A Novel Deep Neural Network Compression Model for Airport Object Detection

LYU Zonglei[1,2*], *PAN Fuxi*[1,2], *XU Xianhong*[1,2]

1. Information Technology Research Base of Civil Aviation Administration of China, Tianjin 300300, P. R. China;
2. College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, P. R. China

(Received 25 May 2020; revised 10 June 2020; accepted 30 June 2020)

Abstract: A novel deep neural network compression model for airport object detection has been presented. This novel model aims at disadvantages of deep neural network, i.e. the complexity of the model and the great cost of calculation. According to the requirement of airport object detection, the model obtains temporal and spatial semantic rules from the uncompressed model. These spatial semantic rules are added to the model after parameter compression to assist the detection. The rules can improve the accuracy of the detection model in order to make up for the loss caused by parameter compression. The experiments show that the effect of the novel compression detection model is no worse than that of the uncompressed original model. Even some of the original model false detection can be eliminated through the prior knowledge.

Key words: compression model; semantic rules; pruning; prior probability; lightweight detection

CLC number: TP183 **Document code**: A **Article ID**: 1005-1120(2020)04-0562-12

0 Introduction

New technologies and new concepts make the connotation of smart airports more fulfilling. In the flight area, there are planes taking off and landing, and ground support vehicles, such as fuel trucks, trailers, water trucks, etc., operating. Moreover, there are many departments and the overall situation is complex. Therefore, the efficient operation of the flight area is the key to the efficient operation of the whole airport.

After landing at the airport or before taking off, the airplane needs to receive various ground services on the apron, such as fuel filling, passenger ferry, cargo transportation and cleaning services. These services need to be performed by various service equipment according to the specified process, and need to be completed through the joint coordination of information, equipment, personnel and other resources. The scheduling of ground service vehicles involves many factors. Unreasonable coordination will lead to low efficiency, vehicle collision and other events, so we need to conduct real-time monitoring and detection of vehicles and aircraft on the apron.

The object detection algorithm can realize the detection of vehicles and aircraft on the apron, so as to get the arrival and departure time of aircraft, and the start and end time of ground service vehicles. Currently, the algorithms for object detection mainly include region-based convolutional neural networks (RCNNs)[1], fast-RCNN[2], mask RCNN[3], faster-RCNN[4], light-head RCNN[5], region-based fully convolutional networks (RFCNs)[6] and you only look once (YOLO)[7].

With the people's demand for hardware applications gradually increases, applications in the direction of computer vision have gradually been required to be embedded in mobile terminals, without heavy back-end. However, neural networks are both computationally intensive and memory intensive, which makes them difficult to deploy in embedded sys-

*Corresponding author, E-mail address: zllv@cauc.edu.cn.

How to cite this article: LYU Zonglei, PAN Fuxi, XU Xianhong. A novel deep neural network compression model for airport object detection[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(4): 562-573.

http://dx.doi.org/10.16356/j.1005-1120.2020.04.007

tems. Therefore, model compression is starting to get a lot of attention. With the compression model, the detection model can be embedded into mobile devices, and many target detection tasks can be identified directly at the front end for lightweight without considering the delay caused by a large amount of network transmission, which greatly improves the real-time and sensitivity of target recognition. As for compression models, there are also many related researches. In the summary article of deep neural network model compression, some algorithms and strategies of model compression in recent years have been summarized, and the performance, advantages and disadvantages and latest research results of these methods have been analyzed and summarized[8]. At present, the main model compression methods include model layer pruning[9], model channel pruning[10], lightweight network design, knowledge distillation[11], quantification, architecture search and so on. There are also some research departments specializing in the relationship between the pruning rate of the neural network model and the compression effect.

Traditional compression models are often achieved by reducing the number of parameters or the depth of the model to maximize the role of each parameter[12]. This method usually sacrifices the expression ability of the model, which will cause a certain loss of detection accuracy. In recent years, semantic compression has attracted people's attention in some different fields, among which there are many studies in the direction of image compression. For example, a semantic compression scheme for digital images based on vector quantization and data hiding is proposed[13], which can easily obtain high-quality reconstructed images with original size.

Machine learning can be seen as the process of abstracting knowledge from concrete objects. Abstraction itself is a kind of compression and generalization of information. Therefore, summarizing the abstract knowledge obtained by learning the model itself can also achieve the effect of model compression. We have learned a lot of knowledge in life, abstracting these knowledge into prior knowledge and adding it to the target detection model can both help the model detection and compress the model from another angle. Wang proposed a target tracking strategy based on the prior knowledge of the target[14], which ultimately improved the performance of the target tracking algorithm. So we propose a compression model based on prior semantic information, which extracts semantic information from the detection results of the uncompressed model, and then embeds these semantic information into the compression model as prior knowledge. This method will improve the accuracy of the compression model loss, and achieve a customized compression model.

The apron is special compared to other scenes. The types of objects on the apron are relatively fixed. The ground service vehicle has a fixed position relative to the aircraft. According to the aircraft type and relative position, the type of vehicle can be determined.

