A Multi-scale Radar HRRP Target Recognition Method Based on Pyramid Depthwise Separable Convolution Network

2022 7th International Conference on Image, Vision and Computing (ICIVC) | 978-1-6654-6734-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIVC55077.2022.9886642

Jiaxing He
Air and Missile Defense College
Air Force Engineering University
Xi'an, China
hejiaxing@stu.xidian.edu.cn

Xiaodan Wang
Air and Missile Defense College
Air Force Engineering University
Xi'an, China
afeu_wang@163.com

Yafei Song
Air and Missile Defense College
Air Force Engineering University
Xi'an, China
yafei_song@163.com

Qian Xiang
Air and Missile Defense College
Air Force Engineering University
Xi'an, China
qianxljp@163.com

***Abstract*—Radar high resolution range profile (HRRP) contains important structural features such as target size and scattering center distribution, which has attracted extensive attention in the field of radar target recognition. In order to solve the problem of feature extraction and recognition in HRRP target recognition, we propose an HRRP target recognition method based on one-dimensional Pyramid Depthwise Separable Convolutional (PyDSC) neural network. For the processed data, pyramid convolution is selected, and convolution kernels of different sizes are used on different input channels, which can better extract the features of different scales and improve the overall recognition ability. At the same time, Depthwise Separable Convolution (DSC) technology is applied to PyConv network, a standard convolution operation is divided into two steps: deep convolution and point convolution, which can reduce the network complexity, reduce the amount of parameters and improve the speed of HRRP target recognition. Finally, we verify the effectiveness of the proposed method through experiments. The experimental results show that: 1) compared with the other three convolutional neural networks, our proposed PyDSC can significantly improve the recognition accuracy with a small increase in overhead; 2) Compared with the original PyConv, PyDSC can effectively reduce the complexity of the model.**

Keywords*—*high resolution range profile, pyramid convolution neural network, Depthwise separable convolution, one-dimensional convolution neural network

I. Introduction

Radar high resolution range profile (HRRP) is the amplitude of the coherent sum of the complex sub echoes of the target scattering points in each range unit, which represents the projection of the target scattering center echo on the radar line of sight[1]. Because HRRP contains important structural features such as target size and scattering center distribution, HRRP target recognition has attracted extensive attention in the field of radar automatic target recognition[2]. In recent years, many traditional machine learning algorithms have been widely used in HRRP recognition[3] [4], but these machine learning methods rely on manual feature extraction, and can not better capture the deep information of data for accurate recognition. Therefore, how to effectively learn the features in the data is an important problem to improve the recognition ability of HRRP.

Deep learning can automatically extract the features of data by using deep neural network. Due to its strong feature extraction ability, deep learning algorithm has been widely used and achieved good results in the fields of face recognition [5], image classification [6], computer vision [7]. Convolutional neural network (CNN) has strong feature extraction ability to mine the local features of data through convolution layer.

Convolutional neural network is widely used in the field of HRRP recognition. Yang et al. [8] proposed HRRP recognition method based on convolutional neural network and constructed two kinds of CNN networks. One CNN network directly extracts features from one-dimensional HRRP, and the other CNN network extracts features from two-dimensional data after HRRP conversion. Compared with template matching method, the target recognition rate is greatly improved, and the results show that one-dimensional HRRP data is more conducive to feature extraction. Guo et al. [9] proposed a HRRP target recognition method based on deep multi-scale one-dimensional convolution neural network, which uses multi-scale convolution kernel to extract features with different accuracy. Wan et al.[10] transformed one-dimensional HRRP data into two-dimensional spectrogram and identified it with 2d-dcnn, which achieved good results. Lu et al. [11] extracted the bi-spectrum spectrum feature of HRRP and used it as the input of convolutional neural network to extract deep features.

In order to solve the problem of insufficient recognition ability of CNN network for HRRP data, this paper proposes an HRRP target recognition method based on one-dimensional pyramid convolution neural network (PyDSC). According to the characteristics of HRRP data, pyramid convolution is selected, which can better extract local features of different scales and improve the overall recognition ability. At the same time, we also add DSC technology to the pyramid convolution network to reduce the excessive parameters introduced by using multi-scale convolution kernel.

