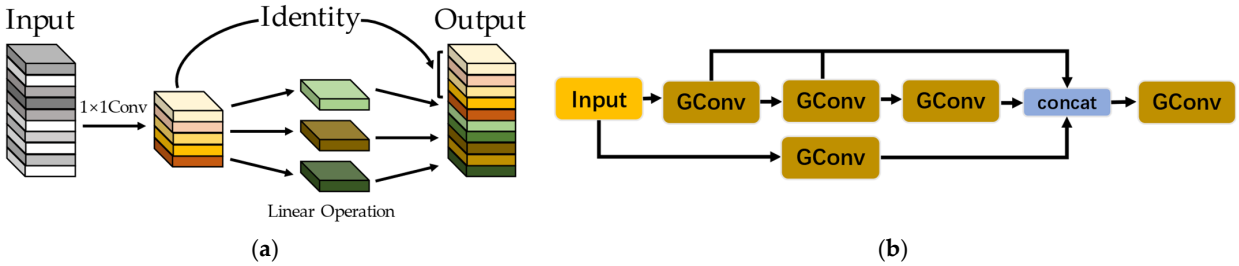

Figure 3. Schematic diagram of Ghost module. (**a**) Ghost module; (**b**) G-ELAN module.

During the standard convolutional process, the needed quantity of FLOPs can be formulated as follows:

$$FLOPs_C = c \times k^2 \times n \times h' \times w' \quad (1)$$

where c is the quantity of input channels, k is the kernel size of convolution, n is the quantity of channels for output, and h' and w' are the height and width of output data. On the other hand, the FLOPs of the Ghost module are depicted in Equation (2):

$$FLOPs_G = c \times k^2 \times n/s \times h' \times w' + n/s \times d^2 \times (s-1) \times h' \times w' \quad (2)$$

where d is the kernel size of the linear transformation and s is the quantity of similar information produced by linear transformation of $m(m \leq n)$ original feature maps. By comparing Equations (1) and (2), it can be clearly seen that the collective quantity of FLOPs and computing overheads in the Ghost module were greatly reduced without varying the dimension of the output.

In this study, the Ghost module is employed to swap out the vanilla convolution in the Backbone, reducing model volume and computational overheads while maximizing the effectiveness of feature extraction. In addition, as shown in Figure 3b, this paper introduces a novel G-ELAN, including the hierarchical residual connections and substantial Ghost modules. The architecture of G-ELAN can extract abundant feature information through diverse levels and scales, thus improving inference velocity. Consequently, the utilization of the Ghost modules and G-ELAN blocks substantially lowers the quantity of traditional 3×3 convolutional layers that exist in the initial YOLOv7-tiny, resulting in an improvement in the overall efficiency and effectiveness.

3.2.2. Improved Activation Function

The activation function is a crucial mathematical segment in convolutional neural networks that can introduce non-linearity to neurons and enable the network to model complex and non-linear relationships, thereby enhancing robustness. Leaky ReLu utilized in the YOLOv7-tiny network solves the problem of neurons being unable to learn when negative values are present. The equation for it is shown below.

$$Leaky\ ReLU = \begin{cases} x, x \geq 0 \\ \alpha x, otherwise \end{cases} \quad (3)$$

where α is the angle that directs the negative slope (used for negative input values), with a default value of 0.01. However, the non-linearity of Leaky ReLu is not great and it is unable to offer constant prediction under several circumstances.

The Mish activation function was proposed by Misra et al., which possesses the characteristics of smoothness and non-monotonicity. The function's expression is shown in Equation (4):

$$Mish = x \times \tanh(\ln(1 + e^x)) \quad (4)$$

The curves of Mish, ReLu, and Leaky Relu are shown in Figure 4. It can be seen that the Mish holds the lower bound but has no upper bound. When both the left and right limits approach the extreme values, the gradient of the Mish approaches 1, which can effectively avoid the slow convergence problem generated by a zero gradient during network training. Compared to Leaky Relu and Relu, the Mish activation function has a better degree of non-linearity, which can significantly upgrade the generalization and precision of the network. Hence, the Mish function is introduced in the Neck of the G-YOLOv7 network in order to enhance the competence of non-linear information extraction.

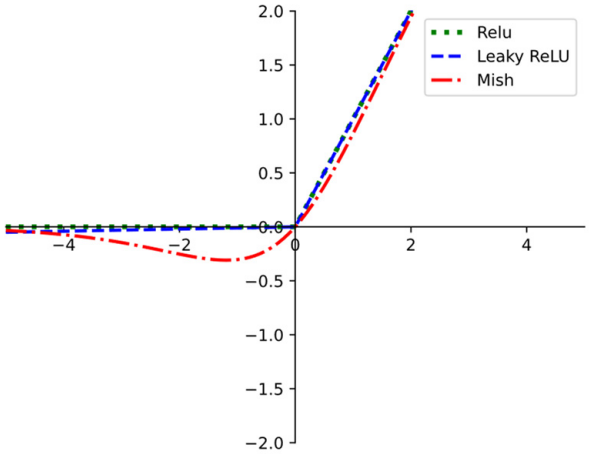

Figure 4. Activation function diagrams for Mish, Relu, and Leaky Relu functions.