In this paper, we propose a novel deep neural network compression model for airport object detection. By using the uncompressed model to obtain temporal and spatial semantic rules, airport rules are added to the model after parameter compression to assist detection, which improves the accuracy of the model, makes up for the loss caused by parameter compression, and realizes the customized model.

The rest of this paper is organized as follows: the principles and strategies of parameter model will be first introduced. Then a semantic compression model based on the parameter compression model with temporal and spatial prior semantic rules will be proposed, and its principle and method will be explained. Finally, it will be applied to the core computing module of PaddlePi-K210 terminal, and the results will be compared and analyzed.

1 Related Work

The compression model can be divided into a parameter compression model and a semantic compression model, the part of the parameter compression is compressed by using the channel pruning compression strategies proposed by Liu et al.[10], and then a new semantic compression model is pro-

posed on the basis of the parameter compression. The following will introduce several compression strategies of the parameter compression model and the principle of semantic compression model.

1.1 Parameter compression model strategy

The parameter compression algorithm in this paper is aimed at the classification model. The principle is based on the gamma coefficient of batch normalization (BN) layer for channel pruning. The channel pruning strategy can greatly reduce the model parameters and calculation amount, and reduce the model's occupation of resources. The following will introduce the sparse training and several different channel pruning strategies based on YOLOv3 in the parameter compression model strategy.

Sparse training is a strategy to improve the generalization ability of the model. The most important function of sparse training is to use the size of the BN gamma coefficient corresponding to each channel to determine which channel layers can be pruned. It use the gamma coefficient of BN layer to mark the important channel of feature map output by convolution layer and those channels with small gamma that can be pruned. In the whole sparse training process, by adding an extra gradient to the model with a constant sparse factor, the compression can achieve a higher effect. In fact, sparse training is a game process of accuracy and sparsity. Large sparse factors are generally fast in sparse but faster in accuracy. Small sparse factors are generally slow in sparse but slower in accuracy. The sparse speed can be accelerated with a large learning rate, and the adoption of a small learning rate in the later stage will also contribute to the recovery of accuracy.

The first channel pruning strategy is a conservative strategy. This strategy is based on YOLOv3, in which there are five groups of 23 shortcut connections that correspond to add operations. This strategy does not pruning the directly connected layers of the shortcut, which avoids the problem of dimensional processing. But it also achieves a higher pruning rate, which is very helpful for reducing the model parameters. Although it has the lowest pruning rate, it handles the details of pruning very well. This pruning strategy can be used in YOLOv3, YOLOv3-spp and YOLOv3-tiny.

The second channel pruning strategy prunes the convolutional layer involving shortcuts. The pruning uses the mask of the first convolutional layer in each group of shortcuts. A total of five masks are used to achieve the pruning of five groups of shortcut-related convolutional layers, which improves the pruning rate and the activation offset processing after pruning involving shortcuts, and by modifying the threshold rules, we can set a higher pruning rate. The setting of pruning rate and the precision change after pruning are closely related to the sparse training. This pruning strategy can be used in YOLOv3 and YOLOv3-spp.

The channel pruning rate of the third channel pruning strategy is relatively high. It first finds the mask of each convolutional layer with a global threshold, and then for each group of shortcuts, it takes the union of the pruning masks of each convolutional layer connected, and it prunes with the merged mask, so that each relevant layer is considered. At the same time, by limiting the reserved channels of each layer and adding processing to the activation offset value, the accuracy loss during pruning can be reduced. This pruning strategy can be used in YOLOv3 and YOLOv3-spp.

1.2 Semantic compression model strategy

Machine learning can be seen as a process of abstracting knowledge from concrete objects, and abstraction itself is a compression and generalization of information. Therefore, summarizing the abstract knowledge obtained by learning the model itself can also achieve the effect of model compression from another perspective. There are many rules in the airport. If we integrate these rules into the model as prior knowledge, we can greatly improve the model detection effect, because prior knowledge can help us obtain as many prior rules as possible before the detection. These prior rules can play a guiding role in the detection. In other words, the use of prior knowledge also indirectly determines the quality of the model.

Combining the above principles, we can first extract the specific location distribution information and time distribution information of the target object from the detection results of the parameter compression model. Through statistics and calculation of the time and space distribution of the target object, the corresponding time and space prior probability can be obtained. In fact, these prior semantic information can be abstracted as prior distribution rules of the target object, and then the prior probability obtained by combining time and space and the confidence degree of the model detection after compression are calculated by using Bayesian formula to obtain the posterior probability. The results calculated by this method can not only use the information of training samples, but also consider the time distribution probability and spatial location distribution probability of the target object, which can alleviate the precision loss caused by the parameter compression model and greatly improve the accuracy of the experimental detection results. At the same time, the semantic information is abstracted to summarize prior business rules, and implanted into the new model from the outside, which realizes semantic compression, and also realizes a customized compression model to a certain extent.

2 Lightweight Model with Added Prior Semantic Information

This section will be divided into the parameter compression model and the semantic compression model.

