Central Attention Mechanism for Convolutional Neural Networks

Y. X. Geng, L. Wang, Z. Y. Wang and Y. G. Wang

Abstract—**Model performance has been significantly enhanced by channel attention. The average pooling procedure creates skewness, lowering the performance of the network architecture. In the channel attention approach, average pooling is used to collect feature information to provide representative values. By leveraging the central limit theorem, we hypothesize that the strip-shaped average pooling operation will generate a one-dimensional tensor by considering the spatial position information of the feature map. The resulting tensor, obtained through average pooling, serves as the representative value for the features, mitigating skewness during the process. By incorporating the concept of the central limit theorem into the channel attention operation process, this study introduces a novel attention mechanism known as the "Central Attention Mechanism (CAM)." Instead of directly using average pooling to generate channel representative values, the central attention approach employs star-stripe average pooling to normalize multiple feature representative values into a single representative value. In this way, strip-shaped average pooling can be utilized to collect data and generate a one-dimensional tensor, while star-stripe average pooling can provide feature representative values based on different spatial directions. To generate channel attention for the complementary input features, the activation of the feature representation value is performed for each channel. Our attention approach is flexible and can be seamlessly incorporated into various traditional network structures. Through rigorous testing, we demonstrate the effectiveness of our attention strategy, which can be applied to a wide range of computer vision applications and outperforms previous attention techniques.**

Index Terms—**Convolutional Neural Network, Attention Mechanism, Nonlinear Memory Structure, Feature Collection Skewness.**

I. Introduction

CONVOLUTIONAL neural network topologies often utilize attention strategies to enhance the utilization of relevant feature information while reducing the impact of irrelevant features. There are three types of attention: channel attention, spatial attention, and a combination of spatial and channel attention. Channel attention enhances the relative significance of channels by dynamically assigning weights, necessitating feature extraction. However, the use of channel attention also requires average pooling to gather feature information and generate representative values. Unfortunately, this pooling operation introduces skewness to the collected information, leading to biased feature representations and reducing the accuracy of network architecture training. Consequently, training becomes more challenging.

With classical attention methods like the SE attention method introduced in [10], the channel 2D tensor is compressed, making channel attention more susceptible to extreme values such as noise. This vulnerability arises due to the global average pooling operation involved in the squeezing process. However, in the CBAM attention strategy proposed by [11], although the channel attention component enhances the aggregation of essential feature information, similar to the SE attention approach, it does not adequately tackle the issue of skewness influence. The alteration of values serves to enrich the feature information. The CA attention approach, as described in [12], integrates location information, which plays a vital role in achieving spatial selectivity. However, during channel weight assignment, it utilizes the same parameter training for each channel. No significant operational modifications have been made to the impact. In [7], it is recommended to utilize Batch Normalization (BN) as a means to mitigate overfitting. The purpose of Batch Normalization (BN) is to regulate training parameters, ensuring that the outputs adhere to a normal distribution. However, BN does not modify the distribution of the feature's representative value, leading us to believe that it has no impact on the training environment.

In this research, we present a novel channel attention approach that extracts channel information across multiple batches and utilizes global average pooling on the extracted feature information to derive representative feature values. This method is grounded in the central limit theorem, where the eigenvalues are expected to follow a normal distribution. Specifically, our attention method utilizes strip average pooling to capture spatial information. To extract spatial information from multiple directions, we utilize two strip-shaped average pooling modules. However, considering the contextual connections among feature information, we integrate the extracted features to generate two new two-dimensional feature distributions. The two-channel features are formed by establishing connections between the new feature distributions. Subsequently, feature data is extracted using strip average pooling to generate a new one-dimensional tensor representing the values. We combine the two-dimensional channels to form a one-dimensional tensor, where each tensor value represents a channel, ensuring the enhancement of feature information. Afterwards, the feature map is subjected to an activation process where assigned weights are applied. As the central limit theorem is integrated into our attention approach to ensure that the retrieved features adhere to

Manuscript received March 21, 2024; revised August 15, 2024. This work was partly supported by the Chinese National Natural Science Foundation (Grant 71472081).

Y. X. Geng is a postgraduate student of the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: gen9yanx1n@ustl.edu.cn).

L. Wang is a Professor in the School of Computer Science and Software Engineering at the University of Science and Technology Liaoning, situated in Anshan 114051, China. (e-mail: wangli9966@ustl.edu.cn).

Z. Y. Wang is a programmer at the Automation Design Institute, Metallurgical Engineering Technology Co., Ltd., Dalian 116000, China (e-mail: wzyuan572351326@163.com).

