

## Research Article

# E-Commerce Picture Text Recognition Information System Based on Deep Learning

Bin Zhao <sup>1</sup>, WenYing Li,<sup>1</sup> Qian Guo,<sup>1</sup> and RongRong Song<sup>2</sup>

<sup>1</sup>School of Economics and Management, Bengbu University, Bengbu, Anhui 233030, China

<sup>2</sup>Anhui Radio and Television University, Hefei Branch, Hefei, Anhui 230001, China

Correspondence should be addressed to Bin Zhao; zhaobin@bbc.edu.cn

Received 4 November 2021; Revised 7 December 2021; Accepted 16 December 2021; Published 3 January 2022

Academic Editor: Bai Yuan Ding

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

For the accuracy requirements of commodity image detection and classification, the FPN network is improved by DPFM ablation and RFM, so as to improve the detection accuracy of commodities by the network. At the same time, in view of the narrowing of channels in the application of traditional MWI-DenseNet network, a new GTNet network is proposed to improve the classification accuracy of commodities. The results show that at different levels of evaluation indexes, the dpFPN-Netv2 algorithm improved by DPFM + RFM fusion has higher target detection accuracy than RetinaNet-50 algorithm and other algorithms. And the detection time is 52 ms, which is significantly lower than 90 ms required for RetinaNet-50 detection. In terms of target recognition, compared with the traditional MWI-DenseNet neural network, the computation amount of the improved MWI-DenseNet neural network is significantly reduced under different shunt ratios, and the recognition accuracy is significantly improved. The innovation of this study lies in improving the algorithm from the perspective of target detection and recognition, so as to change the previous improvement that only can be made in a single way.

## 1. Introduction

With the maturing of artificial intelligence and image processing technology, the unmanned sales are changing the Commodity sales model. It can be seen that the image detection and classification methods are widely used as an important technical support for unmanned sales. Among them, the target detection can be divided into two-stage detection method based on target region and end-to-end one-stage detection method [1, 2]. Where, the detection methods based on target region include Faster-RCNN [3], etc., and the end-to-end detection methods include You Only Live Once (YOLO), Single Shot MultiBox Detector (SSD), RetinaNet, etc. [4–8]. Segmentation model can be divided into semantic segmentation and instance segmentation. Here, the semantic segmentation models include SegNet, etc. [9–11], and the instance segmentation models include Mask-RCNN, etc. [12, 13]. With the wide application of deep learning, on the basis of target detection, the neural networks such as Convolutional Neural Networks,

VGGNet, etc., are applied to classification. But in the application of neural networks, increasing the network layers and amplifying the channels in the network feature graph, the effect of “widening and deepening” of convolutional neural network can be realized. Although it can greatly improve the network performance, it is at the cost of increasing the parameters and increasing the computations. This makes network deployment difficult and reduces the timeliness of network applications. In the context of network lightweight, MWI-DenseNet can repeatedly reference the feature maps by virtue of the built-in multipath dense connection mechanism. Thus, it greatly reduces the computations and parameters. And on the premise of ensuring classification accuracy, it greatly improves network efficiency and network adaptability. The dense connection mechanism has the advantage of feature reuse, but it also causes different layers to repeatedly reference the gradient information of the same layer for parameter optimization, which results in poor optimization effect of network learning. The network update relies heavily on the deep gradient

information. But under the dense connection mechanism, the deep gradient information mainly comes from the shallow repeated gradient information. This will inevitably affect the deep learning ability. In this regard, CSPNet further enriches the gradient propagation path by crossing the stage path to avoid the repeated reference of the same gradient information [14].

Inspired by this, this paper uses the deep gradient flow truncation strategy to improve the multipath integration dense connection block and transition layer structure, thus creating a new gradient truncation convolutional network (GTNet). It has obvious advantages, such as less computation, high classification accuracy, large gradient difference, and so on.

## 2. Commodity Inspection Based on FPN

The fusion feature pyramid convolution network constructed in this paper takes DenseNet as the backbone network. It defines the feature graph of some layers of DenseNet as the original feature pyramid [15–18]. Then the DPFM module is used to implement the fusion of the three layer feature maps, which are a certain feature diagram, the front and rear layer feature diagram in the original feature pyramid. Under the participation of more DPFM modules, each predict layer gets enough semantic information and precise location information, which can accurately identify objects of all sizes.

In addition, the process reuse feature diagram shows that the calculation amount of feature fusion is greatly reduced. And the efficiency of target detection is improved effectively. First of all, the DPFM module is used to build FPN-Net v1. Then, the RFM module is used to carry on the feature fusion of FPN-Net v1. Finally, the dual-path fusion feature pyramid convolution network FPN-Net v2 was constructed. The above process is shown in Figure 1.

*2.1. Dual-Path Feature Fusion Module.* Figure 2 shows the structure of the DPFM module.

The traditional feature fusion module FPN only has a top-down fusion path. So it can only realize the feature fusion between the previous layer and the current layer. Due to the low reuse rate of feature layer, the fusion of multiple layer is required, which can increase the amount of computation. What's more, it is difficult to fully retain the location information. Different from FPN module, the DPFM module realizes feature fusion of previous layer  $(i + 1) - th$  layer and current layer  $(i - 1) - th$  layer through top-down fusion path. And it can carry out the feature fusion of latter layer  $(i - 1) - th$  layer and current layer  $(i - 1) - th$  layer through bottom-up fusion [19].

The multiple layer feature diagrams are fused by multiple DPFM loops, which makes DPFM + RFM fusion feature pyramid detection network formed. It is shown as Figure 3.

Figure 3 shows the two fusion. Where, the red line represents the top-down fusion path, and the blue line represents the bottom-up fusion path. Specifically, in the red line, the feature graph of previous layer  $(i + 1) - th$  is input.

After  $1 \times 1$  convolution, the number of output channels is consistent with the feature graph of current layer  $(i) - th$ . Then, the bilinear interpolation algorithm is introduced to perform Upsample and  $(i + 1)$  fusion input is obtained. In the blue line, the feature graph of later layer  $(i - 1) - th$  is input. After  $1 \times 1$  convolution, the number of output channels is consistent with the feature graph of current layer  $(i) - th$ . Then, the pass through algorithm is introduced to perform Downsample and  $(i - 1)$  fusion input is obtained. After that,  $(i - 1)$  fusion input  $(i - 1) - th$  and  $(i) - th$  are spliced in the channel dimension. The spliced feature graph is performed with  $3 \times 3$  convolution, and the final fusion feature layer can be obtained. The number of size channel is consistent with  $(i) - th$  [20–23].