The main contributions of this paper are as follows:

This research was funded by the National Natural Science Foundation of China, grants number 61876189, 61806219, and 61703426.

(1) According to the characteristics of HRRP data, we choose pyramid convolution technology and use a variety of convolution cores with different scales to extract features of different sizes, so as to better extract the features in the dataset.

(2) Using DSC technology, by decomposing a convolution operation into point convolution and depth convolution, the parameters of the model are reduced and the time of target recognition is reduced;

(3) A comparative experiment is carried out on the simulated dataset, and PyDSC is compared with other CNN models and other HRRP target recognition methods. In addition, the effectiveness of each part of PyDSC was demonstrated by ablation test. Finally, the timeliness of PyDSC model complexity is evaluated.

The rest of this paper is organized as follows. In Section II, we introduce the details of our VGM data preprocessing model and 1D PyDSC. The third section introduces the details of the comparative experiment and ablation experiment and analyzes the results of the experiment. In the section IV, the conclusion is drawn..

II. Methods

In this section, we describe in detail the data preprocessing method based on VGM and the 1D PyDSC. Therefore, an HRRP recognition based on PyDSC (PyDSC) is established for HRRP recognition

A. Pyramid Depth Separable Convolutional Neural Network (PyDSC)

In this section, we introduce pyramid convolution neural network (PyConv) and depthwise separable convolution neural network (DSC) respectively. We use PyDSC to extract the features of different dimensions in the dataset, and then use the characteristics of DSC to reduce the complexity of network structure and improve the timeliness of HRRP recognition.

1) Pyramid convolution neural network (PyConv)

Pyramid conv uses multi-scale convolution kernels to extract features of different sizes. For the huge difference in dimension, a single standard convolution kernel size is difficult to fully capture the information of all scales. We design convolution kernels with different scales. Fig. 1 shows the structure of PyConv.

For a standard one-dimensional CNN, we set the size dimension of its input sample $C_i * W_i$, where C_i is the number of channels and W_i is the width of the sample. Identical convolution kernels act on all input channels at the same time to obtain the feature map of the output channel. The number of output channels is equal to the number of convolution kernels. the width of the feature map is $w_o = \frac{(W_i - K + 2P)}{S} + 1$. The output feature maps size is $C_o * W_o$.

In pyramid convolution, a pyramid convolution layer is constructed by using convolution kernels with different kernel sizes. As shown in Fig. 2, the kernel size gradually increases and depth gradually decreases from top to bottom. For the convolution kernels in the bottom, they can only act on some channels of the input feature maps.

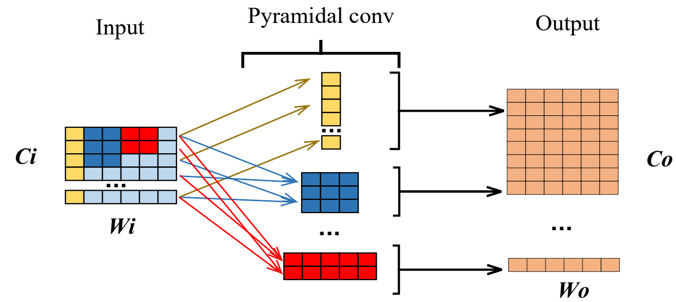


Fig. 1. Pyramid Convolution Neural Network

Given this situation, PyConv introduces group convolution technology[12]. As shown in Fig. 2, the input feature maps are divided into different groups, and the convolution kernels are used for processing in each group to ensure that the convolution kernel of each layer can completely act on all channels of the input.

Through the group convolution technology, the convolution kernels of different layers can generate feature maps with the same size and different number of channels, and then connect these feature maps according to channels to get a complete output feature map.

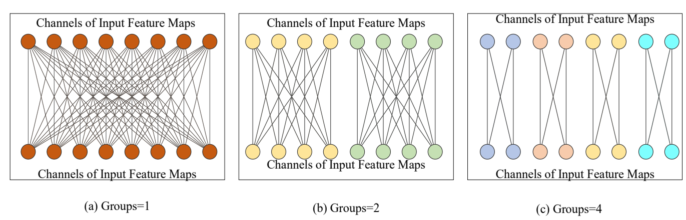


Fig. 2. Group convolution

2) Depthwise separable convolution

DSC decomposes a standard convolution operation into two steps: depthwise convolution and pointwise convolution. Based on the standard CNN, DSC processes the channel domain and spatial domain separately, which greatly reduces the number of parameters and computational complexity of the model.