3.3. Seatbelt Classification Based on RST-Net

With the popularity and advancement of deep learning techniques, numerous CNNs with prominent performances have been proposed. It is noteworthy that the residual network proposed in 2016 by He et al. has served as a foundation for further investigations and improvements by many researchers, resulting in a considerable number of variants with better performance. Although the improved algorithms based on the residual network can effectively test whether a driver wears a seatbelt, they have substantial model sizes and massive computational overhead, constraining their practicality in resource-limited scenarios. Moreover, due to various factors, such as complex road surroundings, lighting, weather, and car window reflection, CNNs' ability to extract target features is influenced, resulting in a decline in detection accuracy.

Considering the matters of the above algorithms, this paper comprises research and innovation that strengthen the performance and efficiency of Res-Net.

3.3.1. Lightweight Residual Network

The overall idea utilized in this paper is to utilize the scaling coefficient in the Batch Normalization (BN) as a criterion for channel selection, evaluating the importance of channels without demanding tailored hardware. The scaling coefficient and network weights are first trained concurrently, and L1 regularization is subsequently introduced to penalize the scaling coefficient. Finally, the convolution channel with a small scaling coefficient is pruned. This pruning process helps obtain a lightweight network by removing redundant or less important channels. In particular, the training objective of our approach is presented by Equation (5):

$$L = \sum_{(x,y)} l(f(x,w),y) + \lambda \sum_{\gamma \in \Gamma} g(\gamma) \tag{5}$$

where (x, y) represent the input and objective, respectively; W represents the trainable values; the former item corresponds to regular training loss; $g(\gamma) = |\gamma|$ denotes L1-norm; and λ represents the penalty coefficient that regulates the value of the loss function.

Batch normalization [39] is a classical improvement approach in most contemporary CNNs, typically being added after each convolution in Res-Net. Specifically, the purpose of BN is to facilitate aggregation and enhance the capacity of generalization and robustness, as formulated by Equation (6):

$$\hat{z} = \frac{z_{in} - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}; z_{out} = \gamma \hat{z} + \beta \tag{6}$$

where $\mu_{\mathcal{B}}$ and $\sigma^2_{\mathcal{B}}$ denote the average and regular variance of input features, respectively; γ and β represent the trainable scale coefficient and bias, respectively; and the utilization of ϵ is to avoid situations where the regular variance is zero.

After the preliminary training of the model, a global threshold $\hat{r}$ is introduced to examine whether feature channels will be pruned. Figure 5 displays the procedure of the channel pruning technique. Eventually, we acquire a narrower RS-Net with fewer parameters compared to the original Res-Net through utilizing the channel pruning approach, which can be directly used for recognition missions based on CNNs, making it more accessible and practical for applications.

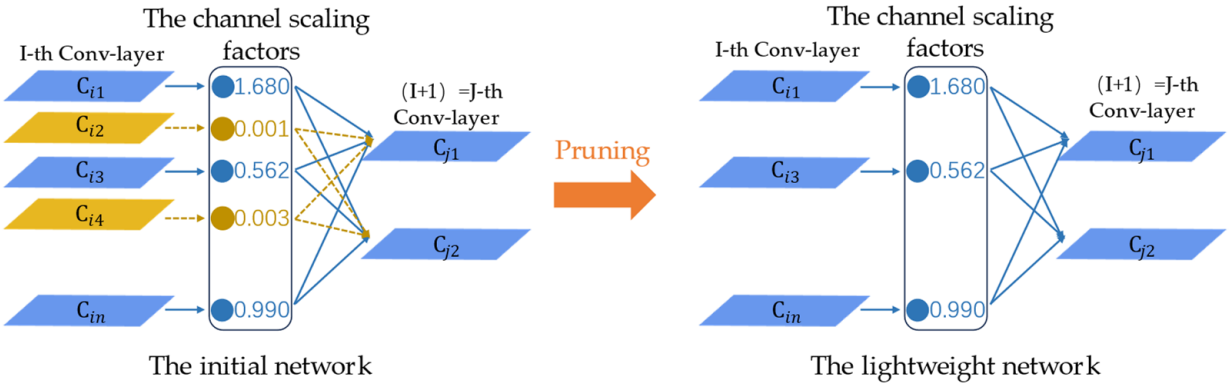

Figure 5. The process of channel pruning approach.

3.3.2. Triplet Attention

After the above-mentioned channel pruning approach, the performance of RS-Net and the competence of feature extraction usually suffer due to the variations in construction compared to initial Res-Net. To address this issue, the common method is to introduce a fine-tuning operation, where the parameters of the pruned model are retrained to make up for the supplemental reduction in performance. Although this method can assist in the restoration of the network, it may not fully realize the demand of seatbelt recognition in complicated environments. Hence, a novel and prevalent attention mechanism is introduced to recover and strengthen the capacity of the slight residual network.