Y. G. Wang is a postgraduate student of the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: gen9yanx1n@ustl.edu.cn)

a normal distribution, we term it 'central attention'. In Figure 1, we present a comparison with alternative attention techniques.

To simplify the network architecture during training and significantly increase accuracy, our primary goal is to ensure the retrieved feature information follows a normal distribution without adding extra parameters or computational load. At the same time, any convolutional neural network architecture can utilize our proposed attention approach because it is adaptable, lightweight, and plug-and-play. In the experimental section, we will demonstrate how our central attention strategy can significantly enhance the performance of a network architecture using a pretrained model. We utilize various dataset distributions in the experimental phase to showcase the advantages of our proposed attention technique over other similar methods, all of which yield optimal results. This demonstrates the attention method's capability for transfer learning. Furthermore, we tested it on several network architectures to illustrate that our attention method performs effectively even when the number of learnable parameters and computational requirements are equal. We aspire that our attention strategy can contribute to the advancement of convolutional neural networks in some capacity.

II. Related work

Convolutional neural network is a feedforward neural network. The LeNet architecture, first proposed by [1], [13], [20], established the fundamental framework of convolutional neural networks. Subsequently, this framework was further developed and expanded upon. For example, the AlexNet network, proposed in [2], [14], [21], [24], was developed based on LeNet, with subsequent deepening of the network architecture. This led to significant improvements in the model's feature extraction capabilities. The VGG network model, proposed in [3], [15], [16], [25], [27], increased the depth to 19 layers, making the model more complex and thereby enhancing its fitting power. In addition to increasing the network depth, we explore alternative methods to enhance the performance of feature extraction in convolutional neural networks.

Inception Net, proposed by [4], [16], [26], not only addresses the depth of the network model but also increases its width. [5] refined the model design established by [4], [16], [26] with the introduction of the Xception network. This model, as proposed by [5], emphasizes the relationships and spatial dependencies between channels. We also draw inspiration from the insights presented in [5], particularly the impact of nonlinear expression relationships between channels on model performance. However, continuously deepening or widening the network may yield unsatisfactory results. We have studied the ResNet network model, proposed by [6], [8], [1], which incorporates a mechanism to prevent degradation while deepening or widening the network.

Through our learning process, we observed that gradually deepening the network can lead to the phenomena of vanishing and exploding gradients. To address these issues, [7] introduced the Batch Normalization (BN) layer, which enhances gradient propagation by modifying the input data. At the same time, some researchers address this problem directly in their model designs. For example, [8], [18] proposed the DenseNet network, which disregards the traditional notions of depth and width. Instead, it maximizes feature reuse across layers, effectively mitigating the gradient vanishing phenomenon. Over time, researchers have increasingly prioritized lightweight models, with the most prominent example being the MobileNet series proposed by [9], [19], [23], which emphasizes the relationship between channels. We believe that designing a module specifically aimed at capturing the nonlinear expression relationships between channels and directly integrating it into the convolutional neural network model will enhance the model's performance.