For one-dimensional DSC, enter the sample dimension $C * W$, where C is the number of channels and W is the width of the sample. The convolution kernel size is k, the step size is s, and the padding is p.

For standard convolution, each convolution kernel will use the information on all input channels at the same time and output a single channel. Each convolution kernel corresponds to an output channel.

As shown in Fig. 3, in the depthwise separable convolution, we first split the convolution kernel into a single channel form for depthwise convolution, and convolute each channel without changing the depth of the input feature map. In this way, the number of output feature maps and input channels is the same. Then, after the intermediate feature map is obtained, the feature map is transformed into the required number of output channels C using pointwise convolution kernel， the dimension of the output feature map is $C_{out} * w_{out}$.

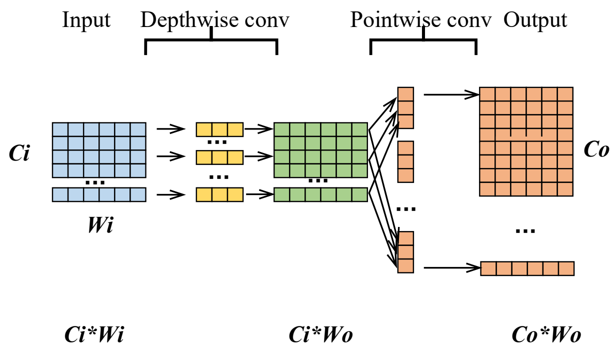


Fig. 3. Depthwise Separable Convolution

3) PyDSC

In section 1) and 2), we have described PyConv and DSC networks in detail. In this section, we apply DSC technology to the PyConv network, set the depth of each layer in PyConv to 1. And there is only one input channel in each group, to reduce the network complexity, reduce the number of parameters and improve the recognition speed.

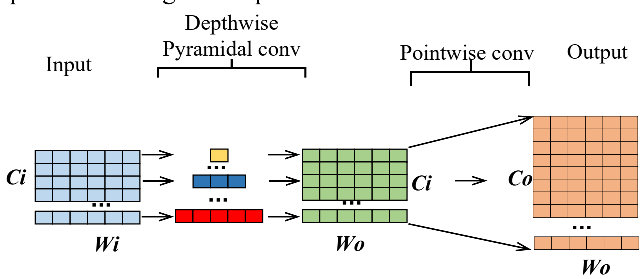


Fig. 4. Structure of PyDSC

The model structure of PyDSC is shown in Fig. 4 . Each channel of the PyDSC input feature map has exactly one convolution kernel corresponding to it, and the convolution kernels in the pyramid contain several different scales. It can be seen from 3.2 that after passing through the depthwise conv the size of the obtained feature map changes and the number of channels remains unchanged. Then, the feature map with the required dimension is output by pointwise Conv.

According to reference [13], we link Pyramid-depthwise (PyDW) Conv, pointwise (PW) conv, batch normalization, activator, and max-pooling, five different layers into a convolution block, which is shown in Fig. 5. In PyDSC, we use three such convolution block s, and the number of filters is 32, 64 and 64 respectively.

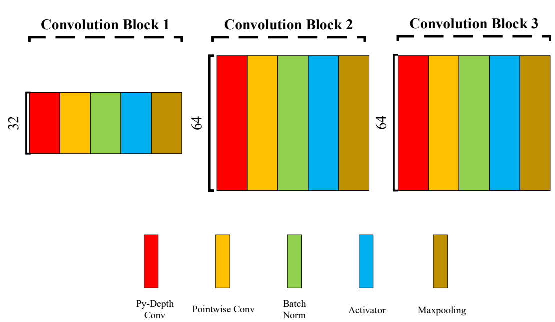


Fig. 5. Network structure of PyDSC

In the part of data preprocessing, we first extract a few kinds of information from the dataset, and decompose the continuous features of multimodal distribution. Taking the original sample $s=(r,y)$ as the input, where $\mathbf{r}_j = \{c_{1,j},\dots,c_{N_c,j},d_{1,j},\dots,d_{N_d,j}\}, j\in\{1,\dots,n\}$ is a row of the original dataset and y is the label. As Formula 3 in 3.1 VGM decomposition is used to deal with the continuous features $c_{i,j}$, and discrete features $d_{i,j}$ are encoded by One-Hot encoder. Finally, the processed data $s' = (r', y)$ is obtained