In Figure 3, the DPFM module corresponding to the 4-th layer fuses the front 3-th layer, the current 4-th layer and the rear 5-th layer. The fusion result is used as the input of the top-down fusion path of the DPFM module corresponding to the 3-th layer. At the same time, the 2-th layer is used as the input of the bottom-up fusion path of the DPFM module corresponding to the 3-th layer, and the fusion feature operation is repeated. The multiplex fusion feature mechanism shown in the aforementioned process greatly expands the position information and semantic information of the prediction layer. The fewer original feature layer in the built-in Back Bone not only facilitates module deployment but also reduces the amount of computation. The bidirectional fusion of deep and shallow information is repeated. First of all, it can alleviate the problem of feature information loss caused by pooling. Then, the rich semantic information and accurate location information to the prediction layer can be provided. Finally, the target detection can be precisely achieved. From the application effect, the detection accuracy of cyclic multiplexing bidirectional feature fusion pyramid network is relatively ideal. Especially, it can accurately detect small size objects, and it requires less parameter amount and calculation amount. Thus, the detection efficiency can be improved.

*2.2. Recombination and Fusion Module.* After introducing the recombination and fusion module, the semantic characterization capability of feature pyramid can be further enhanced, and the consequence of feature loss caused by pooling can be alleviated.

Analysis is made based on Figure 4. First, the pass through method is used to split the input feature map of the latter layer  $(i - 1) - th$  layer into four sub-blocks with the same size as the current layer  $(i) - th$  layer. They are spliced in the channel dimension in a fixed order. Then, the splicing results are reconstructed and spliced with the current layer  $(i) - th$  layer, and the  $1 \times 1$  convolution is performed.

The input offset will cause significant changes to pooled output, as shown in Figure 5.

This shows that when the input feature graph shifts to the right by 1 feature point, the maximum pooling result will change significantly before and after the migration. For example, the input  $(i) - th$  layer is  $[0, 0, 1, 1, 0, 0, 1, 1]$ . Its average difference is divided into two sub-blocks  $[0, 1, 0, 1]$

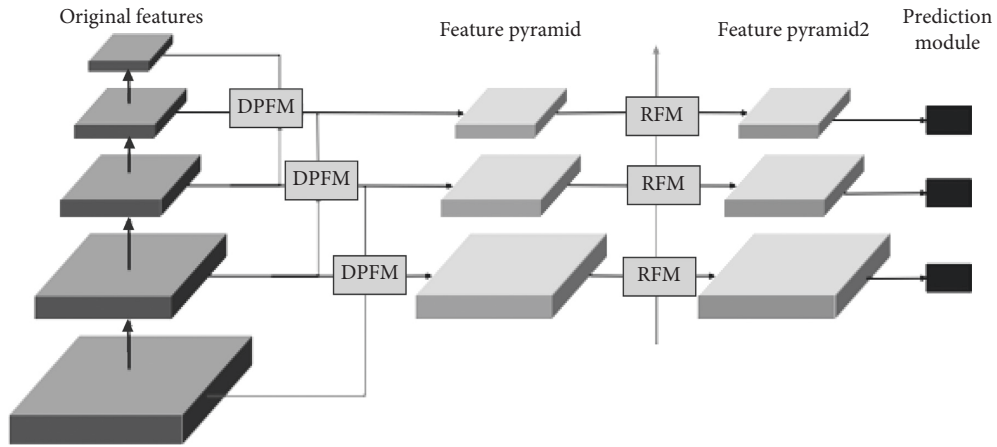


FIGURE 1: DPFM + RFM fusion feature pyramid convolution network structure.

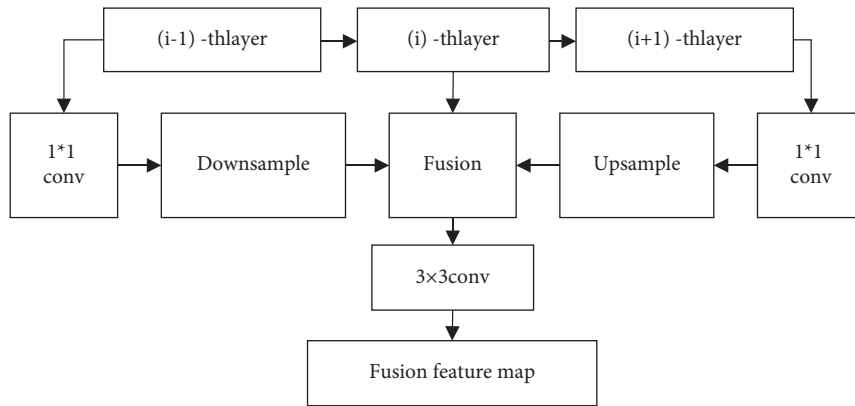


FIGURE 2: Feature fusion module.

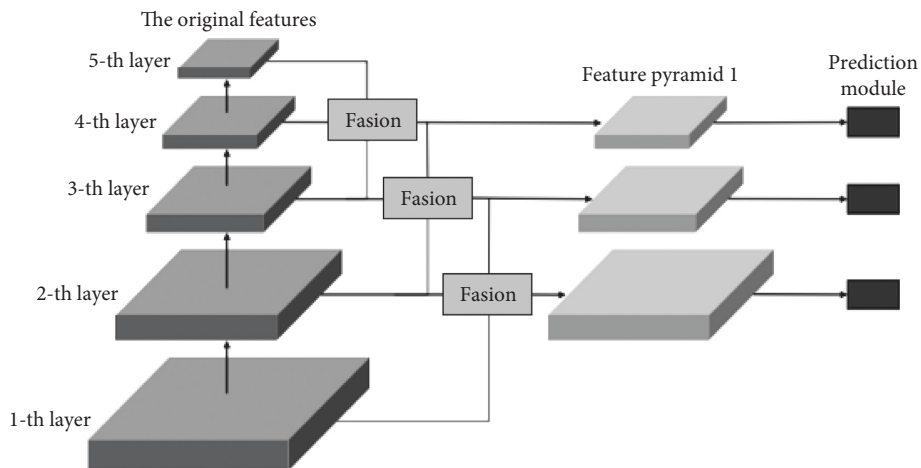


FIGURE 3: DPFM + RFM fusion feature pyramid detection network.