Triplet attention proposed by Misra et al. [16] establishes relationships among various domains of feature information through exploiting inter-dimensional feature maps under a marginal computing cost, which is illustrated in Figure 6. Unlike the traditional attention module, triplet attention places emphasis on the significance of encoding cross-dimension information mutually rather than calculating irrespectively.

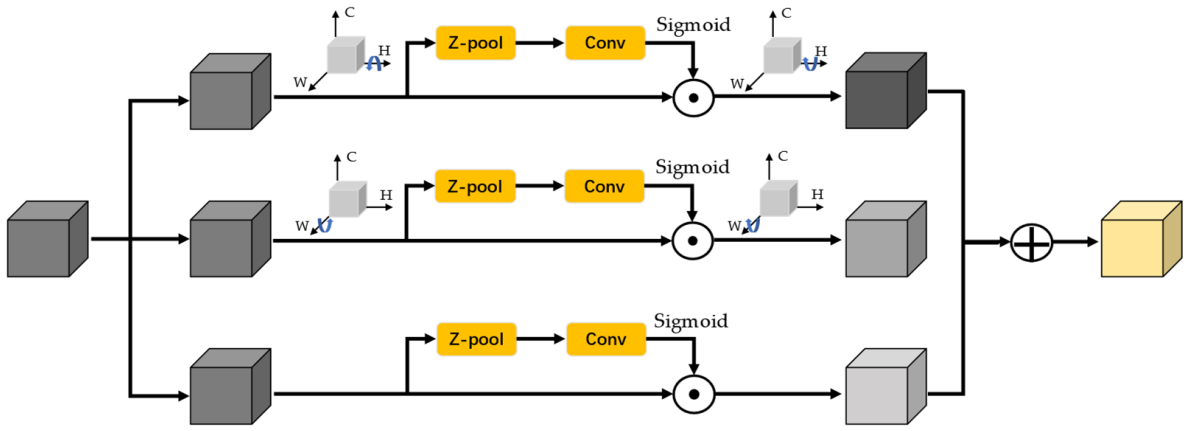

Figure 6. Illustration of the triplet attention module.

In particular, the input $\chi \in \mathbb{R}^{C \times H \times W}$ first feeds into three various branches, which separately execute interaction operations of features from diverse dimensions, i.e., (C, H), (C, W), and (W, H), where W, H, and C denote the width, height, and the quantity of channels of the input, respectively. For the (C, H) branch, χ is rotated 90° counterclockwise along the H axis. The rotated tensor denoted as X_1 is of the dimension (W, H, C). Subsequently, the Z-pool block is utilized to produce a tensor of shape (2, H, C). The Z-pool operation performs max pooling and average pooling at the first dimensionality and eventually concatenates the values, as shown in Equation (7).

$$Z-\text{pool}(\chi) = \left[\text{MaxPool}_{0d}(\chi), \text{AvgPool}_{0d}(\chi)\right] \tag{7}$$

where $0d$ represents the 0^{th} dimension that performs max pooling and average pooling operations, and [ ] denotes the concatenation conduct.

Subsequently, the output of Z-pool is transmitted through a conventional convolution layer with kernel shape $k \times k$ and a BN layer to obtain a tensor of dimension (1, H, C). The expected weights are generated through delivering the tensor to a sigmoid function and then applied to X_1. Ultimately, the acquired tensor is rotated 90° clockwise along the H axis to preserve the initial dimension of χ.

For the (C, W) and (W, H) branches, the processing operations are akin to a former bypass. The unique distinction is that the (C, W) bypass rotates along the W axis, while a rotation operation does not exist in the (W, H) branch. These operations capture inter-channel and spatial feature maps, allowing the network to exploit interactions between various domains of feature maps effectively.

Eventually, the final refined attention map y can be depicted in Equation (8):

$$y = \frac{1}{3}(\theta_1 X_{CH} + \theta_2 X_{CW} + \theta_3 X_{WH}) \tag{8}$$

where θ_1, θ_2, and θ_3 are the three inter-domain attention weights; X_{CH}, X_{CW}, and X_{WH} denote the tensor generated by the various branches.

Figure 7 represents the complete improvement procedure. In this study, Res-Net is first pruned using the channel pruning technique. The channel pruning approach can utilize the scaling coefficients in BN and sort them according to their values, eventually filtering out the insignificant feature channels to lower the number of computational operations and accomplish an inference speedup without compromising precision. In addition, the triplet attention module is inserted into the lightweight network and RST-Net can finally be obtain with efficient operation, high robustness, and high overall performance through capturing the feature information between various domains and adjusting the level of attention to diverse positions.