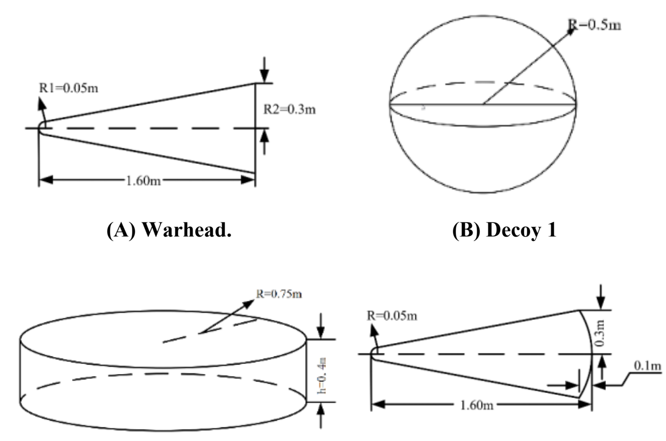
In the PyDSC module, we use PyDSC to perform HRRP recognition, as shown in Fig. 7. PyDSC model takes the processed data $s' = (r', y)$ as the input , and outputs the probability distribution $P(y)$ of different classes in the test dataset.

III. Experiment and Analysis

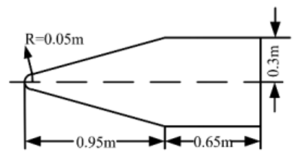
We conducted experiments to evaluate the performance of PyConv. We experimentally determine the hyper-parameters of VGM and PyDSC, and compare PyConv with other CNN methods and other methods. At the same time, we evaluate the availability of each module in PyConv. Finally, we compared the model parameters and recognition time of PyDSC with 1d-CNN and PyConv.

A. Dataset

We use an electromagnetic simulation software FEKO to simulate radar echoes from 5 mid-course ballistic targets, and the physic characteristics of these targets are shown in Fig. 5. Based on the high-frequency asymptotic theory, the physical optics method is adopted for simulation. The FEKO simulation parameters are set as follows: azimuth angle is 0-180°, angle step is 0.05°, the pitch angle is 0, the center frequency is 10 GHz with start frequency 9.5 GHz and end frequency 10.5 GHz and the number of frequency sampling points is 128. At last, the default optimal mesh size and horizontal polarization are adopted. Finally, there are 18005 samples attained through processing by inverse fast Fourier transform, each target contains 3601 HRRP samples of different degrees and each sample is a 256-dimensional vector. And we randomly split these samples into two datasets: training dataset with 14404 samples and testing dataset with 3601 samples.



(A) Warhead. **(B) Decoy 1** **(C)Decoy 2** **(D)Decoy 3**



(E) Decoy 4

Fig. 6. The physic characteristics of five mid - course ballistic targets. (A) Warhead. (B) Decoy 1. (C)Decoy 2. (D) Decoy 3. (E) Decoy 4

B. Metrics

We use accuracy, precision, recall, F1 score, and G-means as the main metrics to measure the classification performance of the PyDSC. Accuracy indicates the overall performance of PyDSC. Precision quantifies the specific classification capacity of each class, while recall indicates the detection rate for a specific class. These metrics are defined as:

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

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

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

where TP, TN, FP, and FN[14] are true positive, true negative, false positive, and false negative respectively.

F1 score is the harmonic mean value of precision and recall. G-means is a comprehensive parameter of positive and negative class accuracy.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$G - \text{mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}$$

C. Experimental Procedure

1) Experimental environment and parameter setting

The experiments were run on a personal computer with an AMD Ryzen 9 5900X@4.4GHz CPU, NVIDIA GeForce RTX 2080 Ti GPU, and 32 GB RAM using Python 3.7.