and  $[0, 1, 0, 1]$ . The input before the offset is  $[1, 1, 1, 1]$ . What's more, the input after the offset is  $[0, 1, 0, 1]$ . Hence, the convolution kernel parameter is set to be equal to 1. Before the offset, perform  $1 \times 1$  convolution for  $[0, 1, 0, 1]$ ,  $[0, 1, 0, 1]$  and  $[1, 1, 1, 1]$  to obtain  $[1/3, 1, 1/3, 1]$ ; after the

offset, perform  $1 \times 1$  convolution for  $[0, 1, 0, 1]$ ,  $[0, 1, 0, 1]$  and  $[0, 1, 0, 1]$  to obtain  $[0, 1, 0, 1]$ . It can be seen that the convolution fusion results corresponding before and after the offset are closer. If the convolution core parameters are trained and optimized, the difference between the

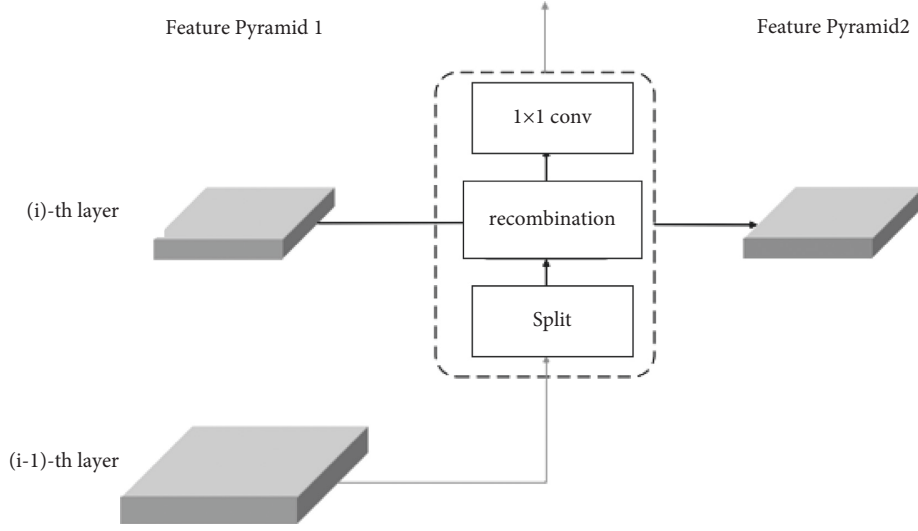


FIGURE 4: Recombination fusion module.

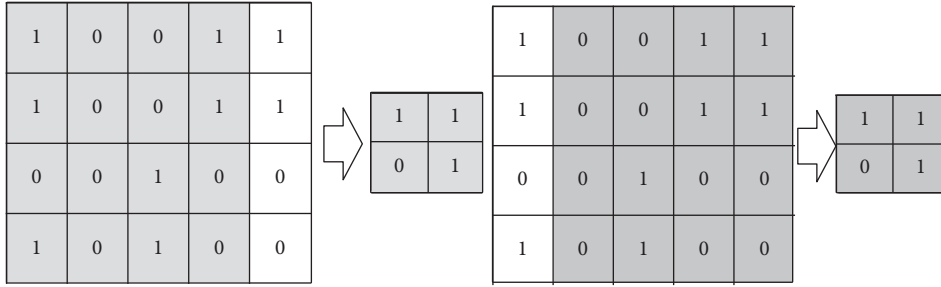


FIGURE 5: Schematic diagram of input offset changing pooled output.

convolution fusion results before and after the offset can be further reduced. The convolution fusion method proposed in this paper can suppress the interference of pooling to the detection accuracy and achieve higher detection accuracy with fewer parameters and operations.

The RFM module is deployed on the DPFPN-Net v1 architecture to construct a three-feature pyramid structure network DPFPN-Netv2.

### 3. Commodity Classification Based on Gradient Truncation Convolutional Network

On the basis of Multiway Integrated Dense Connection Convolutional Network(MWI-DenseNet), the connection blocks and transition layers are optimized to create a new GTNet, and the institutional differences are shown in Figure 6 below:

In the optimized structure, GTNet split the input feature graph  $x_0$  on the channel dimension to  $x'_0$  and  $x''_0$ , and the ratio of the latter is the resolution ratio  $\beta$ . After that,  $x''_0$  was input into the MWI-Dense block [24–26]. After completing the n-layer convolution calculation, the dense combination feature graph  $x_n$  was finally output, which was input into the first transition layer to obtain the output value  $x_t$ . It was used

to connect to  $x'_0$  and then input it into the second transition layer to finally obtain the fusion output  $x_c$ .

Taking the weight of convolution kernel as an example and excluding other parameters, the extraction features of MWI-DenseNet forward propagation and parameter optimization process of back propagation are analyzed. The mathematical expression of forward propagation process of the transition layer and each layer in the original dense connection block is as follows:

$$x_1 = H_1(W_1, x_0), \quad (1)$$

$$x_2 = H_2(W_2, [x_0, x_1]), \quad (2)$$

$$x_2 = H_3(W_3, [x_0, x_1, x_2]). \quad (3)$$

Here,  $x_i$  is the output characteristic graph of the  $i$ th layer,  $H_i(\cdot)$  represents the linear transformation of the  $i$ -th input, and  $W_i$  refers to the weight set of the convolution kernel at the  $i$ -th layer.