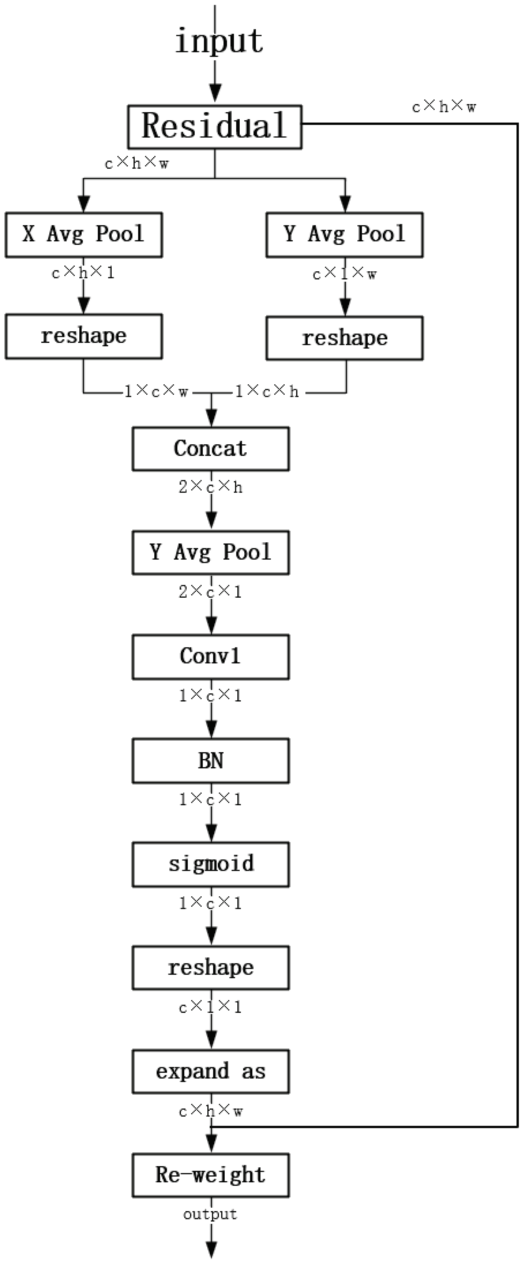

Fig. 1. Schematic diagram of the central attention mechanism

Attention mechanism. The attention mechanism is commonly described as a computational unit that prioritizes features based on their significance. In computer vision, attention mechanisms primarily involve soft attention, which can be further categorized into channel attention and spatial attention. Channel attention involves establishing relationships between features. For instance, the SE attention mechanism, introduced in [10], compresses global information from each channel and activates it through full connection operations, dynamically assigning channel weights.

However, some researchers argue that the SE computing unit may not be optimal for gathering global information. For instance, the CBAM attention mechanism was proposed by [11]. CBAM builds upon SE and introduces an operation to enhance maximum value aggregation, processed in a similar manner. A fully connected operation is crucial for its parameters. Nevertheless, the attention calculation units introduced in [10] and [11] share a limitation: they do not incorporate location information, significantly limiting the effectiveness of their attention mechanism. Therefore, [12] proposes a Coordinate Attention (CA) mechanism, integrating location information into features and activating them through a fully connected operation.

To harness the necessary information, all channel attention approaches utilize the acquired feature data by employing fully connected operations within the encoding-decoding structure. We found that this approach did not adequately consider how the retrieved feature information would impact the network architecture during training. If skewness is not corrected, the distribution of the extracted feature information will deviate from the normal distribution, leading to inaccurate feature representation values. To mitigate the influence of skewness and enhance the precision of the attention mechanism in judging feature importance, our central attention mechanism leverages the central limit theorem.

III. Attention mechanism

To enable the input feature $X = [x_1, x_2, \ldots, x_c] \in R^{c\times h\times w}$ to pass through the central attention and produce the output $Y = [y_1, y_2, \ldots, y_c]$, we designed an attention calculation unit. Our objective is to collect feature information through compatibility, activate it based on the inherent logic between the features and then apply a series of operations using the concept of the central limit theorem. To achieve this, we rearrange and store the collected feature data based on the logical relationships between their contexts. This preserves the feature logic and accelerates subsequent activation operations.

A. Parallel logical storage

We collect the feature information of the input feature tensor X along two mutually perpendicular directions. The collected feature information, $x_1 \in R^{c\times 1\times w}$ and $x_2 \in R^{c\times h\times 1}$, is stored in a tree. We implement parallel storage to manage the one-to-many relationship of the shape. At the same time, we readjust the feature distribution model of the feature information x_1 and x_2 respectively, modifying them to the distribution models $x_3 \in R^{1\times c\times w}$ and $x_4 \in R^{1\times h\times c}$, which are more suitable for the collection of feature information. The calculation formula is as follows:

$$x_3 = F_c^w(x_i, x_i + 1) \quad i \in (1, 2, ..., c-1) \tag{1}$$

$$x_4 = F_c^h(x_j, x_j + 1) \quad j \in (1, 2, ..., c-1) \tag{2}$$

$$x_3 = D(x_3) \tag{3}$$

Among them, F_c^w indicates connecting x_i along the h direction based on the channel, while similarly F_c^h indicates connecting x_j along the w direction based on the channel, where i and j represent the i-th channel and the j-th channel, respectively. The function D reverses the spatial dimensions of x_3, such that the vertical dimensions of $x_3 \in R^{1\times c\times h}$ are equal to those of x_4 after the inversion. For the convenience of operation, we connect x_3 and x_4 along the channel to obtain $x_5 \in R^{2\times c\times h}$, and its calculation formula is as follows:

$$x_5 = CAT(x_3, x_4) \tag{4}$$

where CAT represents the connection operation along the channel direction.

Discussion: To ensure the accuracy of the obtained feature information, we gather feature information along the vertical direction. According to the model used in its design, this article can work in addition to the vertical and horizontal directions.