The feature vector is input into the PyDSC module. We set the size and layers of the convolution kernel of the PyConv layer according to the following rules: the size of the convolution kernel of each layer is an odd column that increases successively from top to bottom. At the same time, we set the odd sequence to start from 1, so that the structure of the PyConv layer only depends on its layers.

We set the number of channels of the three function blocks to 32, 64, and 64 respectively.

The specific structures of PyDW Conv and PW conv are given in the table below. Each PyDW Conv layer consists of 4 layers. The *kernel size* is [1, 3, 5, 7] from top to bottom, and the *stride* is uniformly set to 2. And *padding* is set to be half of the *kernel size*, it can ensure that the size of the feature map obtained of each layer is the same. The number of convolution kernel channels of each layer is set as shown in table I.

PW conv's *kernel size* and *stripe* are both 1, and *padding* is set to 0. The activation function is set to Mish[15].

TABLE I. PyDSC Parameter Setting

	#	Layer	Kernel Size	Filters	Output size
	1	Input		-	257*1
	2	PW Conv		64	257*64
Conv block1	3	PyDW conv	1,3,5,7,71	16,16,16,12,4	129*64
	3	PW conv	1	32	129*32
	4	BatchNorm2d	-	-	129*32
	5	Mish	-	-	129*32
	6	MaxPool2d	-	-	127*32
Conv block2	7	PyD conv	1,3,5,7,71	8,8,8,4,4	64*64
	8	PW conv	1	64	64*64
	9	BatchNorm2d	-	-	64*64
	10	Mish	-	-	64*64
	11	MaxPool2d	-	-	62*32
Conv block3	12	PyD conv	1,3,5,7,71	16,16,16,12,4	31*64
	13	PW conv	1	64	31*64
	14	BatchNorm2d	-	-	31*64
	15	Mish	-	-	31*64
	16	MaxPool2d	-	-	31*32

There is no automatic parameter tuning algorithm at this stage. We conduct a large amount of experiments based on existing references to compare and analyze the influence of different parameters. The final parameters of the model are determined through a large number of experiments. The training parameter settings are given in table II.

TABLE II. PyDSC parameter setting

Epoch	300
Batch Size	512
learning rate	0.01
Activation	Mish[15]

2) Experimental setup

To test the performance of the HRRP recognition method based on PyDSC, we designed the following experiments:

a) ***Experiment 1****: training experiment of PyDSC model*

To test the effects of different layers and convolution kerenel scales on the performance of PyDSC. To explore the influence of different depths of convolution kernels in PyDSC, we designed a comparative experiment.

b) ***Experiment 2****: Comparative experiment between PyDSC and other CNNs*

To verify the superiority of PyDSC, we compare it with three other method based on CNN, including standard 1D-CNN, Depthwise Over-Parameterized (DO) CNN[16], and PyConv without DSC. In the comparative experiment, we make the basic parameters of all methods consistent, as shown in table II.

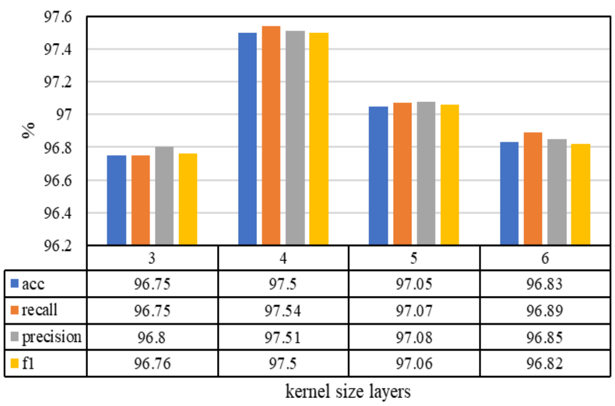
c) ***Experiment 3****: Timeliness analysis of the model*

Timely HRRP recognition is as important as accurate identification. Therefore, we also analyze the FLOPs, parameters, memory, and times of PyDSC and compare them with other CNN methods.

D. Experiment Results and Analysis

1) Experiment 1: training experiment of PyDSC model.