Input the combination feature graph  $x_n$  into the transition layer to obtain the output value  $x_t$  of the current stage:

$$x_t = H_t(W_t, [x_0, x_1, x_2, \dots, x_{n-1}, x_n]). \quad (4)$$

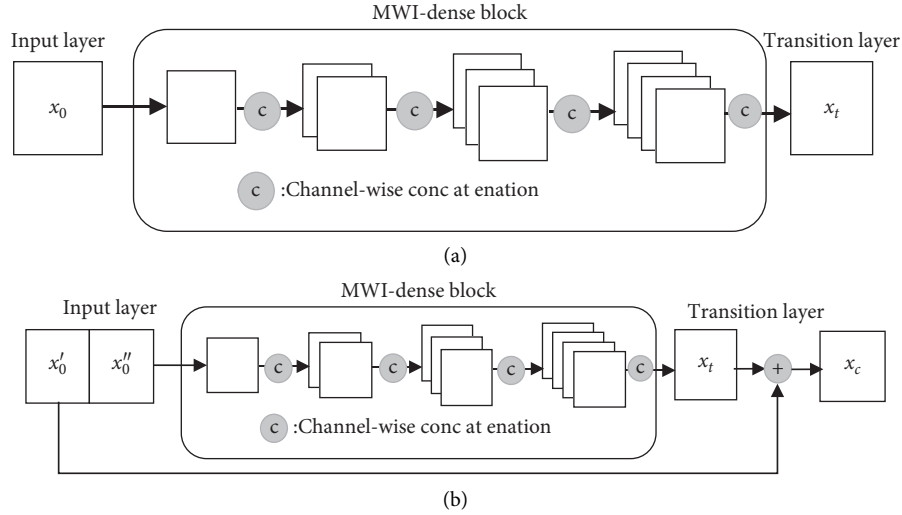


FIGURE 6: GTNet and MWI-DenseNet connect blocks with transition layer structures. (a) Structure of MWI-DenseNet connection block and transition layer. (b) Structure of GTNet connection block and transition layer.

The update process of back propagation of each layer weight is expressed as follows:

$$W'_1 = f_{bp}(W_1, g_0), \quad (5)$$

$$W'_2 = f_{bp}(W_2, g_0, g_1), \quad (6)$$

$$W'_3 = f_{bp}(W_3, g_0, g_1, g_2), \quad (7)$$

$$W'_n = f_{bp}(W_n, g_0, g_1, g_2, \dots, g_{n-1}), \quad (8)$$

where  $g_i$  is the gradient set of loss function about weight  $W_i$ ,  $f_{bp}(\cdot)$  is the weight optimization algorithm updated by back propagation, and  $W'_i$  is the weight after completing a round of update.

Weight updating algorithm of the transition layer is as follows:

$$W'_t = f_{bp}(W_n, g_0, g_1, g_2, \dots, g_{n-1}, g_n). \quad (9)$$

If the network depth reuses shallow gradient information to update the weight, it will constrain the efficiency of network operation and affect the ability of network learning.

In the optimized structure, the forward propagation process of dense block and two-layer transition layer is shown as follows:

$$x_n = H_n(W_n, [x''_0, x_1, x_2, \dots, x_{n-1}]), \quad (10)$$

$$x_n = H_n(W_t, [x''_0, x_1, x_2, \dots, x_{n-1}, x_n]), \quad (11)$$

$$x_c = H_t(W_c, [x'_0, x_t]). \quad (12)$$

The dense block and the first transition layer perform the back propagation process, which inputs  $g''_0$  to replace  $g_0$ , and the weight back propagation process is shown as follows:

$$W'_n = f_{bp}(W_n, g''_0, g_1, g_2, \dots, g_{n-1}), \quad (13)$$

$$W'_t = f_{bp}(W_n, g''_0, g_1, g_2, \dots, g_{n-1}, g_n), \quad (14)$$

$$W'_c = f_{bp}(W_c, g'_0, g_t). \quad (15)$$

The synchronous comparison shows that the gradient information of updating  $W'_t$  and  $W'_t$  is not consistent, which indicates that the transition layer is split into two parts, and the shallow layer gradient information cannot enter the deep layer for weight update.

#### 4. The Commodity Image Detection and Classification Model Constructed in This Study

The commodity image detection and classification model constructed in this paper based on the aforementioned methods is shown in Figure 7.

The process of Commodity inspection and classification is listed as follows:

Step 1: initialize the parameter weight  $W$  and bias  $b$ .

Step 2: first, the input conv6\_1, conv6\_2 and conv6\_3 were carried on DPFM fusion. The output fusion feature graph was used as the input of the second DPFM. Then, the input conv6\_3 and conv5\_8 were combined to perform DPFM fusion again, and a 4-layer fusion feature pyramid 1 was constructed.

Step 3: fusion feature pyramid 1 through the bottom-up fusion path to achieve RFM cycle fusion, so as to build fusion feature pyramid 2.

Step 4: the prediction layer solves the predicted value and determines the loss situation;

Step 5: back propagation, updates parameter weight  $w$  and bias  $b$ ;

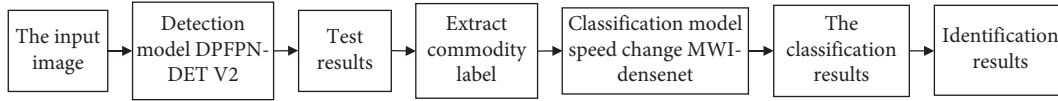


FIGURE 7: Block diagram of commodity identification algorithm.

Step 6: repeat the above steps until the set iterative termination times  $T$  is reached;

Step 7: show the prediction results in the output picture, including the bounding box and the bounding box confidence.

Step 8: combined with the detected images, the new GTNet network is used to classify the commodity images, and the classification results are output.

## 5. Experimental Verification

**5.1. Construction of Experimental Environment.** This experiment is carried out on the Ubuntu platform of Linux operating system, using TensorFlow framework and Python programming language. Training data are matched by an online enhancement method. After the completion of a round of training, different amplification methods are randomly referenced for data update, but there is no need to perform data amplification in the network test stage.

**5.2. Network Structure and Parameters.** The FPN-NET v2 backbone network structural parameters are listed in Table 1.

In each Conv layer, after convolution, it is connected with the BN layer and activation function layer (ReLU function is selected). Therefore, the convolution layer is equivalent to the combination layer of “convolution + BN + activation function.” By circularly fusing the original feature map of 6 layers with 4 DPFM, a fusion feature pyramid of 5 layers is constructed to construct FPN-Net v1. By circularly fusing feature pyramid 1 of 4 RFM, a fusion feature pyramid 2 of 5 layers is constructed to jointly build the FPN-Net v2.

**5.3. Commodity Inspection Results.** The Tensorboard tool is used to realize the visualization of network training-related indicators, and the change of Net v2 training loss value are shown in Figure 8.

It can be seen from the figure that in the process of increasing epoch, the loss value of network training slowed down, and the loss value stabilizes at 2.5, which means Net v2 training convergence.