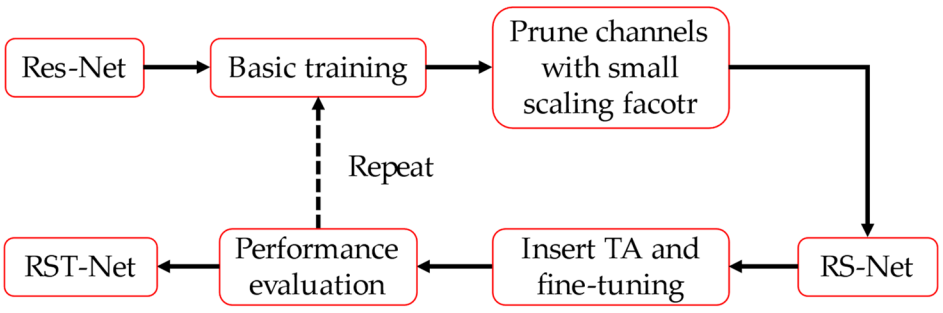

Figure 7. Overall flow chart of improvement.

4. Experimental Result

In this section, many experiments are conducted to illustrate the efficacy of the proposed networks. We first introduce the characteristics of the datasets for seatbelt detection scenarios and diverse evaluation indicators in Section 4.1; in Section 4.2, the experimental

settings and implementation configuration are presented. In Section 4.3, we carry out various ablative experiments to demonstrate the rationality of every element. Finally, the improved models are compared with the prevalent networks.

4.1. Dataset and Evaluation Metrics

A dataset is a principal condition for implementing experiments and evaluating the performance of algorithms in seatbelt detection tasks. Hence, this paper gathered pictures from a traffic department. The pictures were taken by cameras positioned in different parts of a road, encompassing various lighting conditions. The dataset contains various types of motor vehicles. The quantity of samples utilized in various instances is illustrated in Table 1. Specifically, the training and validation experiments for windshield positioning involve 4000 and 363 samples, with 366 samples employed for the test. Moreover, 3333, 400, and 250 images were utilized in the training, validation, and test steps of the seatbelt classification algorithm, respectively. Additionally, Figure 8 showcases several samples used in the experiment.

Table 1. The quantity of examples for various instances.

Different Scenarios	Training Set	Validation Set	Validation Set
Windshield	4000	363	366
Seatbelt	3333	400	250

(**a**) (**b**) (**c**)

Figure 8. Sample pictures of the experimental dataset. (**a**) Daytime images; (**b**) nighttime images; (**c**) driver seatbelt images.

To quantitatively judge the behavior of different approaches and ensure the fairness of the comparative experiment, the evaluation metrics, such as Accuracy, Precision, Recall, F1 score, and Mean Average Precision (mAP), were introduced.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1score = \frac{2TP}{2TP + FP + FN} \tag{12}$$

where TP indicates a true positive, TN represents a true negative, FP denotes a false positive, and FN is a false negative.

$$AP = \int_0^1 PRdr \tag{13}$$

$$mAP = \frac{1}{N}\sum_{i=0}^{n} AP_i \tag{14}$$

where P indicates *Precision*, R represents *Recall*, N denotes the total amount of detection categories, and AP_i is the average accuracy of class i. By calculating the area below the P-R curve for each category, the corresponding AP can be determined, and then the mAP can be determined by averaging the AP values for each category.

Moreover, Params, GFLOPs, and Size are opted to judge the computational complexity of model. Params stands for the quantity of parameters within a model and the spatial complexity is denoted in millions (M). GFLOPs refer to a network with a billion floating-point operations per second, indicating the temporary complexity in Gigabytes (G).

4.2. Experimental Configurations

The training of diverse networks was carried out utilizing the programming language Python with NVIDIA GeForce RTX 3060. Table 2 represents the detailed experimental configurations.

Table 2. Experimental platform of this paper.

Item	Edition
CPU	Intel i5-12500H
GPU	NVIDIA GeForce RTX 3060
OS	Windows 11
CUDA	11.7.1
Pytorch	1.13.1
Python	3.8

For windshield detection networks, batch size 8 and initial learning rate 0.0001 were used. The input pictures were reshaped to 640 × 640, and weight decay was set to 0.0005. The input images for the seatbelt detection stage had the dimension of 256 × 256 and the batch size was 8. The original learning rate was 0.00001, and weight decay was 0.1. The loss function employed was Cross Entropy [40], and the training process encompassed a total of 50 epochs. Furthermore, Adam was utilized as the optimizer.

4.3. Ablation Studies and Analysis

In the windshield detection phase, the influences of different advancements on the network are investigated. Specifically, the improved ELAN based on the Ghost module and Mish activation function is sequentially applied to initial YOLOv7-tiny to verify rationality and performance. Table 3 represents the obtained experimental results.

Table 3. Results of ablation experiments for diverse modules of the GM-YOLOv7 network.