To explore the influence of the recognition ability of different pyramid convolution kernels layers in PyDSC, we designed a comparative experiment. We fix the size of convolution kernel and set the size of convolution kernel of each layer in an odd order. For example, the convolution kernels of three pyramid layers are [1,3,5], and the convolution kernels of four pyramid layers are set as [1,3,5,7].

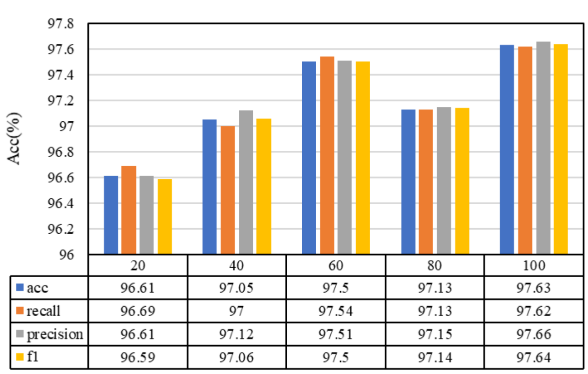


	3	4	5	6
acc	96.75	97.5	97.05	96.83
recall	96.75	97.54	97.07	96.89
precision	96.8	97.51	97.08	96.85
f1	96.76	97.5	97.06	96.82

Fig. 7. Overall Performance of different layers in PyDSC

As shown in Fig. 7, we can see that accuracy, recall, precision, and F1 score achieve the best performance when the layers of convolution kernel are set to 4. Therefore, we set the number of kernel layers as 4 to achieve the best classification effect.

Then we fix the number of convolution kernels as [1,3,5,7]. For the initial layer, the convolution kernel number {c4_20, c4_40, c4_60, c4_80, c4_100}. In Fig.8, we can see that with the increasing number of convolution kernels in the initial layer, the recognition accuracy exceeds 97% when the convolution kernel size is greater than 60. However, as the number of convolution kernels increases, the parameters of the model increase rapidly, as shown in Fig. 9. When the number of convolution kernels is increased from 60 to 100, the performance improvement of less than 1% requires 165% additional flops. Considering the classification performance and overhead, we choose to set the number of convolution cores of the initial layer to 60.



	20	40	60	80	100
acc	96.61	97.05	97.5	97.13	97.63
recall	96.69	97	97.54	97.13	97.62
precision	96.61	97.12	97.51	97.15	97.66
f1	96.59	97.06	97.5	97.14	97.64

Fig. 8. Overall Performance of different conv kernel size

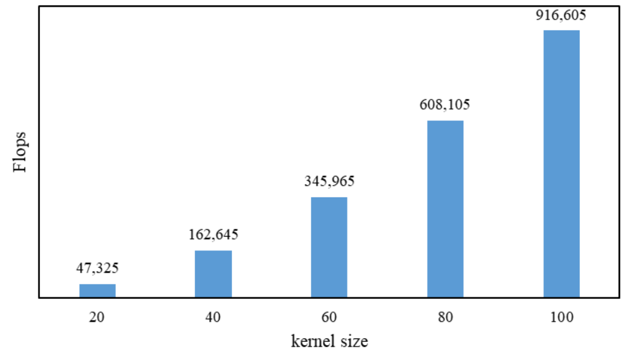
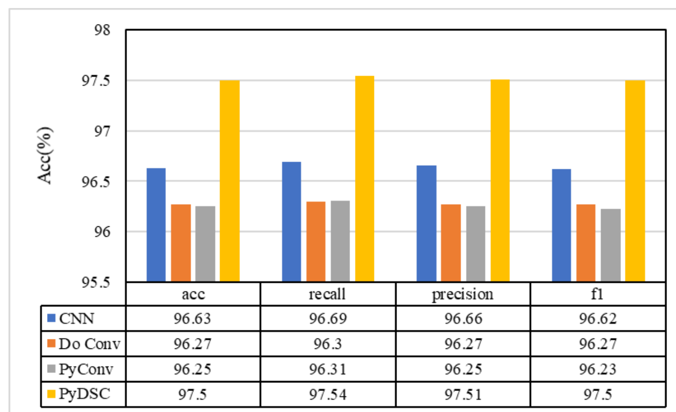