In this experiment, the test data were divided into three categories: small  $\in (0, 128 \times 128]$ , media  $\in (128 \times 128, 256 \times 256]$ , large  $\in (256 \times 256, 512 \times 512]$ . The size of the test set was  $512 \times 512$ . The experimental results of FPN-Net v1, FPN-Net v2, RetinaNet detection algorithms are listed in Table 2.

According to the analysis of experimental results about the above table, all the test indexes of FPN-Net v2 are higher than other detection algorithms. If the  $AP_{0.5}$  is selected as the evaluation index, it can be found that the  $AP_{0.5}$  of FPN-Net v2 is 88.61. However, because the FPN-Net v1 is equipped with fusion mechanism, the detection accuracy of small size

objects is improved. On this basis, the FPN-net v2 uses the recombination fusion module to further improve the detection accuracy of small size objects.

The parameters and operation market of the aforementioned models are listed in Table 3.

Where the different circles correspond to different models and the radius of the circle is proportional to model  $AP_{0.5}$ . The larger the area of the circle and the closer the center of the circle to the upper left corner is, the faster the speed for the model to achieve higher detection accuracy is. On the whole, the circle corresponding to FPN-Net v1 model is closer to the upper left corner. Comparing the same ideas, the circle corresponding to the FPN-Net v2 model is closer to the upper left corner than the FPN-Net v1 model, which shows that the FPN-Net v2 model has advantages in terms of detection accuracy and inference speed.

**5.4. Commodity Classification Results.** The Tensorboard is used to realize the visualization of various indicators required by network training, and the split ratio  $\beta=0.5$  is set. The loss of network training is shown in Figure 9.

To analyze Figure 9, it can be seen that after the network training beginning, the epoch value is gradually increased, and the network loss is overall in a decreasing trend. However, the decreasing speed is slowing down, and it finally stabilizes around Loss = 0.9. At this point, the network training process has completed convergence.

The accuracy curve of network training verification set is shown in Figure 10.

Here, after several rounds of network training, the classification accuracy of the network for verification sets keeps increasing and finally stabilizes around 99%, which has reached the requirement. It indicates that the network has a good learning fit for training data.

To test the actual accuracy of the network, the labeled target area is cut out from the original image, and the pixel value of “0” is expanded to  $224 \times 224$ . Hence, the classification test data set was obtained. Considering that the original labeled area sizes of classified images are different, to understand the impact of different size targets on network performance, the sample images in the test data set are divided into three categories: small  $\in (0, 128 \times 128)$ , media  $\in (128 \times 128, 256 \times 256)$ , and large  $\in (256 \times 256, 512 \times 512)$ .

The partial classification test results of GTNet for classification test data sets are listed in Figure 11.

Here, when the split ratio  $\beta=0.30$ , the GTNet has the highest classification accuracy, and the classification accuracy of GTNet is higher than that of MWI-DenseNet for all kinds of test sets. It means that the deep gradient flow cut-off mechanism has the application advantages, and it can improve the classification accuracy of the network.

TABLE 1: FPN-NET v2 backbone network structural parameters.

Group	Block	Layer	Output
	Input		$512 \times 512 \times 3$
Conv1_x	Initializer block		$128 \times 128 \times 32$
conv2_x	MWI-dense block 1	Layer $\times 6$	$64 \times 64 \times 64$
	Transition layer	$1 \times 1$ conv; pooling; SE module	
conv3_x	MWI-dense block 2	Layer $\times 12$	$32 \times 32 \times 256$
	Transition layer	$1 \times 1$ conv; pooling; SE module	
conv4_x	MWI-dense block 3	layer $\times 24$	$16 \times 16 \times 256$
	Transition layer	$1 \times 1$ conv; pooling; SE module	
conv5_x	MWI-dense block 4	layer $\times 8$	$8 \times 8 \times 256$
	Transition layer	$1 \times 1$ conv; SE module	
	conv6_1	conv3 $\times 3, s=2$	$4 \times 4 \times 512$
conv6_x	conv6_2	conv3 $\times 3, s=2$	$2 \times 2 \times 512$
	conv6_2	conv3 $\times 3, s=2$	$1 \times 1 \times 512$

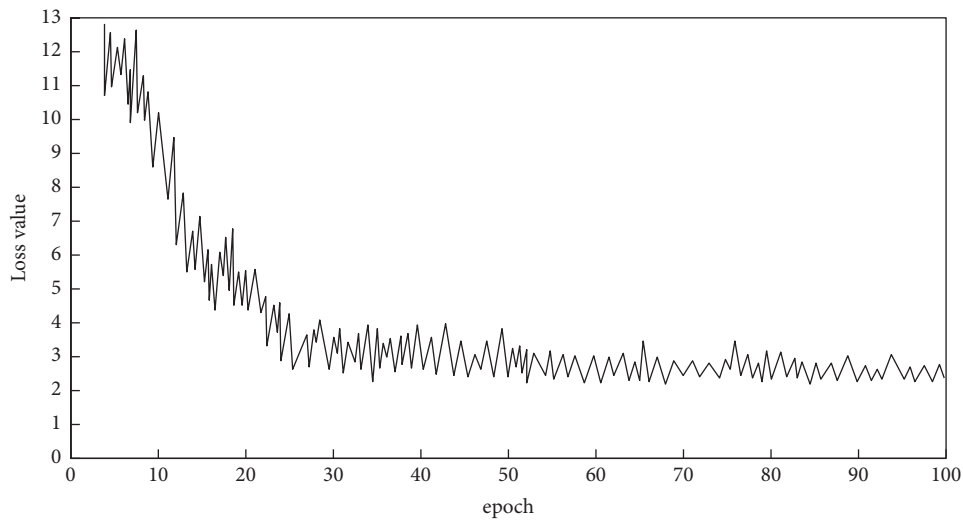


FIGURE 8: Loss value curve of network training process.

TABLE 2: Test results of FPN-Net and various test models (%).

Model	Backbone	AP <sub>0.5</sub>	AP <sub>0.7</sub>	AP <sub>0.9</sub>
RetinaNet-50	RetinaNet	86.05	65.37	45.02
FPN-Net v1	MWI-DenseNet	86.96	70.15	45.27
FPN-Net v2	MWI-DenseNet	88.61	72.03	45.92

TABLE 3: AP<sub>0.5</sub> (%), number of parameters and reasoning time (ms) for each model.