Model	mAP@0.5/%	mAP@0.5:0.95/%	Params/M	GFLOPs	Size/MB
YOLOv7	99.5	81.1	6.0	13	12.3
G-YOLOv7	98.8	79.7	4.8	9.8	9.9
M-YOLOv7	99.5	81.5	6.0	13	12.3
GM-YOLOv7	99.3	84.2	4.8	9.8	9.9

Comparing the original network with G-YOLOv7 in Table 3, the conclusion that can be drawn is that when utilizing the lightweight technique in the experiments, the mAP of G-YOLOv7 slightly decreases, but the number of parameters and computational operations is considerably diminished. However, it can be found that the mAP of M-YOLOv7 is improved, while there is no significant change in Params, confirming the effectiveness of introducing the Mish activation function. Furthermore, when comparing the initial model with GM-YOLOv7, the mAP@0.5:0.95 improved by 3.1%, while the

network's parameters and computational operations significantly decreased by 20% and 24.6%, respectively, implying the predominance of the improvements employed in this paper. This is because the improved ELAN has the advantages of optimizing network structure and lowering parameters and computation overheads. Simultaneously, the Mish activation function possesses a stronger non-linear representation, thereby improving the network's performance.

Moreover, the detection performance of the GM-YOLOv7 and YOLOv7-tiny networks is compared and presented in Figure 9. By comparing two separate sets of images, it can be observed that the object detection probability of the GM-YOLOv7 network has effectively improved. In summary, the GM-YOLOv7 network not only slims down the structure of network but also upgrades performance, creating an excellent balance between detection accuracy and being lightweight.

(**a**) (**b**)

Figure 9. Visualization results. (**a**) YOLOv7-tiny; (**b**) GM-YOLOv7.

In the seatbelt detection phase, the ablation results based on ResNet are represented in Table 4. The results pointed out that the ResNet baseline model achieved scores of 94% for accuracy, 94.07 for the F1 score, 25.5 M in Params, and 4.1 in GFLOPs.

Table 4. Results of ablation experiments for each improvement of RST-Net.

Model	TA	Pruning Ratio		Accuracy/%	F1 Score/%	Params/M	GFLOPs
		25%	50%				
ResNet				94	94.07	25.5	4.1
	✓			96.75	97.76	25.6	5.4
	✓	✓		98.25	98.25	13.2	3.1
	✓		✓	94.25	94.18	9.9	2.3

For the analysis of Table 4, when the TA is merely inserted into ResNet, the improved network can automatically adjust the weight values of various channels and positions on feature maps, enabling better focus on the feature information of various types of seatbelts in the complicated traffic background. As a result, the accuracy and F1 score improved by 2.75% and 3.69%, respectively. However, the model embedded with TA increased the number of computational operations by about 24.4% compared to ResNet, requiring vast scale matrix operations and increasing inference time.

In order to alleviate the problem of network feature redundancy and lower the model's volume and computing overhead, channel pruning approaches with various pruning ratios were introduced. When setting the pruning rate to 25%, the accuracy and F1 score increased by 4.25% and 4.18%, respectively, compared with baseline. This is principally because the channel pruning method reduces several redundant parameters that may interfere with the model's learning. Additionally, it mitigates overfitting to a certain extent, enhances the generalization capability, and lowers the quantity of parameters by 48% compared to baseline. Furthermore, when the pruning ratio was increased to 50%, the Params and GFLOPs significantly decreased by 60% and 44%, respectively, while the accuracy and F1 score merely increased by 0.25% and 0.11% compared to the baseline model. The accuracy and F1 score of the model with a 25% pruning ratio were 4% and 4.07% higher, respectively, than the model with a 50% pruning ratio at the expense of a 25% increase in Params and GFLOPs. By comparing the experimental results of a 25% and 50% pruning ratio, the conclusion that a network with a lower pruning factor is more likely to improve precision and performance than a radical one could be drawn. Therefore, the 25% pruning ratio was eventually selected to obtain RST-Net due to the salient differences in the quantity of parameters and network performance.

Furthermore, to validate the rationality of triplet attention, a variety of prevalent attention mechanisms commonly used in computer visual scenarios were selected for comparative experiments, including SE, CBAM, CA, Efficient Channel Attention (ECA) [41], and Shuffle Attention (SA) [42]. Table 5 exhibits the experimental results.

Table 5. Comparative experiments based on diverse attention mechanisms.

Model	Attention Module	Accuracy/%	Precision/%	Recall/%
ResNet	SE	97	97.47	95.5
	SA	95.75	96.45	95
	ECA	96	95.54	96.5
	CBAM	96.75	96.98	96.5
	CA	97.25	98.96	95.5
	TA	98.25	98.49	98

For the observation of Table 5, it is obvious that the model embedded with TA outperforms the existing prevalent attention modules. The argument that TA can provide better behavior is discussed as follows. The classical channel attention mechanisms such as SE and ECA merely model the interaction among various channels without focusing on the spatial information, leading to inadequate feature extraction and suboptimal performance. Moreover, although CBAM and SA calculate both channel and spatial attention, inter-channel and spatial information failed to be completely utilized. In contrast to these

attention mechanisms, TA leverages inter-dimensional information, leading to improved feature representation and enhanced performance.

4.4. Model Comparations

For the windshield detection scenario, single-stage detection models such as YOLOv3 [43], YOLOX [44], and YOLOv5 were utilized to train model weights under an identical environment and dataset. The comparison between these models and the GM-YOLOv7 network primarily focused on mAP@0.5:0.95, Params, and GFLOPs, which are crucial comparison indicators. Table 6 displays the comparison results of diverse detectors.