B. Weight activation operation

We propose a weight activation operation by leveraging the feature information generated in the first step. Our proposed operation satisfies the following three characteristics: Firstly, it is convenient to operate. Secondly, it can capture nonlinear relationships between features. Finally, it requires fewer parameters and less computation. In order to meet the above characteristics, we performed the following operations. We conducted an interactive fusion operation between the features of x_5 to obtain $x_6 \in R^{2\times c\times 1}$, followed by a 1×1 convolution (Conv) operation to obtain $x_7 \in R^{1\times c\times 1}$. The following calculation formula:

$$x_7 = Conv(x_6) \tag{5}$$

We collect and process the feature information of x_7 based on the logical relationships between channels to obtain x_8, which is then normalized.

$$x_8 = BN(x_8) \tag{6}$$

To achieve multi-classification, we activate x_8 using the sigmoid function to obtain M with the following formula:

$$M = sigmod(x_8) \tag{7}$$

To implement the logical operation of the original feature, we use the reshape function to modify the logical arrangement of M, enabling logical matching with the original feature. We construct T $\in R^{c\times 1\times 1}$ by modifying M, then broadcast T according to the shape of X using the expand_as operation to obtain Y. The calculation formula is as follows:

$$T = reshape(M) \tag{8}$$

$$Y = T.expand\_as(x) \tag{9}$$

The reshape function and the expand_as function are utility functions commonly used in Python programming. They respectively involve reshaping a tensor to a new shape and expanding it without altering the data.

IV. Experimental part

This section compares our attention method with several other methods of the same class, following a description of our experimental setup. Lastly, we present the results of comparing our proposed central attention technique with alternative attention methods for object recognition and image classification.

A. *Take Resnet-34 as the baseline*

To verify the performance superiority of central attention, we conducted experiments using various attention methods and compared them based on the experimental results. During training, we used Resnet-34 as the baseline and employed the standard SGD optimizer for all models. The momentum, weight decay, and initial learning rate parameters in this optimizer were set to 0.9, 5×10^{-4}, and 0.01, respectively. The results are shown in Table 1, which demonstrate that our proposed central attention method outperforms other attention methods with similar features. This further confirms the superior performance of the central attention mechanism. We believe the advantage of the central attention mechanism lies in its non-linear storage structure, which departs from the traditional encoding-decoding framework.

TABLE I
With Resnet-34 as the baseline, Validating the performance superiority results of central attention.

Settings	Param.(M)	FLOPs(G)	Top-1(%)
Resnet-34	21.797	77.24	91.80
Resnet-34+SE	21.798	77.30	91.89
Resnet-34+CA	21.798	77.41	91.99
Resnet-34+CBAM	21.798	77.47	91.89
Resnet-34+CAM	21.797	77.38	92.38

B. *Take Resnet-50 as the baseline*

To verify that the central attention mechanism is not dependent on a specific network model, we conducted experiments using the ResNet-50 model as the baseline, which has more layers and parameters than the ResNet-34 model. We trained it using the same optimizer as for the ResNet-34 model. The results are shown in Table 2, demonstrating that our attention method is equally effective across different network models. In addition, we provide a rendering of adding CAM on ResNet-50 as shown in Figure 2 and 3.

TABLE II
Results comparing the performance of attention methods using Resnet-50 as a baseline.

Settings	Param.(M)	FLOPs(G)	Top-1(%)
Resnet-50	25.557	87.98	92.00
Resnet-50+CBAM	25.558	88.22	92.10
Resnet-50+SE	25.558	88.05	92.23
Resnet-50+CA	25.558	88.15	92.21
Resnet-50+CAM	25.557	88.13	92.42
Resnet-50+GAM	25.87	88.53	92.26
Resnet-50+SK	25.557	88.38	92.24

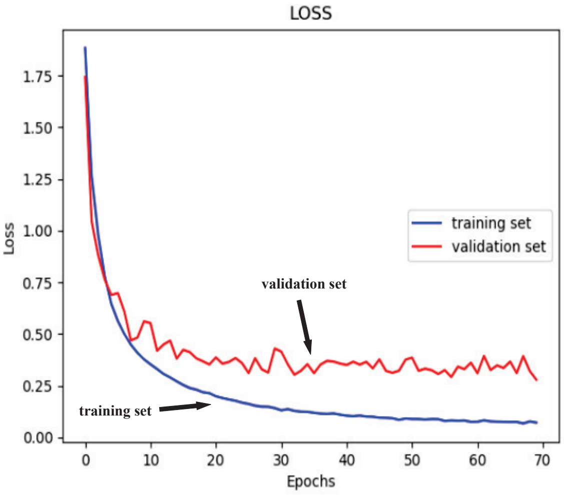

Fig. 2. Take Resnet-50 as the baseline, take the era as the abscissa, and use the loss rate as the trend graph of the ordinate respectively.