Model	Number of parameters	AP <sub>0.5</sub>	Inference time (ms)
RetinaNet-50	30.4 M	86.05	90
DPPFN-Netv1	25.7 M	86.96	51
DPPFN-Netv2	25.8 M	88.61	52

When the split ratio  $\beta = 0.50$ , the input feature graph is half-input into the connection block and the transition layer. At this time, the classification accuracy of GTNet for the four types of test sets is still higher than that of MWI-DenseNet, and the overall computation is reduced.

When the split ratio  $\beta = 0.70$ , more input feature maps are connected to the transition layer through identity mapping, which directly affects the test accuracy of GTNet. However, at the same time, the computation is greatly reduced, and the whole reasoning process is relatively fast.

In this paper, the Billion floating point operations (BFLOPs) are used to evaluate the network computation. The GTNet network test results are listed in Figure 12 below:

It can be seen that under the conditions of different splitting ratio, the computations of GTNet are less than that of MWI-DenseNet. With the change of splitting ratio, the computations and accuracy rate of GTNet will also change.

When the split ratio  $\beta = 0.30$ , compared with MWI-DenseNet, the computation of GTNet was reduced by 3.2%. When the split ratio  $\beta = 0.5$ , the computations of GTNet are



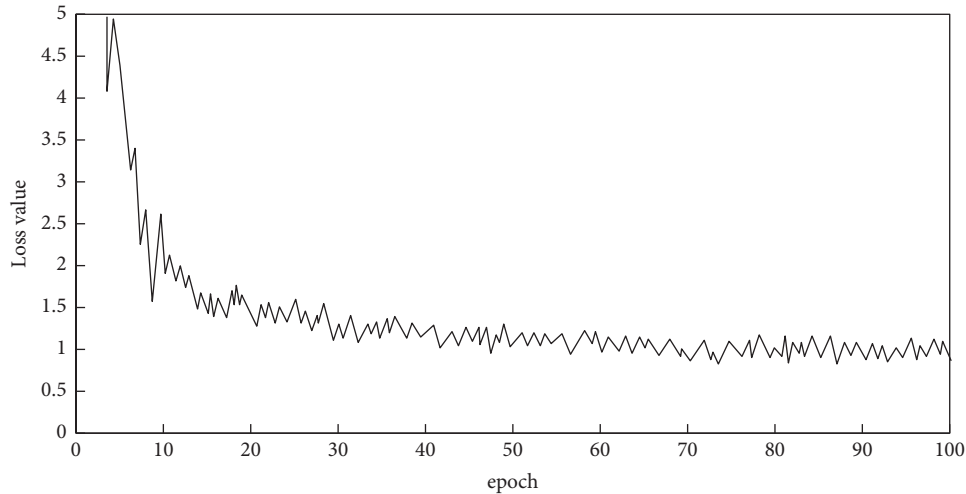


FIGURE 9: Loss value curve of network training process.

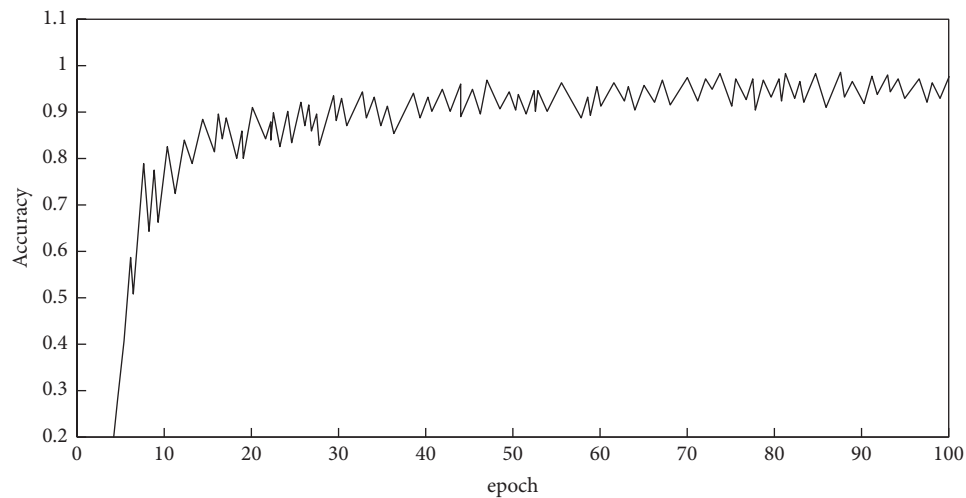


FIGURE 10: Network training process verification set accuracy curve.

12.7% less than that of MWI-DenseNet. When the split ratio  $\beta=0.70$ , only 30% of the feature graph is input to the connection block. Hence, the computations of GTNet are greatly reduced, which is 20.1% lower than that of MWI-DenseNet. It is also the lowest among all the schemes.

The horizontal coordinate represents the calculation amount, and the vertical coordinate represents the accuracy, so the coordinate system is established, as shown in Figure 13.

Obviously, the radius of the circle is proportional to the accuracy of the model. The larger the area of the circle and the closer the center of the circle to the upper left corner is, the fewer parameters required in the model reasoning process is. It means that the higher detection accuracy can be

achieved. Therefore, under any condition of the split ratio  $\beta$ , the comprehensive performance of GTNet is better than that of MWI-DenseNet. Namely, the GTNet has comparative advantages in calculation amount, parameters, and accuracy.

Here, the commodity images to be identified are first input into FPN-Net v2 model to obtain the commodity detection result output1. The target boundary box is marked in the output image, making the target area cut out. The size of the target area is expanded to  $224 \times 224$ . The new product target image is input into GTNet( $\beta=0.30$ ) model. Thus, the product classification result output2 is obtained. The product boundary box and product category are identified in the output image, as shown in Figure 14.



