Rega-Net: Retina Gabor Attention for Deep Convolutional Neural Networks

Chun Bao, Jie Cao, Yaqian Ning, Yang Cheng, and Qun Hao

Abstract—Extensive research works demonstrate that the attention mechanism in convolutional neural networks (CNNs) effectively improves accuracy. Nevertheless, few works design attention mechanisms using large receptive fields. In this work, we propose a novel attention method named Rega-Net to increase CNN accuracy by enlarging the receptive field. To the best of our knowledge, increasing the receptive field of the CNN requires increasing the size of the convolution kernel, which also increases the number of parameters. For solving this problem, we design convolutional kernels to resemble the nonuniformly distributed structure inspired by the mechanism of the human retina. Then, we sample variable-resolution values in the Gabor function distribution and fill these values in retina-like kernels. This distribution allows essential features to be more visible in the center position of the receptive field. We further design an attention module including these retina-like kernels. Experiments demonstrate that our Rega-Net achieves 79.96% Top-1 accuracy on ImageNet-1k for classification and 43.1% mAP on COCO2017 for object detection. The mAP of the Rega-Net increased by up to 3.5% compared to baseline networks.

Index Terms—Attention mechanism, Gabor, retina-like kernels.

I. INTRODUCTION

Deep learning networks are now being used in various fields related to computer vision, such as image recognition [1], object detection and recognition [3], [4], image dehazing [5], and 3-D vision [6], [7]. There is also a lot of research work in remote-sensing applications based on machine learning [8]. He et al. [9] proposed a semantic information-modulated (SIM) deep subpixel mapping (SPM) network (SIMNet). They used low-resolution semantic images before reinforcing the representation of spatial context features. He et al. [10] also developed an urban tree-specific SPM architecture to delineate the contextual characteristic of the urban tree patterns and generalize large-scale areas. Zhu et al. [11] proposed a knowledge-guided land pattern depicting (KGLPD) framework to bridge the “knowledge gap” in remote sensing.

Manuscript received 15 January 2023; revised 2 April 2023; accepted 19 April 2023. Date of publication 25 April 2023; date of current version 9 May 2023. This work was supported in part by the Funding of the Foundation Enhancement Program under Grant 2019-JCQJ-JJ-273 and in part by the National Natural Science Foundation of China under Grant 61871031 and Grant 61875012. (Corresponding author: Jie Cao.)

Chun Bao, Jie Cao, and Qun Hao are with the Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, Beijing Institute of Technology, Beijing 100081, China, and also with the Yangtze Delta Region Academy, Beijing Institute of Technology, Jiaxing 314003, China (e-mail: baochun@bit.edu.cn; caojie@bit.edu.cn; yangcheng2007@163.com; qhao@bit.edu.cn).

Yaqian Ning is with the Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, Beijing Institute of Technology, Beijing 100081, China (e-mail: daydayupnyq@163.com).

Digital Object Identifier 10.1109/LGRS.2023.3270186
over the distribution of the Gabor function. Finally, we fill the sampled values as the weights of convolutional kernels. We make this convolution into a feature attention module. This approach adds additional retina-like Gabor features extracted by central kernels without changing the original feature extraction of the CNN. The contributions of this letter are mainly in three aspects.

1) We propose a novel structure of circular convolutional kernels combined with the properties of the retina-like variable resolution. These retina-like convolutional kernels allow essential information to be more visible in the center position of the receptive field. At the same time, we make the retinal convolutional kernel parameters follow the Gabor function distribution. This architecture expands the receptive field of the CNN when extracting features.

2) We design the above retina-like convolutional kernel as a retina attention structure, which is capable of extracting deeper features as well as multiscale features. Through experimental validation on ImageNet-1k [18] as shown in Fig. 1 and MS COCO [28] datasets, we demonstrate that this structure achieves higher accuracy compared to the conventional attention module.

3) Our proposed retina Gabor attention method is a plug-and-play module that can be applied to various deep-learning tasks, such as image classification, object detection and recognition, semantic segmentation, and so on.