It will introduce how to obtain the uncompressed original model and the compressed model by the parameter compression algorithm in turn. Then, according to the detection results of the uncompressed original model, it will statistically summarize and obtain the prior knowledge of relevant operation rules of the semantic compression on the parameter compression target object on the apron, and add the prior knowledge into the parameter compressed model to realize model, and finally realize the lightweight model with added semantic information.

2.1 Parameter compression model

The most important function of sparse training is to use the size of the BN gamma coefficient corresponding to each channel to determine which channel layers can be pruned. Each layer corresponds to a BN gamma, and its formula which is proposed by Liu et al.[10] is

$$\begin{cases} \hat{z} = \dfrac{z_{\text{in}} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \\ z_{\text{out}} = \gamma \hat{z} + \beta \end{cases} \tag{1}$$

where μ_B and σ_B are the mean and standard deviation values of input activations over B; γ and β the trainable affine transformation parameters (scale and shift) which provides the possibility of linearly transforming normalized activations back to any scales; and z_{in} and z_{out} the input and output of the BN layer. B refers to the current minibatch; and ε is a constant added to prevent the denominator from being 0. We set a global threshold for all layers of the entire network, sort the BN gamma values from small to large, and take the BN gamma value of 70% of the positions as the threshold, so we can get a compact network with fewer parameters, less memory at runtime and less computation.

After sparse training on the original model, by comparing the accuracy of the sparsely trained models with different training times to the target object on the apron, select a model with a relatively good mean average precision (mAP) value as the original model, and then the next step of pruning of the original model is carried out. Due to the different methods of the three channel pruning strategies, there are some subtle differences in the size and loss of model performance after pruning. The pruning model with relatively small size and loss of performance after pruning is selected as the compressed model. In the actual detection of the target object on the apron, pruning will lead to some loss of accuracy, which will increase the probability of missed detection and false detection. Therefore, after choosing a relatively good pruning strategy to prune the model, we should further process the model and the recognition results to improve the accuracy of the model. The principle structure of the parameter pruning

model is shown in Fig.1.

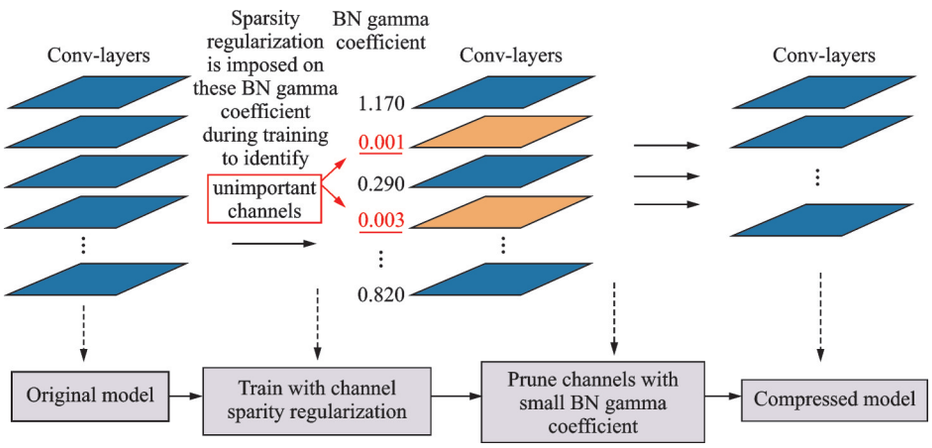

Fig.1 Parameter compression model

2.2 Semantic compression model

Using the uncompressed original model to detect the target object on the apron, we can get the temporal and spatial distribution information of the target object, then calculate the corresponding prior probability, and combine the prior probability with the compressed model detection results to calculate the posterior probability. Finally, it can help to provide the bounding boxes, thus greatly improving the accuracy and achieving semantic compression. The principle structure of the semantic compression model is shown in Fig.2.

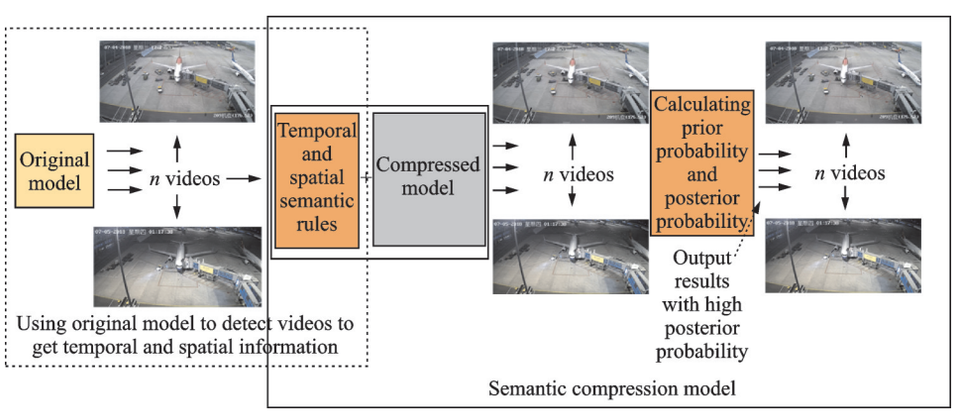

Fig.2 Semantic compression model

The semantic compression model based on prior knowledge of temporal and spatial distribution actually depends on the special scene of airport operations.