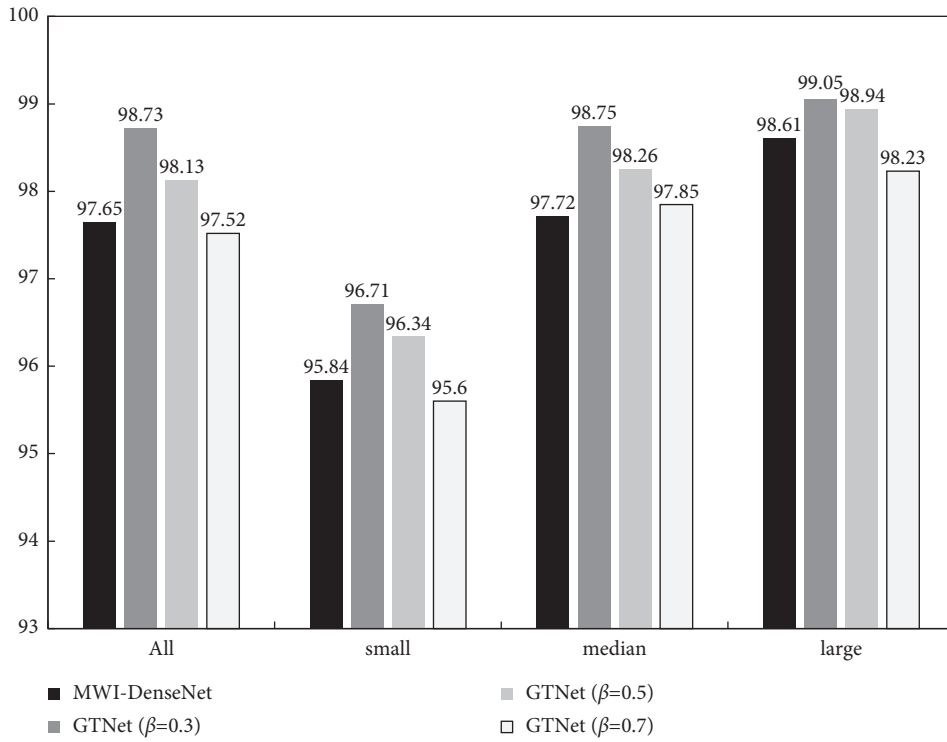


FIGURE 11: GTNet network test results (unit:%).

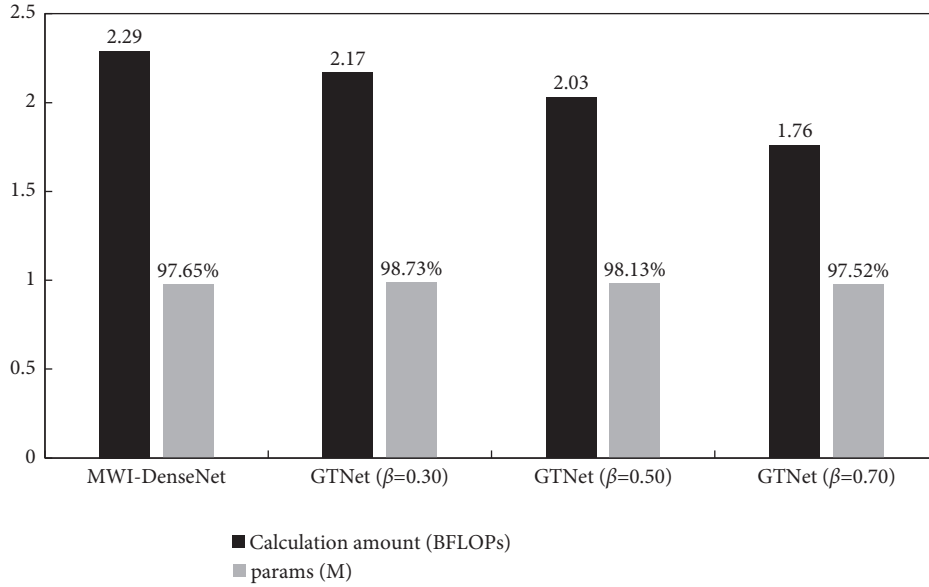


FIGURE 12: GTNet network test results (unit:%).

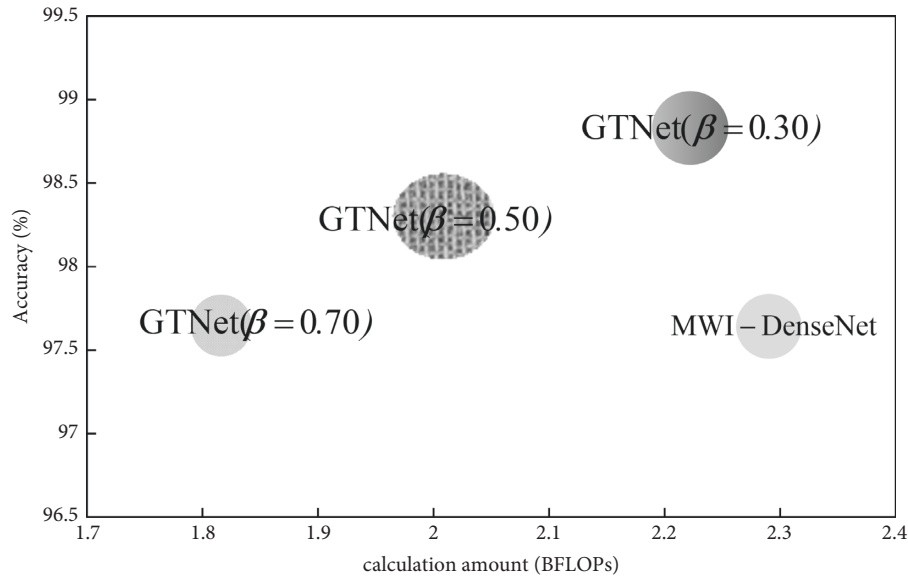


FIGURE 13: Scatter diagram of each model's accuracy and calculation amount.

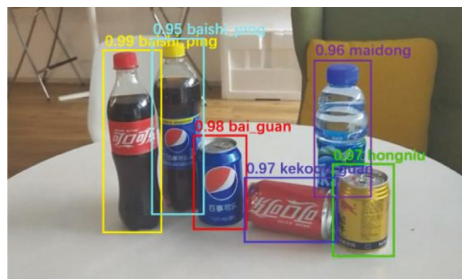


FIGURE 14: Image recognition results.

## 6. Conclusion

It can be seen from the research that the key problem in the commodity detection and classification is how to improve the structure of convolutional neural network. The accuracy of commodity detection and classification can be improved by constantly optimizing the structural parameters. However, the practice of this study also shows that the accuracy can reach more than 85% when the above method detects the commodities of different sizes, especially the small target commodities. Meanwhile, the amount of computation is smaller than other algorithms. However, there are limitations to the above research, that is, there is no preprocessing of image features.