Fig. 9. Parameters of different conv kernel size

2) Experiment 2: Comparative Experiment between PyDSC and other CNNs

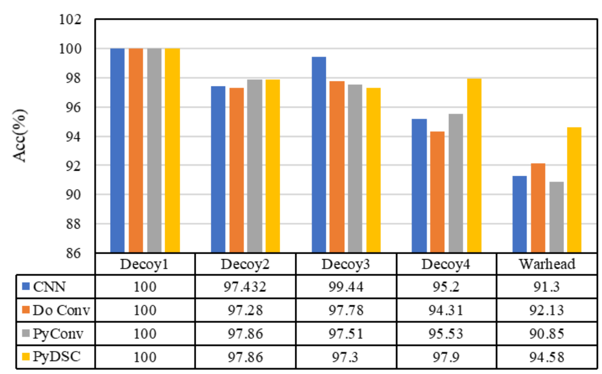
We trained on the HRRP dataset and tested on the testsets . The experimental results are shown in Fig. 10 and 11.



	acc	recall	precision	f1
CNN	96.63	96.69	96.66	96.62
Do Conv	96.27	96.3	96.27	96.27
PyConv	96.25	96.31	96.25	96.23
PyDSC	97.5	97.54	97.51	97.5

Fig. 10. Performance comparison of different CNN on HRRP

In Fig. 10, we can see that on the HRRP dataset, the PyDSC method is superior to the other methods in all metrics, which proves the effectiveness of our proposed method.

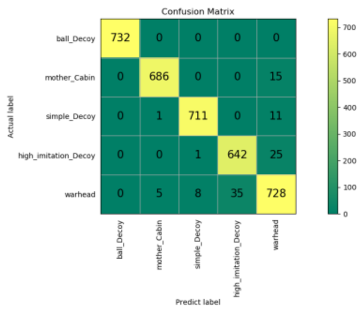


	Decoy1	Decoy2	Decoy3	Decoy4	Warhead
CNN	100	97.432	99.44	95.2	91.3
Do Conv	100	97.28	97.78	94.31	92.13
PyConv	100	97.86	97.51	95.53	90.85
PyDSC	100	97.86	97.3	97.9	94.58

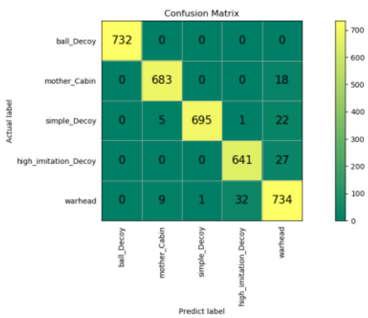
Fig. 11. Performance comparison of different CNN on each target

Fig. 11 show the accuracy of different CNN methods for five different types of targets respectively. Compared with other 3 methods, PyDSC has the highest accuracy for DECOY2, DECOY4 and Warhead targets, and the performance of

DECOY3 is only slightly backward in few metrics. The comparative experimental results show that PyDSC improves the accuracy for Warhead target.



(A) . Confusion matrix of 1DCNN



(B). Confusion matrix of PyDSC

Fig. 12. confusion matrix of two methods

As shown in Fig. 12, we can see that compared with CNN, the recognition effects of Warhead target in the confusion matrix of PyDSC are improved, but the classification of DECOY3 is reduced.

3) *Experiment 3: Timeliness analysis of the model*

We analyze the parameters and time in the HRRP recognition of the model. According to table III, PyDSC significantly reduces parameters, memory occupation and recognition time compared with PyConv.

TABLE III. COMPARISON OF MODEL COMPLEXITY OF DIFFERENT METHODS

Model	para	mem (MB)	Time(ms)
CNN	780,605	3.98	**73.9**
PyConv	940,565	5.17	95.4
PyDSC	**345,956**	**3.13**	100.1

However, compared with the original CNN, PyDSC, and PyConv are both not good enough in memory occupation and recognition time. DSC reduces the parameters of the network model by decomposing the original convolution layer into two different layers: depth convolution layer and point convolution layer, and this increases the number of layers of the PyDSC. As shown in Table III, compared with PyConv, the number of network layers of PyDSC is increased by 3. Therefore, although PyDSC reduces the computational complexity, it slightly increases the recognition time.