The temporal and spatial distribution of objects on the apron has a great correlation. Although there are differences in some individuals, they are generally subject to certain temporal and spatial distribution. The spatial distribution relationship is shown in Fig.3.

In terms of time distribution, the No.209 stand of Guiyang Airport is selected, and the time when

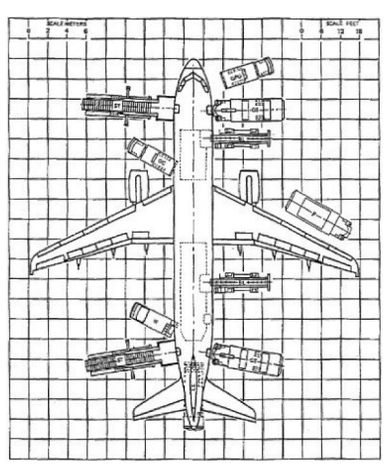

Fig.3 Spatial distribution relationship

the plane arrives at the stand is taken as the time 0:00 and the flight starts subsequent operations. Thus, the relative time distribution of several key links and arrival time is obtained, as shown in Table 1. We can see that the time distribution of objects on the apron is regular.

Table 1 Temporal distribution relationship

Time of plane wheel block / min	Opening time of corridor bridge/ min	Arrival time of luggage vehicle/min	Arrival time of refuel vehicle/ min	Arrival time of food vehicle/ min	Arrival time of rubbish vehicle/ min	…
13:53	13:54	13:55	13:54	14:00	14:01	…
10:00	10:01	10:02	10:04	10:06	10:06	…
1:15	1:16	1:16	1:18	1:19	1:22	…
…	…	…	…	…	…	…

In order to count the time and space distribution of each target object on the apron, select n pieces of video at the same angle. Based on the number of aircraft arrival frames, calculate the position and frame number of each target object on the apron after the aircraft reaches its position.

By using the Intersection over union (IOU) rule to calculate and compare with the error parameter, we can find that the error parameter takes 10 to 20, which is equivalent to the average IOU of about 0.7, and the error parameter takes 20 to 30, which is equivalent to the average IOU of about 0.6.

The statistical algorithm is as follows:

Firstly, a video is selected to be detected by using the original model. The coordinates of the bounding boxes identified in each frame are output and saved to a txt file in the format

$$\text{output}=(x_1,y_1,x_2,y_2,\text{cls},t,\text{num}) \tag{2}$$

where x_1 is the abscissa of the upper left corner of the bounding boxes, y_1 the ordinate of the upper left corner of the bounding boxes, x_2 the abscissa of the lower right corner of the bounding boxes, y_2 the ordinate of the upper left corner of the bounding boxes, cls the object category in the bounding boxes, t the corresponding number of frames, and num the number of times the object has appeared in this position in history.

The initial num is set to 1, then $m(2\leqslant m\leqslant n)$ videos are continuously detected, and the format of each detected information in each frame is

$$\text{detect}=(x_1,y_1,x_2,y_2,\text{cls},t,\text{num}) \tag{3}$$

for any one in the txt file

$$\text{output}=\{(x_{1i},y_{1i},x_{2i},y_{2i},\text{cls}_i,t_i,\text{num}_i)|0\leqslant i\leqslant \text{len}\} \tag{4}$$

len is the length of txt file, if $\text{cls}=\text{cls}_i$ and $t=t_i$ are the same, set error=20, if

$$\begin{aligned} x_{1i}-\text{error}\leqslant x_1\leqslant x_{1i}+\text{error}\\ y_{1i}-\text{error}\leqslant y_1\leqslant y_{1i}+\text{error}\\ x_{2i}-\text{error}\leqslant x_2\leqslant x_{2i}+\text{error}\\ y_{2i}-\text{error}\leqslant y_2\leqslant y_{2i}+\text{error} \end{aligned} \tag{5}$$

statisfy, it can be explained that the target object appeared at the position near the area of $s=(x_1, y_1, x_2, y_2)$ in this frame, then add the num count to one. If no match is found, the detected coordinate information will be added to the txt file in the same format, and finally get the time-space distribution of all target objects.