Table 6. Comparison of windshield detection results of various networks.

Network	mAP@0.5/%	mAP@0.5:0.95/%	Params/M	GFLOPs	Size/MB
YOLOv3-tiny	98.4	61.8	8.7	13.0	16.6
YOLOX-s	99.0	79.6	8.8	26.5	68.0
YOLOv5-s	99.2	80.0	7.0	15.8	14.4
YOLOv7-tiny	99.5	81.1	6.0	13.0	12.3
GM-YOLOv7	99.3	84.2	4.8	9.8	9.9

From Table 6, by comparing the mAP@0.5:0.95, the proposed lightweight network outperforms YOLOv3-tiny, YOLOx-s, and YOLOv5-s by 22.4%, 4.6%, and 4.2%, respectively. This reveals that the GM-YOLOv7 network effectively utilizes the feature extraction of the improved ELAN at the expense of minimal computation overhead. The proposed network achieves the greatest detection performance in terms of mAP@0.5:0.95 while preserving the minimum Params and GFLOPs. Compared to the widely used YOLOv5-s, GM-YOLOv7 demonstrates improved performance while considerably decreasing the Params and GFLOPs by 31% and 38%, respectively, enabling the algorithm to be more appropriate for deployment in mobile devices. Through contrasting the above principal evaluation indexes, GM-YOLOv7 exhibits significant advantages over other popular networks in the light of detection precision, Params, and GFLOPs, making it a reliable and effective choice for windshield detection tasks.

For the seatbelt detection scenario, experimental comparisons between the proposed network and multiple CNNs, including AlexNet, DenseNet, EfficientNet [45], ResNeXt [46], and Wide ResNet [47], were conducted. Table 7 displays the results of these experiments.

Table 7. Comparison of seatbelt detection results of multiple models.

Model	Accuracy/%	Precision/%	Recall/%	F1 Score/%	Params/M	GFLPOs
AlexNet	86.74	82.67	93	87.53	61.1	0.93
DenseNet	97.75	97.99	97.50	97.74	20.0	5.7
EfficientNet	97.50	97.98	97	97.49	12.2	1.3
ResNeXt	96.75	96.06	97.50	96.77	25.0	5.6
Wide ResNet	97.25	95.22	99.5	97.21	68.9	15.0
RST-Net	98.25	98.49	98	98.25	13.2	3.1

From Table 7, it is beyond dispute that RST-Net is more efficient than several prominent algorithms, providing powerful support for practical applications. Although the Recall of RST-Net was lower than Wide ResNet by a small amount, its Params and GFLOPs were merely 0.19 times and 0.2 times that of Wide ResNet, respectively. Additionally, RST-Net achieves an F1 score that is 10.72% higher than AlexNet, 1.48% higher than ResNeXt, and 1.04% higher than Wide ResNet, while its Params and GFLOPs were considerably smaller than the above models. Furthermore, when contrasting the major performance indexes comprehensively, RST-Net demonstrates preferable performance, especially in the light of the Params, further substantiating the efficiency of the improvements proposed in this study.

Additionally, to verify the merit of RST-Net for seatbelt detection, Gradient-weighted Class Activation Mapping (Grad-CAM) [48] was employed to depict visualization results of various models employed in this chapter. Figure 10 displays visualization results. The first three rows show a driver that wore a seatbelt, and the last three rows exhibit a situation where a driver did not use a seatbelt. Obviously, RST-Net has the capability to concentrate more attention on the salient feature information of the seatbelt, even in darker and low-contrast images. Additionally, RST-Net not only distinguishes whether the driver is wearing the seatbelt through focusing on the relevant regions where seatbelt features may be present, but also screens out the irrelevant areas located in an image's background, thereby improving detection accuracy. On the contrary, the other models either do not work well or lack generalization.