Combined with the result in section III, while significantly improving the recognition performance, PyConv also improves the complexity of the model. After adding DSC, PyDSC effectively reduces the parameters of the model and significantly improves the recognition performance of the model with a small increase in model complexity.

IV. CONCLUSION

In this paper, PyConv is selected for efficient extraction according to the multi-scale feature in the preprocessed data, and the DSC is used to reduce the complexity of the model. On this basis, we establish an HRRP target recognition based on PyDSC. Through experiments, we apply PyDSC to the HRRP dataset. The comparative study shows that our method is better than the other CNNs. Additional research on ablation test and model complexity shows that PyDSC can significantly improve the recognition performance when the model parameters and recognition time increase slightly compared with the standard CNN.

ACKNOWLEDGMENT

This research was funded by the National Natural Science Foundation of China, grants number 61876189, 61806219, and 61703426.

REFERENCES

[1] K. Liao, J. Si, F. Zhu, and X. He, “Radar HRRP Target Recognition Based on Concatenated Deep Neural Networks,” *IEEE Access*, vol. PP, pp. 1–1, 2018.

[2] J. Liu, N. Fang, B. F. Wang, and Y. J. Xie, “Scale-space theory-based multi-scale features for aircraft classification using HRRP,” *Electronics Letters*, vol. 52, no. 6, pp. 475–477, 2016.

[3] L. Lei, X. Wang, Y. Xing, and B. Kai, “Multi-polarized HRRP classification by SVM and DS evidence theory,” *Control & Decision*, vol. 28, no. 6, 2013.

[4] L. I. Rui, X. Wang, L. Lei, and A. Xue, “Ballistic Target HRRP Fusion Recognition Combining Multi-class Relevance Vector Machine and DS,” *Information and Control*, 2017.

[5] M. Ranzato, Y. L. Boureau, and Y. Lecun, “Sparse feature learning for deep belief networks,” 2007.

[6] Hinton, E. G., Salakhutdinov, and R. R., “Reducing the Dimensionality of Data with Neural Networks.,” *Science*, 2006.

[7] I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” *arXiv:1406.2661 [cs, stat]*, Jun. 2014, Accessed: Sep. 28, 2020. [Online]. Available: http://arxiv.org/abs/1406.2661

[8] Y. Yang, J. Sun, Y. U. Shengkang, and X. Peng, “High Resolution Range Profile Target Recognition Based on Convolutional Neural Network,” *Modern Radar*, 2017.

[9] C. Guo, T. Jian, X. U. Congan, H. E. You, S. Sun, and N. A. University, “Radar HRRP Target Recognition Based on Deep Multi-Scale 1D Convolutional Neural Network,” *Journal of Electronics & Information Technology*, 2019.

[10] J. Wan, C. Bo, B. Xu, H. Liu, and J. Lin, “Convolutional neural networks for radar HRRP target recognition and rejection,” *EURASIP Journal on Advances in Signal Processing*, vol. 2019, no. 1, 2019.

[11] W LU, Y S ZHANG, C XU. "HRRP target recognition method based on bispectrum-spectrogram feature and deep convolutional neural network", *Systems Engineering and Electronics*,vol. 42 no. 8, 2020

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, 2012, vol. 25. Accessed: Sep. 15, 2021. [Online]. Available: https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[13] Q. Xiang, X. Wang, Y. Song, L. Lei, R. Li, and J. Lai, “One-dimensional convolutional neural networks for high-resolution range profile recognition via adaptively feature recalibrating and automatically channel pruning,” *International Journal of Intelligent Systems*, vol. 36, Oct. 2020, doi: 10.1002/int.22302.

[14] S. Huang and K. Lei, “IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks,” *Ad Hoc Networks*, vol. 105, p. 102177, Aug. 2020, doi: 10.1016/j.adhoc.2020.102177.

[15] D. Misra, *Mish: A Self Regularized Non-Monotonic Neural Activation Function*. 2019.

[16] J. Cao *et al.*, “DO-Conv: Depthwise Over-parameterized Convolutional Layer,” *arXiv:2006.12030 [cs, eess]*, Jun. 2020, Accessed: Sep. 15, 2021. [Online]. Available: http://arxiv.org/abs/2006.12030