According to the txt file of the obtained frame number and coordinate position distribution information of all target objects, for an arbitrary input format

$$\text{input}=(x_1,y_1,x_2,y_2,\text{cls},t) \tag{6}$$

the objects of the category can be queried, and the number of occurrences of the frame at this position is num, so the probability formula of the category object at this position at this time is

$$P(A)=\left\{\frac{\text{num}}{n}|1\leqslant\text{num}\leqslant n\right\} \tag{7}$$

We set $P(A)$ as the probability that an object is detected, and $P(B)$ as the prior probability that the object appears at this position at this time in history. $P(A|B)$ is the probability that there are objects of this category in a certain position and time, and the actual situation is the same, which can be regarded as the true positive rate of the model. $P(C)$

is the probability that there are objects of this kind in a certain position and time, but the actual situation is opposite, which can be regarded as the false positive rate. So the posterior probability is

$$P(B|A)=\frac{P(AB)}{P(A)}=\frac{P(B)P(A|B)}{P(A)}=\frac{P(B)P(A|B)}{P(B)P(A|B)+P(C)(1-P(B))}=\frac{\text{num}\cdot P(A|B)}{\text{num}\cdot P(A|B)+P(C)(1-\text{num})}$$
$$1\leqslant \text{num}\leqslant n \tag{8}$$

where $P(A|B)$ and $P(C)$ can be obtained by testing the model. Finally, a threshold is set in advance. When the posterior probability is greater than the threshold, the candidate box is output.

3 Experiment

3.1 Parameter compression model

This experiment is carried out on YOLOv3 models and visual geometry group (VGG) network of Pytorch. The following will introduce the data preparation, sparse training and pruning of the parameter compression model.

First of all, after setting up the environment according to the needs, make our own relevant data sets, and then download the relevant YOLOv3 pretraining weights to complete the data preparation work.

Run the train file to start sparse training, test the accuracy of the model during the training process, find the relatively optimal training times, and get the corresponding configuration (CFG) and weight files. The sparse factor defaults to 0.001, and the sparse factor can be adjusted appropriately according to the data set category distribution size, mAP, and BN distribution. The default value is used here. Command-prune 0 is suitable for the second channel pruning strategy, and command-prune 1 applies to other pruning strategies.

In this experiment, three models of YOLOv3, YOLOv3-spp, YOLOv3-tiny are selected for comparative experiments. After training, it is found that the performance of YOLOv3 is the best for 60 times of training, 66 times for YOLOv3-spp training, and 60 times for YOLOv3-tiny training. The next step is to select the compression pruning strategy. Three compression pruning strategies are used to compress the pruning models. After pruning with the results as shown in Tables 2—4, for the YOLOv3 and YOLOv3-spp model, it can be found that the effect of the second pruning strategy is relatively the best, and the loss of precision rate P and mAP loss is relatively small after compression, so we choose the second pruning strategy for experiment. For the YOLOv3-tiny model, we can only use the first pruning strategy, so choose it for experiment. In this paper, the models are pruned three times for experiment.

Table 2 Parameters of YOLOv3 model

Parameter	Original model	Pruning strategy	Pruning once	Pruning twice	Pruning three times
Size/MB	469	Pruning strategy 1	52.3	27.8	24.6
		Pruning strategy 2	37.00	6.09	0.97
		Pruning strategy 3	35.20	11.20	4.39
mAP	0.464	Pruning strategy 1	0.463	0.464	0.461
		Pruning strategy 2	0.463	0.463	0.463
		Pruning strategy 3	0.463	0.462	0.463
P	0.589	Pruning strategy 1	0.552	0.471	0.445
		Pruning strategy 2	0.563	0.582	0.585
		Pruning strategy 3	0.551	0.472	0.492

Table 3 Parameters of YOLOv3-tiny model

Parameter	Original model	Pruning strategy	Pruning once	Pruning twice	Pruning three times
Size/MB	66.3	Pruning strategy 1	33.70	15.65	6.16
mAP	0.574	Pruning strategy 1	0.551	0.551	0.553
P	0.727	Pruning strategy 1	0.687	0.661	0.665

Table 4 Parameters of YOLOv3-spp model

Parameter	Original model	Pruning strategy	Pruning once	Pruning twice	Pruning three times
Size/MB	477	Pruning strategy 1	53.6	26.4	23.2
		Pruning strategy 2	38.70	6.65	1.33
		Pruning strategy 3	36.40	10.90	3.68
mAP	0.38	Pruning strategy 1	0.243	0.242	0.243
		Pruning strategy 2	0.357	0.357	0.359
		Pruning strategy 3	0.318	0.316	0.319
P	0.667	Pruning strategy 1	0.009 81	0.009 61	0.009 61
		Pruning strategy 2	0.667	0.669	0.655
		Pruning strategy 3	0.666	0.670	0.665

The PaddlePi-K210 terminal core computing module is a terminal computing module custom developed for the Baidu PaddlePaddle platform. It has compact appearance and superior performance, and can be used in AI core operation processing unit. The core processing AI chip is kendryte K210. The chip supports models within 5 m in size, and embeds the model after three prunings and the modified recognition program into the chip, which can realize the lightweight recognition of the camera.

3.2 Processing results with non-maximum suppression

Since it is observed that the model has no better effect on image detection than video detection, we consider performing non-maximum suppression processing on the detection results again. Because the displacement of the object is not large in one second, we superimpose the obtained detection results of consecutive frames, then perform non-maximum suppression deduplication processing, and finally retain the results with higher confidence as the results of the next consecutive frames, and different IOU thresholds are set for each type to ensure the best detection effect. Thus we optimized the video detection effect.