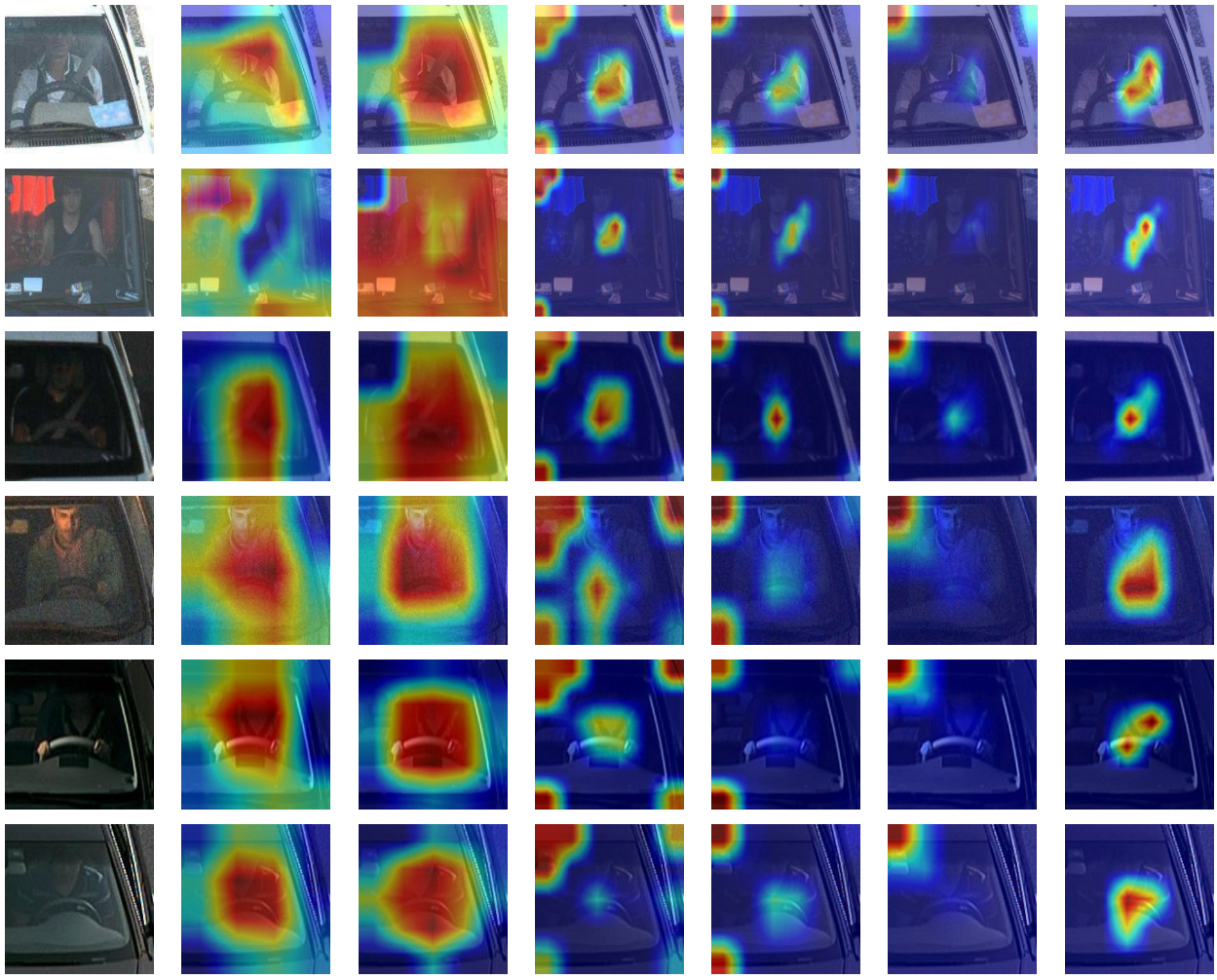

Figure 10. Visualization of feature maps produced by various models. (**a**) Original image; (**b**) AlexNet; (**c**) DenseNet; (**d**) EfficientNet; (**e**) ResNeXt; (**f**) Wide ResNet; (**g**) RST-Net.

5. Conclusions

The detection network of seatbelt usage with high precision and compact structure is a prominent and formidable investigation issue in the field of intelligent traffic systems. This paper proposes a high-precision and lightweight seatbelt detection algorithms, GM-YOLOv7 and RST-Net, to address the problems of low detection accuracy and large model parameters in the current work. Firstly, based on the YOLOv7 network, the G-ELAN module and the Mish activation model are added to alleviate the redundant issues of

parameters and address the problem of insufficient non-linear representation capability. Secondly, channel pruning is applied to the ResNet model at a 25% ratio to optimize the architecture and speed up the inference time without decreasing the model's accuracy. Finally, incorporating the TA mechanism into network weakens the interference of the background on seatbelt features and enhances the capability of feature learning.

The experimental results reveal that, compared with YOLOX and YOLOv5, the proposed GM-YOLOv7 network could effectively improve target detection and reduce the number of parameters and computational overheads, achieving a mAP of 84.2% and minimum GFLOPs. Meanwhile, the lightweight RST-Net obtained a precision of 98.49% and an F1 score of 98.25%, outperforming the other improved models. Compared with several large-scale networks, the proposed models achieved the best detection performance with the fewest amounts of parameters and computations. Therefore, the experimental analyses verify the rationality and effectiveness of the proposed networks.

In our future research study, we will pay more attention to image collection for seatbelt detection to further expand the dataset and upgrade the generalization ability of the model. Additionally, although this paper makes a significant contribution by making this algorithm lightweight, the edges or mobile devices and embedded systems maintain a poor real-time performance due to insufficient arithmetic power. Therefore, the focus of future research should be to investigate various techniques, such as knowledge distillation, and the impact of hardware devices on this model's performance to further optimize its structure and upgrade its real-time detection capabilities.