## Data Availability

The experimental data are available upon request.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

The work is sponsored and financially supported by the Study of Cross-Border E-Commerce Promoting the transformation Upgrade of Manufacturers (SK2020A0545), Research on SME Cross-Border E-Commerce Marketing Channel Optimization Strategy with Bengbu Aote Machinery Co., Ltd as an example (2020SK02ZD) and the New Liberal Arts Research and Reform Practice Project of the Provincial Quality Engineering Marketing Major in Anhui Colleges and Universities (2020WYXM118).

## References

- [1] K. Hamid and R. Kazemi Matin, "Common set of weights and efficiency improvement on the basis of separation vector in two-stage network data envelopment analysis," *Mathematical Sciences*, vol. 14, no. 4, pp. 53–65, 2020.
- [2] S. Polepaka, C. S. Rao, and M. Chandra Mohan, "IDSS-based Two stage classification of brain tumor using SVM," *Health and Technology*, vol. 10, no. 1, pp. 249–258, 2020.
- [3] B. Liu, J. Luo, and H. Huang, "Toward automatic quantification of knee osteoarthritis severity using improved Faster

- R-CNN,” *International journal of computer assisted radiology and surgery*, vol. 15, no. 9, pp. 457–466, 2020.
- [4] G. Yin, M. Yu, M. Wang, Y. Hu, and Y. Zhang, “Research on highway vehicle detection based on faster R-CNN and domain adaptation,” *Applied Intelligence*, pp. 1–16, 2021.
- [5] B. Chen and X. Miao, “Correction to: distribution line Pole detection and counting based on YOLO using UAV inspection line video,” *Journal of Electrical Engineering & Technology*, vol. 15, no. 2, p. 997, 2020.
- [6] X. Hu, W. Luo, J. Hu et al., “Brain SegNet: 3D local refinement network for brain lesion segmentation,” *BMC Medical Imaging*, vol. 20, no. 10, pp. 17–111, 2020.
- [7] D. Tiede, G. Schwendemann, A. Alobaidi, L. Wendt, and S. Lang, “Mask R-CNN-based building extraction from VHR satellite data in operational humanitarian action: an example related to Covid-19 response in Khartoum, Sudan,” *Transactions in GIS*, vol. 25, no. 3, pp. 1213–1227, 2021.
- [8] X. Chen, R. Girshick, K. He, and P. Dollár, “TensorMask: a foundation for dense object segmentation,” in *Proceedings of the 2019 International Conference on Computer Vision*, Seoul, Republic of Korea, November 2019.
- [9] X. Chen, J. Du, and H. Zhang, “Lipreading with DenseNet and resBi-LSTM,” *Signal, Image and Video Processing*, vol. 14, pp. 1–9, 2020.
- [10] Y. Li, X. Xie, L. Shen, and S. Liu, “Reverse active learning based atrous DenseNet for pathological image classification,” *BMC Bioinformatics*, vol. 20, no. 8, pp. 445–354, 2019.
- [11] [Please provide complete details for this reference].
- [12] B. Ahmed, T. A. Gulliver and S. alZahir, “Image splicing detection using mask-RCNN,” *Signal, Image and Video Processing*, vol. 14, pp. 1–8, 2020.
- [13] T. Chen, Y. Jiang, W. Jian, L. Qiu, H. Liu, and Z. Xiao, “Maintenance personnel detection and analysis using mask-RCNN optimization on power grid monitoring video,” *Neural Processing Letters*, vol. 51, pp. 1–12, 2019.
- [14] C. Y. Wang, H. Liao, I. H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, “A new backbone that can enhance learning capability of cnn,” 2019, <https://arxiv.org/abs/1911.11929v1>.
- [15] X. Feng, H. Yao, and S. Zhang, “An efficient way to refine DenseNet,” *Signal, Image and Video Processing*, vol. 13, no. 5, pp. 959–965, 2019.
- [16] X. Feng, K. Xia, M. Yang et al., “Long range surface plasmon resonance sensor based on side polished fiber with the buffer layer of magnesium fluoride,” *Optical and Quantum Electronics*, vol. 49, no. 4, pp. 1–12, 2017.
- [17] Y. Guo, G. Ding, and J. Han, “Robust quantization for general similarity search,” *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 949–963, 2018.
- [18] X. Zeng, G. Feng, and X. Zhang, “Detection of double JPEG compression using modified DenseNet model,” *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8183–8196, 2019.
- [19] Y. Zhang, C. Qin, W. Zhang, F. Liu, and X. Luo, “On the fault-tolerant performance for a class of robust image steganography,” *Signal Processing*, vol. 146, pp. 99–111, 2018.
- [20] Y. Zhang, X. Luo, X. Zhu, Z. Li, and A. G. Bors, “Enhancing reliability and efficiency for real-time robust adaptive steganography using cyclic redundancy check codes,” *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 115–123, 2020.
- [21] X. Ning, K. Gong, W. Li, L. Zhang, X. Bai, and S. Tian, “Feature refinement and filter network for person re-identification,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 9, pp. 3391–3402, 2021.
- [22] X. Ning, K. Gong, W. Li, and L. Zhang, “JWSAA: joint weak saliency and attention aware for person re-identification,” *Neurocomputing*, vol. 453, pp. 801–811, 2021.
- [23] X. Ning, F. Nan, S. Xu, L. Yu, and L. Zhang, “Multi-view frontal face image generation: a survey,” *Concurrency and Computation: Practice and Experience*, vol. 3, 2020.
- [24] X. Ning, P. Duan, W. Li, and S. Zhang, “Real-time 3D face alignment using an encoder-decoder network with an efficient deconvolution layer,” *IEEE Signal Processing Letters*, vol. 27, pp. 1944–1948, 2020.
- [25] V. Sedighi, R. Cogranne, and J. Fridrich, “Fridrich. Content-adaptive steganography by minimizing statistical detectability,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2016.
- [26] M. Afif, R. Ayachi, Y. Said, E. Pissaloux, and M. Atri, “An evaluation of RetinaNet on indoor object detection for blind and visually impaired persons assistance navigation,” *Neural Processing Letters*, vol. 51, pp. 1–15, 2020.