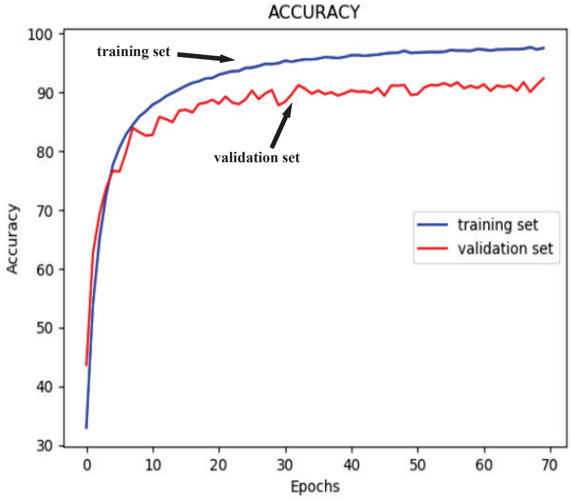

Fig. 3. Take Resnet-50 as the baseline, take the era as the abscissa, and use the accuracy rate as the trend graph of the ordinate respectively.

C. *Application*

1) Image classification: Our implementation is based on PyTorch and ResNet-34. We evaluate it on two datasets: CIFAR-10 and ImageNet. For training with CIFAR-10, we utilize the standard SGD optimizer with decay and momentum parameters, weight decay parameter, and initial learning rate set to 0.9, $5 * 10^{-4}$, and 0.01, respectively. We set the batch size to 128 and conducted training for 70 epochs. For training on the ImageNet dataset, we repeat the training for 100 epochs. In this experiment, we use the Top-1 accuracy metric.

CIFAR-10 Results. The experimental outcomes are presented in Table 3. From the results in the table, we observe that both the central attention mechanism and other attention methods achieve the highest accuracy in data attention after training on the CIFAR-10 dataset. We present Figure 4 and 5, illustrating the trends of error rate and accuracy rate for both training and testing results. Detection experiments on the CIFAR-10 dataset demonstrate that the classification model utilizing the central attention mechanism exhibits superior practical performance compared to other attention methods.

TABLE III
DETECTION RESULTS USING CIFAR-10 TEST WITH DIFFERENT ATTENTION METHODS.

Settings	Param.(M)	FLOPs(G)	Top-1(%)	Top-5(%)
Resnet-34	21.797	77.24	91.80	99.68
Resnet-34+SE	21.798	77.30	91.89	99.68
Resnet-34+CA	21.798	77.41	91.99	99.67
Resnet-34+CAM	21.797	77.38	92.38	99.69
Resnet-34+CBAM	21.798	77.47	91.89	99.62

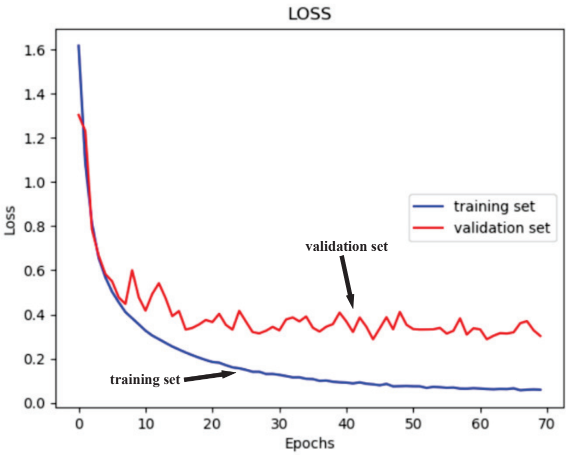

Fig. 4. Loss trends for training and testing results using the CIFAR-10 dataset.

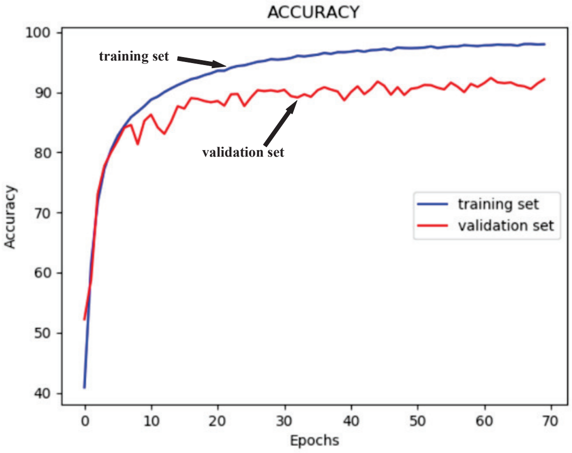

Fig. 5. Accuracy trends for training and testing results using the CIFAR-10 dataset.