3.3 Calculating prior probability

In the experiment, 15 groups of videos were selected to test the original model, which are from the 209th station of Guiyang Airport with an average duration of 8 min, the frame rate is 24 frame per second. So n=15, and the time and space probability distribution of the target object on the apron can be obtained. For any input = (x_1, y_1, x_2, y_2, cls, t), the object can be queried in the txt file at this time and the position is num, so the prior probability $P(A)$ can be obtained by the formula

$$P(A)=\left\{\frac{\text{num}}{n}\middle|1\leqslant \text{num}\leqslant n\right\}$$

For example:

For input = (369, 66, 916, 360, Plane, 30), we can find output = (368, 72, 923, 372, Plane, 30, 12). So we can get num = 12, the corresponding prior probability $P(A)_{\text{YOLOv3}}=0.8$. Similarly, we can get the corresponding prior probability $P(A)_{\text{YOLOv3-spp}}$ and $P(A)_{\text{YOLOv3-tiny}}$.

3.4 Calculating posterior probability

By testing the model after compression, the accuracy of the compressed model can be obtained. The accuracy of the compressed YOLOv3 model $P(A|B)_{\text{YOLOv3}}=0.585$, and the false positive rate $P(C)_{\text{YOLOv3}}$=0.486. Similarly, we can get

$P(A|B)_{\text{YOLOv3-spp}}=0.655$, $P(C)_{\text{YOLOv3-spp}}=0.301$

$P(A|B)_{\text{YOLOv3-tiny}}=0.675$, $P(C)_{\text{YOLOv3-tiny}}=0.243$

Therefore, using the above example, the posterior probability of YOLOv3 $P(B|A)_{\text{YOLOv3}}=0.828$ can be obtained from the formula. Similarly, we can get

$$P(B|A)_{\text{YOLOv3-spp}}=0.896$$
$$P(B|A)_{\text{YOLOv3-tiny}}=0.917$$

We set the threshold to 0.4, and output the bounding box when the posterior probability is greater than the threshold. If it is less than the threshold, the bounding box will be discarded.

3.5 Result comparison analysis

After the experiment, we can find that for any bounding box detected by the compression model, it

will be filtered twice. Firstly, it will be compared with the coordinates in the prior time and space distribution of the original model. If no match is found, it will be considered it is a false detection of the compression model, which sets the prior probability to a very small value. Thus eliminate the situation that the compression model is detected but all prior time positions of the original model are absent and reduce the precise loss of the model caused by compression. Secondly, if a match is found in the prior distribution coordinates, we set an appropriate threshold value by combining the distribution conditions. If the posterior probability is less than the threshold value, according to the prior service rule information on the frame number and position, we can know that although the bounding box appears in the detection results of the original model, the probability of occurrence is very small, and it is also considered as a false detection condition. In this way, we can eliminate some false detection of the original model by using the prior probability of time position distribution, and improve the accuracy of the compression model.

For the same two frames to three models, the experimental results are shown in Figs.4—21.

(1) Original model detection effects

Original YOLOv3 model detection effects are shown in Figs.4—5.

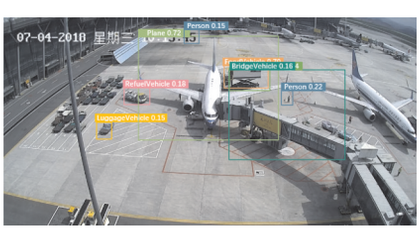

Fig.4 Original YOLOv3 model for Frame 1

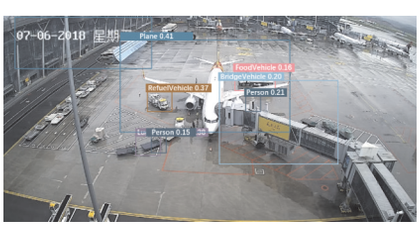

Fig.5 Original YOLOv3 model for Frame 2

Original YOLOv3-spp model detection effects are shown in Figs.6—7.

Original YOLOv3-tiny model detection effects are shown in Figs.8—9.

Fig.6 Original YOLOv3-spp model for Frame 1

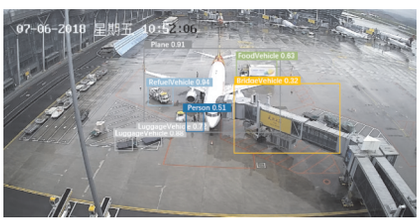

Fig.7 Original YOLOv3-spp model for Frame 2

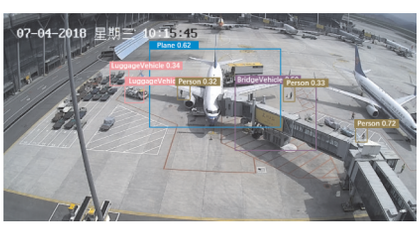

Fig.8 Original YOLOv3-tiny model for Frame 1

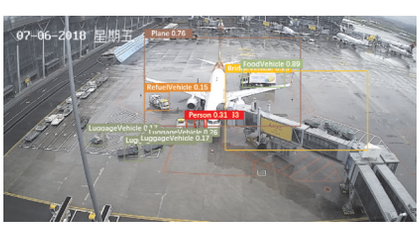

Fig.9 Original YOLOv3-tiny model for Frame 2

(2) Model detection effects after pruning

YOLOv3 model detection effects after pruning are shown in Figs.10—11.