II. RELATED WORKS

*Gabor Filter:* The Gabor filter is similar to the human visual system in frequency and directional characteristics. There is also a large body of research work on the Gabor function in the field of computer vision [26], [29], [30]. The calculation process of the Gabor function operation of the image is shown in (1)

\[
g(x, y, \omega, \varphi, \sigma) = \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \exp\left(i \omega(x' + \varphi)\right) \tag{1}
\]

where \((x', y')\) is the spatial position of the pixel on the image. Here, we only use the real part of the function as shown in (4). \(\omega\) is the central angular frequency of a sinusoidal plane wave, \(\varphi\) is the anticlockwise rotation of the Gabor function (the orientation of the Gabor filter), and \(\sigma\) is the sharpness of the Gabor function along with both \(x\) and \(y\) directions. Regarding the calculation, we treat it similarly as [27]. Normally, we take \(\theta = \pi/\omega\). And \(\phi\) follows the distribution \(U(0, \pi)\)

\[
g(x, y, \omega, \varphi, \sigma) = \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \exp(\omega(x' + \varphi)) \tag{3}
\]

\[
\omega_n = \frac{\pi}{2} \sqrt{2^{(n-1)}} \quad \theta_n = \frac{\pi}{8}(m - 1) \tag{4}
\]

where \(n = 1, 2, \ldots, 5, m = 1, 2, \ldots, 8\).

![Fig. 2. Structure of the Rega kernel.](image)

![Fig. 3. Structure of the retina masks. In (a), we set all points other than activation points to 0. In (b), we set the values of the nonactivation points to 1. (c) Indicate which points we choose.](image)

III. PROPOSED METHOD

A. Rega Kernel

Inspired by the nonuniform sampling of the human eye, we adopt a retina-like design for the structure of convolutional kernels and propose the design idea of the Rega kernel. As shown in Fig. 2, we first generate a nonuniform numerical mask \(M\). This structure is capable of forming a circular-like mask. The size of the mask is 7 × 7, \(M \in \mathbb{R}^{7\times7}\). Depending on the circle’s radius, we fill convolutional kernels with 0 or 1. When the mask value is 1, the original convolutional kernel value is retained in that position. When the mask value is 0, the convolutional kernel’s effect at that position is reduced or removed. In Fig. 3, we refer to the value of the convolutional kernel that satisfies the condition \(r_1 < r \leq r_2\) as the one-gate activation point (OAP), which denotes the sampling point closest to the edge. The sampled points of the convolutional kernel, like the OAP, contribute very little value to the overall feature maps. The sampling points that satisfy the condition \(r \leq r_1\) are called two-gate activation points (TAPs). After one round of filtering, these activation points are sampled for more important information, where \(r\) is the distance from the coordinates of the center point of the surrounding points, \(r_2\) is half the size of the convolutional kernel, and \(r_1\) is the distance of the inner layer. The middle position of the convolutional kernel, which we call the fovea point (FP), indicates the point that contributes most to the feature sampling of the feature map. This design differs from the dilated convolution [31], [32]. The contributions of dilated convolutional kernels are homogeneous. And this uniform sampling increases the size of the receptive field. Our proposed Rega kernel retains the advantage of the increased receptive field, like dilated convolution, while aggregating the information in convolutional kernels. The values in retina-like masks are calculated as shown in the following equation:

\[
M_{i,j} = \begin{cases} 
1, & r_1 < r \leq r_2 \\
0, & \text{otherwise} 
\end{cases} 
\tag{5}
\]

where \(M_{i,j}\) is the values of the retina mask at the position \((i, j)\). We have two considerations in the design of the retina-like mask. As shown in Fig. 3(a), we set all points other than activation points to 0. This motivation is to remove
the influence of nonactivation points. Thus, the interference of weakly correlated features is removed during training. In Fig. 3(b), we set the values of the nonactivation points to 1. The values of the original positions are preserved in this way. However, this structure in Fig. 3(b) also brings some disadvantages. For instance, the number of FLOPs to calculate the parameters during the training process will increase, and the gradient calculation will be more complicated. Taking into account the above factors, we design the convolutional kernel masks in this work’s manner of Fig. 3(a).