TABLE IV
DETECTION RESULTS USING IMAGENET TEST WITH DIFFERENT ATTENTION METHODS

Settings	Param.(M)	FLOPs(G)	Top-1(%)	Top-5(%)
Resnet-34	21.797	77.24	63.01	100
Resnet-34+SE	21.798	77.30	63.26	100
Resnet-34+CA	21.798	77.41	63.96	100
Resnet-34+CAM	21.797	77.38	64.31	100
Resnet-34+CBAM	21.798	77.47	63.46	100

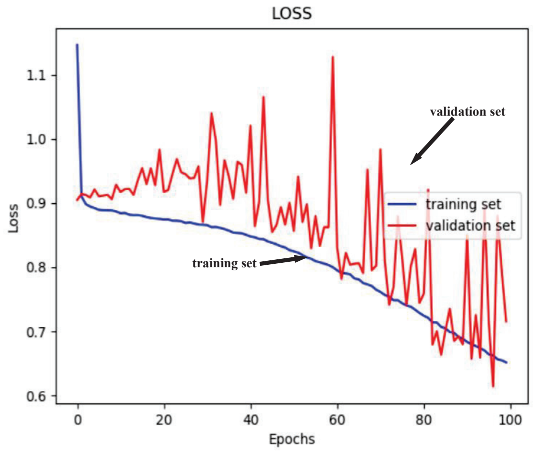

Fig. 6. Loss trends for training and testing results on the Imagenet dataset.

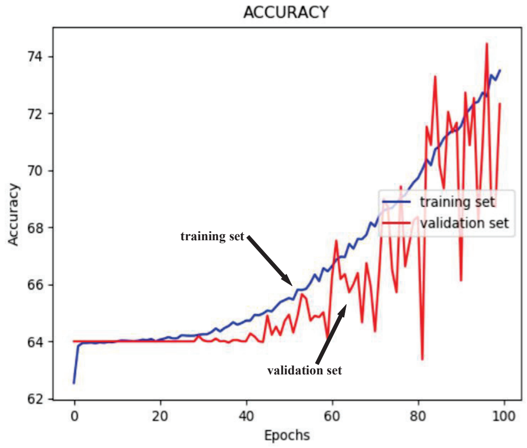

Fig. 7. Accuracy trends for training and testing results on the Imagenet dataset.

ImageNet Results. To demonstrate that the performance of the central attention mechanism is not constrained by dataset distribution, we utilize the ImageNet dataset. We adjust the number of classes to 100 and set the image size to 84 × 84. Finally, we present the training results for ImageNet in Table 4. According to the results, we found that the performance of the tree attention mechanism is comparable to other methods. It is noteworthy that the central attention mechanism demonstrates consistent performance across different conditions, indicating its ability to attend to all data without imposing specific conditions. We also include the trend graph, presented in Figure 6 and 7.

2) Object detection: Implementation details. Our code is implemented using PyTorch and SSD300. We train on two datasets, PascalVoc and MS COCO, with a batch size set to 8. We employ a standard SGD optimizer with an initial learning rate of 0.001, momentum of 0.9, and weight decay of 5×10^{-4}. The paper trains for 120,000 iterations, with the learning rate decayed by 0.1 of the initial learning rate after reaching 80,000 and 100,000 iterations. In this experiment, we use mAP as the metric for accurate detection.

PascalVoc Results. We train the model using the Pascal VOC 2007 dataset. The results are shown in Table 5. According to the results in the table, incorporating central attention into VGG16 leads to significant improvement in detection performance. Additionally, our attention mechanism achieves superior performance with fewer parameters compared to other methods such as SE and CA. Figure 8 presents the renderings of different attention methods, demonstrating that the convolutional model with dimensionality reduction attention exhibits superior transfer learning ability.