Author Contributions: Conceptualization, L.Q. and J.R.; methodology, L.Q. and J.R.; software, L.Q. and J.R.; validation, J.R. and X.Z.; resources, J.R.; data curation, J.R.; writing—original draft preparation, J.R.; writing—review and editing, J.R., L.Q. and X.Z.; visualization, J.R. and X.Z.; supervision, L.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (U1504617).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included within the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, Z.; Ma, Y. Detection and Recognition of Stationary Vehicles and Seat Belts in Intelligent Internet of Things Traffic Management System. *Neural Comput. Appl.* **2022**, *34*, 3513–3522. [CrossRef]
2. Wang, D. Intelligent Detection of Vehicle Driving Safety Based on Deep Learning. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1095524. [CrossRef]
3. Zhang, D. Analysis and Research on the Images of Drivers and Passengers Wearing Seat Belt in Traffic Inspection. *Clust. Comput.* **2019**, *22*, 9089–9095. [CrossRef]
4. Chen, Y.; Tao, G.; Ren, H.; Lin, X.; Zhang, L. Accurate Seat Belt Detection in Road Surveillance Images Based on CNN and SVM. *Neurocomputing* **2018**, *274*, 80–87. [CrossRef]
5. Hosameldeen, O. Deep Learning-Based Car Seatbelt Classifier Resilient to Weather Conditions. *Int. J. Eng. Technol.* **2020**, *9*, 229–237. [CrossRef]
6. Luo, J.; Lu, J.; Yue, G. Seatbelt Detection in Road Surveillance Images Based on Improved Dense Residual Network with Two-Level Attention Mechanism. *J. Electron. Imag.* **2021**, *30*, 033036. [CrossRef]
7. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
8. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
9. Hosseini, S.; Fathi, A. Automatic Detection of Vehicle Occupancy and Driver's Seat Belt Status Using Deep Learning. *SIViP* **2022**, *17*, 491–499. [CrossRef]
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.

11. Yang, K.; Zhang, D.; Li, Y. Seat belt detecting of car drivers with deep learning. *J. China Univ. Metrol.* **2017**, *28*, 326–333.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
13. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2018; pp. 1577–1586.
14. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696.
15. Misra, D. Mish: A Self Regularized Non-Monotonic Activation Function. *arXiv* **2020**, arXiv:1908.08681.
16. Misra, D.; Nalamada, T.; Arasanipalai, A.U.; Hou, Q. Rotate to Attend: Convolutional Triplet Attention Module. *arXiv* **2020**, arXiv:2010.03045.
17. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *arXiv* **2014**, arXiv:1311.2524.
18. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
19. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *arXiv* **2018**, arXiv:1703.06870.
20. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2018**, arXiv:1708.02002.
21. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
22. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; Volume 9905, pp. 21–37.
23. He, Y.; Zhang, X.; Sun, J. Channel Pruning for Accelerating Very Deep Neural Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1398–1406.
24. Deng, B.L.; Li, G.; Han, S.; Shi, L.; Xie, Y. Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. *Proc. IEEE* **2020**, *108*, 485–532. [CrossRef]
25. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2704–2713.
26. Zhang, P.; Zhong, Y.; Li, X. SlimYOLOv3: Narrower, Faster and Better for Real-Time UAV Applications. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE, Seoul, Republic of Korea, 27–28 October 2019; pp. 37–45.
27. Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; Li, H. Learning Structured Sparsity in Deep Neural Networks. *arXiv* **2016**, arXiv:1608.03665.
28. LeCun, Y.; Denker, J.S.; Solla, S.A. Optimal Brain Damage. In Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, USA, 28 March–1 April 1993; pp. 293–299.
29. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2017**, arXiv:1610.02357.
30. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
31. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *arXiv* **2017**, arXiv:1707.01083.
32. Courbariaux, M.; Hubara, I.; Soudry, D.; El-Yaniv, R.; Bengio, Y. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or −1. *arXiv* **2016**, arXiv:1602.02830.
33. Rastegari, M.; Ordonez, V.; Redmon, J.; Farhadi, A. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. *arXiv* **2016**, arXiv:1603.05279.
34. Kolda, T.G.; Bader, B.W. Tensor Decompositions and Applications. *SIAM Rev.* **2009**, *51*, 455–500. [CrossRef]
35. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks. *arXiv* **2019**, arXiv:1810.12348.
36. Woo, S. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521.
37. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Online, 18–24 July 2021; pp. 11863–11874.
38. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. *arXiv* **2021**, arXiv:2103.02907.
39. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
40. Mao, A.; Mohri, M.; Zhong, Y. Cross-Entropy Loss Functions: Theoretical Analysis and Applications. *arXiv* **2023**, arXiv:2304.07288.
41. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *arXiv* **2020**, arXiv:1910.03151.
42. Yang, Q.-L.Z.Y.-B. SA-Net: Shuffle Attention for Deep Convolutional Neural Networks. *arXiv* **2021**, arXiv:2102.00240.
43. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
44. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
45. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2020**, arXiv:1905.11946.
46. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. *arXiv* **2017**, arXiv:1611.05431.

47. Zagoruyko, S.; Komodakis, N. Wide Residual Networks. *arXiv* **2017**, arXiv:1605.07146.
48. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [CrossRef]