Then, we consider filling the mask with trainable parameters. The structural parameters of retina masks (like 0, 1) are not involved in the gradient calculation in this work. We have two ways to fill values in retina masks. One way is to initialize the trainable parameters with random initialization. The other way is to let parameters follow a specific distribution. As shown in Fig. 2, we follow the sampling rule of the Retia mask to pick up the points for filling in the convolutional kernels, which combines Gabor functions with retina masks. We refer to [26] for the generation of Gabor convolutional kernels. Suppose the feature map of the input Gabor convolutional layer is \( F_{in} \), and the size of the input feature map channel is \( C_{in} \). The output feature map after convolution calculation is \( F_{out} \), and the size of the output feature map channel is \( C_{out} \). The generated Gabor convolutional kernel is shown in the following equation:

\[
K = g(x, y, \omega, \varphi, \sigma)
\]

where \( K \in \mathbb{R}^{C_{in} \times C_{out} \times 7 \times 7} \). The parameters of the Gabor kernel are all learnable parameters that can be used in the gradient calculation when we train the model. After the Gabor kernel dot with the retina mask, we obtain the final retina Gabor convolutional kernel, as shown in the following equation:

\[
\hat{K} = K \otimes M'
\]

where \( M' \) is initialized by copying from \( M \) according to channel, \( M' \in \mathbb{R}^{C_{in} \times C_{out} \times 7 \times 7} \). \( \hat{K} \) is the values of kernel after retina-like sampling. \( \otimes \) denotes element-wise multiplication.

### B. Rega Attention Network

The structure of our designed Rega network is shown in Fig. 4(a). Here, we take ResNet as the base model to illustrate the structure. We enhance the output feature maps \( F_{C1} \) and \( F_{C2} \) in C1 and C2 layers of ResNet by Rega attention module, respectively. The structure of Rega attention is shown in Fig. 4(b). We call the combination of “RG Conv” and “BN + ReLU” retina Gabor (RG) blocks. Moreover, the number of RG blocks is selective in the Rega attention module. The calculation of Rega attention is shown in (8)

\[
R_{a}(F_{C_i}) = \sigma(\text{AvgPool}(\text{RegaConv}(F_{C_i}, \hat{K})))
\]

\[
\hat{F}_{C_i} = \text{RegaConv}(F_{C_i}, \hat{K}_i) = \sum_{n=1}^{N} F^{(n)}_{C_i} \otimes K^{(n)}_i
\]

where \( \hat{F}_{C_i} \) is the feature map obtained after the \( n \)th convolutional kernel operation, \( C_i \) is the number of channels of the input feature map, \( F_{C_i} \) is the input feature map of the \( C_i \) layer, and \( F_{C_i} \in \mathbb{R}^{C_i \times H_i \times W_i} \). \( H_i \) is the height of the feature map, and \( W_i \) is the width of the feature map. \( \sigma \) denotes the sigmoid function. AvgPool is the average pooling layer. We use AvgPool to change the size of output feature maps. \( \hat{F}_{C_i} \) is the \( C_i \) layer for retina convolution operation. \( R_{a}(F_{C_i}) \) denotes the attentional feature matrix obtained after the Rega attention operation. The final output of the attention feature maps is calculated as shown in the following equation:

\[
R_{out}(F_{C_i}) = F_{C_i} \otimes R_{a}(F_{C_i}) = F_{C_i} \otimes \sigma(\text{AvgPool}(\hat{F}_{C_i}))
\]

where \( R_{out}(F_{C_i}) \) is the attention feature maps matrix of the layer \( C_i \). In the structure of Fig. 4(a), we adopt a skipped layer of residual connections. We input the feature maps of C1 and C2 layers into the Rega attention module and obtain the attention feature maps of layers C1 and C2, respectively. Then we concatenate \( R_{C1}(F_{C1}) \) and \( R_{C2}(F_{C2}) \) with ResNet block’s final C4 layer output feature maps. Finally, the 1 \times 1 convolution operation (\( \text{Conv}_{1 \times 1} \)) is used to integrate the final output channels to the same size as C4. The operation is shown in (11), where \( F_{output} \) is the final output feature map, \( F_{output} \in \mathbb{R}^{C_{out} \times H_w \times W_w} \).

\[
F_{output} = \text{Conv}_{1 \times 1}(\text{concat}[R_{C1}(F_{C1}), R_{C2}(F_{C2}), C4]).
\]

### IV. EXPERIMENTS

#### A. Implementation Details

In the experiments, we evaluate Rega-Net on the standard benchmarks: ImageNet-1k for classification and MS COCO 2017 for object detection and recognition. To ensure the fairness of the experiments, we chose the PyTorch framework to evaluate all experiments.