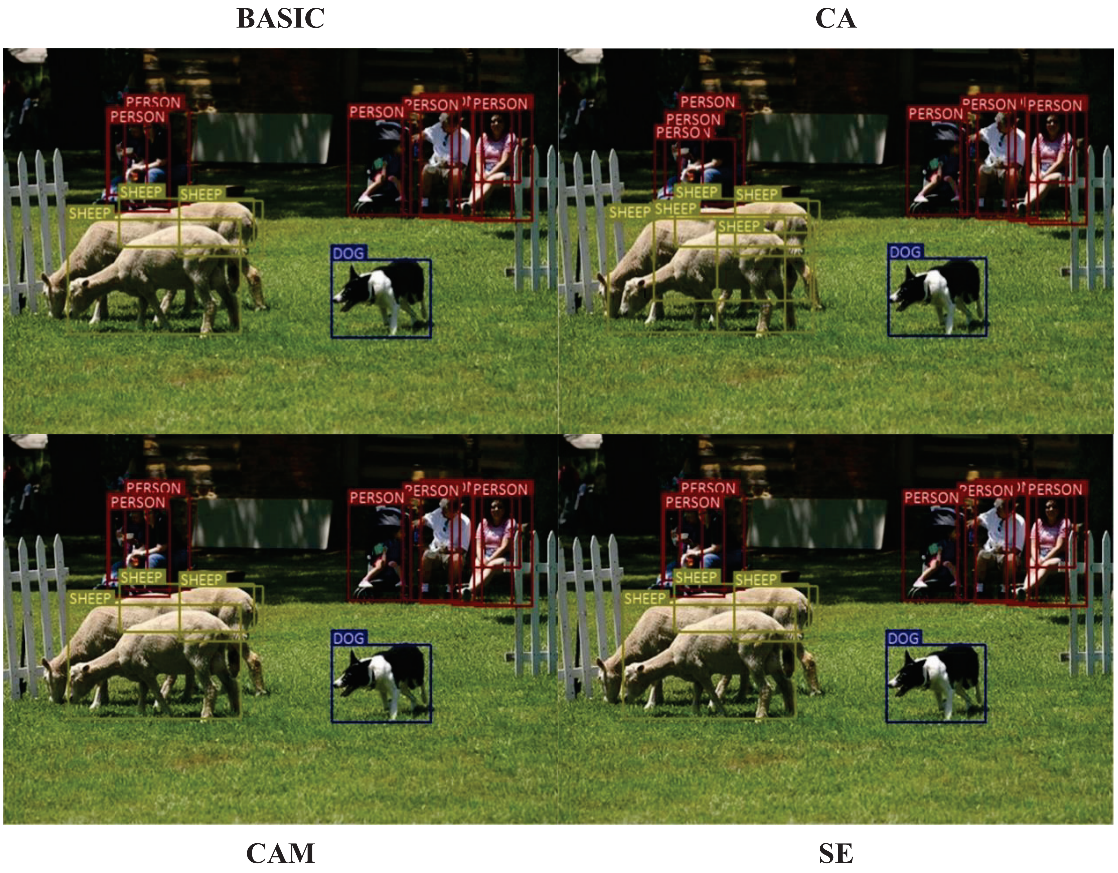

Fig. 8. The rendering of the PascalVoc dataset after training.