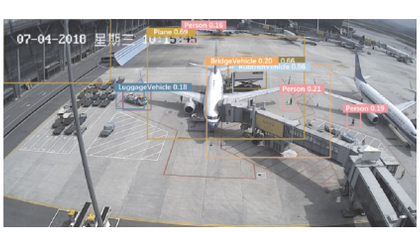

Fig.10 Pruning YOLOv3 model for Frame 1

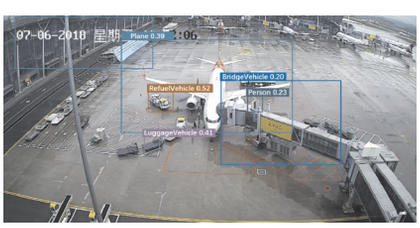

Fig.11 Pruning YOLOv3 model for Frame 2

YOLOv3-spp model detection effects after pruning are shown in Figs.12—13.

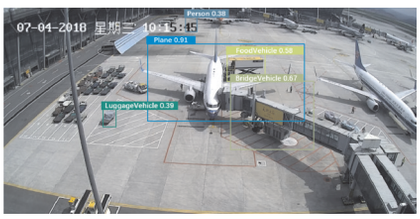

Fig.12 Pruning YOLOv3-spp model for Frame 1

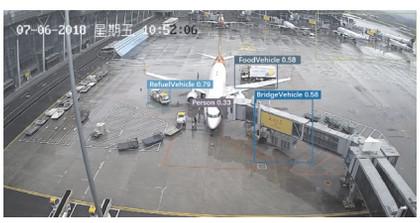

Fig.13 Pruning YOLOv3-spp model for Frame 2

YOLOv3-tiny model detection effects after pruning are shown in Figs.14—15.

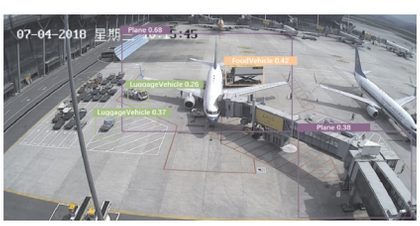

Fig.14 Pruning YOLOv3-tiny model for Frame 1

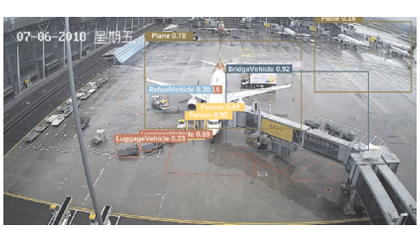

Fig.15 Pruning YOLOv3-tiny model for Frame 2

(3) Model detection effects after adding prior rules

YOLOv3 model detection effect after adding prior rules are shown in Figs.16—17.

Fig.16 YOLOv3 model with prior rules for Frame 1

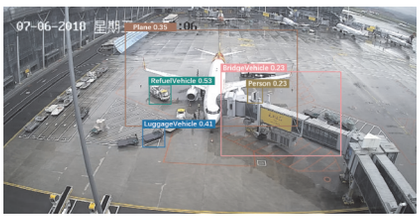

Fig.17 YOLOv3 model with prior rules for Frame 2

YOLOv3-spp model detection effects after adding prior rules are shown in Figs.18—19.

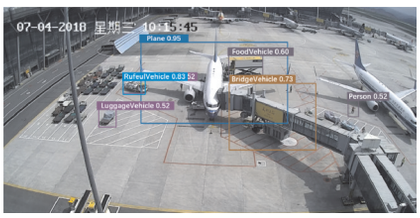

Fig.18 YOLOv3-spp model with prior rules for Frame 1

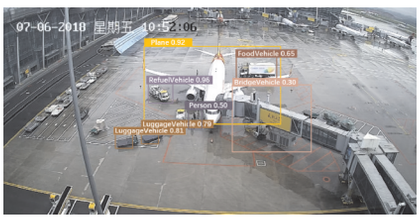

Fig.19 YOLOv3-spp model with prior rules for Frame 2

YOLOv3-tiny model detection effects after adding prior rules are shown in Figs.20—21.

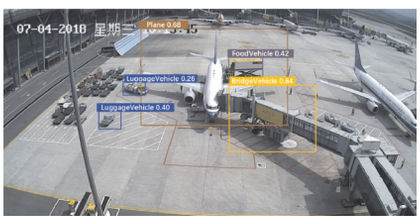

Fig.20 YOLOv3-tiny model with prior rules for Frame 1

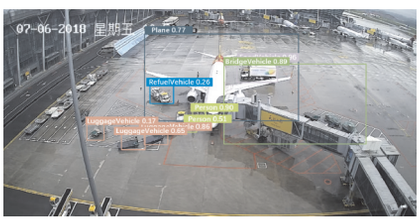

Fig.21 YOLOv3-tiny model with prior rules for Frame 2

It can be seen that after adding prior knowledge, the probability of false detection can be reduced, and the detection effect of the compressed model is no worse than that of the uncompressed

original model, and even some of the original model false detection can be eliminated through the prior knowledge. For the results of the three models, YOLOv3-spp and YOLOv3-tiny have relatively higher accuracy, and the final results are better than YOLOv3. The overall detection results of the models with semantic information are similar to those of the models before compression .