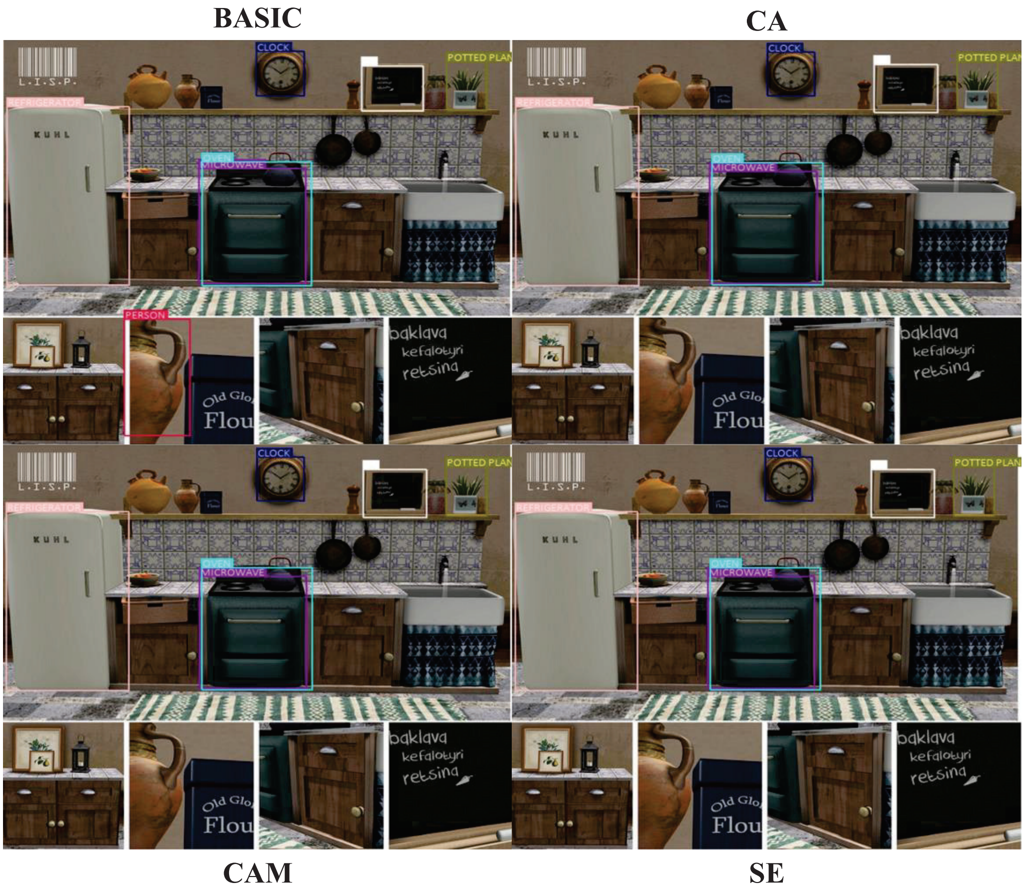

Fig. 9. Rendering after training with Ms coco dataset.

TABLE V
THE RESULTS OF MODEL TRAINING ON THE PASCAL VOC 2007 DATASET.

Settings	Param.(M)	FLOPs(G)	mAP(%)
SSD300	26.15	31.35	0.775
SSD300+CAM	26.15	31.35	0.780
SSD300+SE	26.18	31.35	0.776
SSD300+CA	26.20	31.35	0.778

Ms coco2017 Results. To demonstrate that the superior performance of the attention method in this model experiment is not dependent on the dataset, we use the more complex MS COCO dataset for verification. We present the results obtained using different attention methods in Table 6. The results indicate that central attention is superior to other attention methods in enhancing the feature extraction capability of neural networks. This also demonstrates that the performance impact of central attention is consistent across different dataset distributions. To better understand the experiment, we provide the experimental renderings in Figure 9.

TABLE VI
THE RESULTS OF MODEL TRAINING ON THE MS COCO 2017 DATASET.

Settings	Param.(M)	FLOPs(G)	mAP(%)
SSD300	26.15	31.35	0.368
SSD300+CAM	26.15	31.35	0.375
SSD300+SE	26.18	31.35	0.370
SSD300+CA	26.20	31.35	0.372

V. CONCLUSION

In this study, we develop a novel central attention computational unit for channel attention. This computational unit can mitigate skewness effects during feature extraction due to the central limit theorem, a concept from statistics. We demonstrate the effectiveness of our central attention in image classification and object detection. We anticipate that our attention strategy may provide some benefits when applied to convolutional neural networks.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[2] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Hcp: A flexible cnn framework for multi-label image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1901–1907, 2015.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[5] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258.

[6] M. S. Hanif and M. Bilal, "Competitive residual neural network for image classification," *ICT Express*, vol. 6, no. 1, pp. 28–37, 2020.

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*. pmlr, 2015, pp. 448–456.

[8] N. Martinel, G. L. Foresti, and C. Micheloni, "Wide-slice residual networks for food recognition," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 567–576.

[9] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713.

[10] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[11] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[12] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 713–13 722.

[13] Y. Bengio, Y. LeCun, C. Nohl, and C. Burges, "Lerec: A nn/hmm hybrid for on-line handwriting recognition," *Neural computation*, vol. 7, no. 6, pp. 1289–1303, 1995.

[14] R. Al-Jawfi, "Handwriting arabic character recognition lenet using neural network." *Int. Arab J. Inf. Technol.*, vol. 6, no. 3, pp. 304–309, 2009.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[17] A. Verma, H. Qassim, and D. Feinzimer, "Residual squeeze cnds deep learning cnn model for very large scale places image recognition," in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. IEEE, 2017, pp. 463–469.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] A. Krishnaswamy Rangarajan and H. Krishnan Ramachandran, "A fused lightweight cnn model for the diagnosis of covid-19 using ct scan images," *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, vol. 63, no. 1, pp. 171–184, 2022.

[22] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognitionn*, 2015, pp. 5353–5360.

[23] Z. Qin, Z. Zhang, X. Chen, C. Wang, and Y. Peng, "Fd-mobilenet: Improved mobilenet with a fast downsampling strategy," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1363–1367.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[25] G. Kalliatakis, S. Ehsan, M. Fasli, A. Leonardis, J. Gall, and K. D. McDonald-Maier, "Detection of human rights violations in images: Can convolutional neural networks help?" *arXiv preprint arXiv:1703.04103*, 2017.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognitionn*, 2015, pp. 1–9.

[27] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in Neuroscience*, vol. 13, p. 95, 2019.