4 Conclusions

We propose a novel deep neural network compression model for airport object detection. The model obtains temporal and spatial semantic rules from the uncompressed parameter model, adds these prior knowledge to the compression model, and calculates prior probability and posterior probability to assist the detection, which realizes a semantic compression model with prior rules. Through experiments, we can see that the model with prior rules can make up for the loss caused by parameter compression, and can filter out some false detection of parameter compression model by using prior rules.

References

[1] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Columbus, OH: IEEE, 2014: 580-587.

[2] GIRSHICK R. Fast R-CNN[C]//Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV). Washington DC: IEEE, 2015: 1440-1448.

[3] HE K M, GKIOXARI G, DOLLAR P, et al. Mask R-CNN[J]. IEEE Transactions On Pattern Analysis and Machine Intelligence, 2020, 42(2): 386-397.

[4] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards realtime object detection with region proposal networks[C]//Proceedings of Conference and Workshop on Neural Information Processing Systems. Montreal, Canada: [s.n.], 2015: 91-99.

[5] LI Z, PENG C, YU G, et al. Light-head R-CNN: In defense of two-stage object detector[C]//Proceedings of Conference on Computer Vision and Pattern Recognition. [S.l.]:[s.n.], 2018.

[6] DAI J, LI Y, HE K, et al. R-FCN: Object detection via region-based fully convolutional networks[C]// Proceedings of Conference and Workshop on Neural Information Processing Systems. Barcelona, Spain: Curran Associates Inc, 2016: 379-387.

[7] REDMON J, FARHADI A. YOLO9000: Better, faster, stronger[C]//Proceedings of Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017: 7263-7271.

[8] LONG Xin, BEN Zongcheng, LIU Yan. A survey of related research on compression and acceleration of deep neural networks[J]. Journal of Physics: Conference Series, 2019, 1213(5): 052003.

[9] JIANG Chunhui. Research on deep neural network pruning method[D]. Hefei: University of Science and Technology of China, 2019. (in Chinese)

[10] LIU Z, LI J, SHEN Z, et al. Learning efficient convolutional networks through network slimming[C]// Proceedings of International Conference On Computer Vision. Venice: IEEE, 2017: 2755-2763.

[11] GAO Q, ZHAO Y, LI G, et al. Compression method of super-resolution convolutional neural network based on knowledge distillation[J]. Computer Applications, 2019, 39(10): 2802-2808.

[12] ZHOU Yang. Research on accelerating method of neural network parameter compression and inference[D]. Beijing: University of Chinese Academy of Sciences, 2020. (in Chinese)

[13] LI L, WANG L, CHEN C. A semantic compression scheme for digital images based on vector quantization and data hiding[J]. Multimedia Tools and Application, 2017 ,76(20): 20833-20846.

[14] WANG Huadong. Research on target tracking strategy based on target prior knowledge[D]. Huhehaote: Inner Mongolia University, 2019. (in Chinese)

Author Dr. **LYU Zonglei** was born in Tianjin, China, in 1981. He received the B.S. degree in computer science and technology from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2004, and the Ph.D. degree in computer application technology from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2009. In 2009, he joined the College of Computer Science and Technology, Civil Aviation University of China, as a Lec-

turer, and in 2012 became an associate professor. His current research interests include machine learning, deep learning, object detection.

Author contributions Dr. **LYU Zonglei** designed the study, complied the models, conducted the analysis. Ms. **PAN Fuxi** conducted the analysis, interpreted the results and wrote the manuscript. Ms. **XU Xianhong** contributed to the discussion and background of the study, and modified the manuscript. All authors commented on the manuscript and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: XU Chengting)

一种新的机场目标检测深度神经网络压缩模型

吕宗磊[1,2], 潘芙兮[1,2], 徐先红[1,2]

(1.中国民航信息技术科研基地,天津300300,中国; 2.中国民航大学计算机科学与技术学院,天津300300,中国)

摘要:提出了一种新的机场目标检测的深度神经网络压缩模型。该模型针对深层神经网络模型复杂、计算量大的缺点,根据机场目标检测的需要,从未压缩的模型中提取时空语义规则。通过将这些空间语义规则加入参数压缩后的模型中,以辅助检测。该规则可以提高检测模型的精度,弥补参数压缩带来的损失。实验表明,这种新的压缩检测模型压缩模型的效果并不比未压缩的原始模型差,甚至可以通过先验知识来消除一些原始模型的错误检测。

关键词:压缩模型;语义规则;剪枝;先验概率;轻量级检测