1) **Dataset**: Our image classification experiments are all performed on the ImageNet-1k dataset, which contains 1.28 M training images and 50 k validation images from 1000 classes. We conduct all object detection and recognition experiments on the challenging MS COCO 2017 dataset that includes 80 object classes. Following the common practice, we use all 115k images in the trainval35k split for training and all 5k images in the minival split as validation for the analysis study.

2) **Experimental Setup**: We implement our networks with Python 3.8 and PyTorch 1.8.0. The Rega-Net and benchmark models’ training is conducted on four Geforce RTX 3080Ti GPUs. For the classification task, we set the learning rate initially as 0.01 and decreased it by a factor of 10 after every 30 epochs for 100 epochs in total. The optimization is performed using the stochastic gradient descent (SGD) with
Fig. 5. Structure of the single-structure Rega attention module. We do not use a multiscale feature fusion structure for the ablation study.

### TABLE I

RESULTS OF ABLATION EXPERIMENTS PERFORMED ON THE RESNET-50 NETWORK

| Layer 1 | Layer 2 | Layer 3 | Layer 4 | Top-1 Acc (%) | Top-3 Acc (%) |
|---------|---------|---------|---------|--------------|--------------|
| ✓       | ✓       | ✓       | ✓       | 78.012       | 93.542       |
| ✓       | ✓       | ✓       | ✓       | 78.371       | 93.721       |
| ✓       | ✓       | ✓       | ✓       | 78.623       | 93.821       |
| ✓       | ✓       | ✓       | ✓       | 78.952       | 94.123       |

a weight decay of 1e-4, the momentum is 0.9, and the batch size is 16 per GPU. We train networks on the training set and report the Top-1 and Top-5 accuracies on the validation set with a single 224 × 224 central crop. The learning rate for object detection and recognition tasks is 1e-4. And we choose the MultiStepLR scheduler for the learning rate. The AdamW is used with a weight decay of 1e-3, the momentum is 0.9, and the batch size is 2 per GPU within 12 epochs. We follow the standard set of evaluating object detection via the standard mean average precision (AP) scores at different box IoUs or object scales, respectively.

### B. Ablation Study

For the ablation study, the structure we used is shown in Fig. 5. To reduce the network’s complexity, we do not use a multiscale feature fusion structure. Therefore, our model reaches convergence in a relatively short time.

To verify the effectiveness of our designed Rege attention module, we first trained it in four residual blocks of ResNet-50, Layers 1–4. And we place the Rega attention block after each block and tested it on the ImageNet-1k val dataset, and the results are shown in Table I. In Table I, we summarize that when we add the Rega attention block to Layer 4, the test accuracy is highest on the ImageNet-1k val dataset. And the Top-1 accuracy can be increased by at most 2.468% compared to the original ResNet-50 without Rega attention block on the Image-1k dataset. Therefore, in the following experiments, we prefer to add the Rega attention block to the last layer of feature map extraction for feature enhancement.

### C. Classification on ImageNet-1k

We conduct classification experiments on the ImageNet-1k dataset. The baseline we choose is ResNet-50 and ResNet-101. We compare our Rega-Net with some SOTA attention modules. And we choose the evaluation metrics with GFLOPs, Parameters, and accuracy (Top-1 and Top-5 accuracy). As shown in Table II, Rega-Net almost has the same parameters but achieves 1.128% gains in Top-1 accuracy and 0.332% improvement in Top-5 accuracy (on ResNet-50) with SA-Net. When using the ResNet-101 backbone, compared with SOTA attention modules, Rega-Net has 1.02% accuracy (Top-1) improvement with SENet [19]. Compared with SA-Net [22], Rega-Net has 1.003% gains on Top-1 accuracy and 1.06% gains in terms of Top-5 when we choose ResNet-101 as the backbone. Fig. 6 illustrates the visualization results. In Fig. 6, we can clearly see that the Grad-CAM masks of the Rega-integrated network cover the object regions better than other methods. That is to say, the Rega-integrated network learns well to exploit information in object regions and aggregate features from them.

### D. Object Detection and Recognition

We conduct object detection and recognition experiments on the COCO 2017 benchmark. For the experiment, we reproduce fully convolutional one-stage (FCOS) [34], Faster R-CNN [35], YOLOv7 [36], and RetinaNet [37] in our PyTorch framework to estimate the performance improvement of Rega-Net. The experimental results are summarized in Table III. We can see that Rega-Net improves the accuracy compared with SENet and SA-Net. We use mean AP (mAP) over different IoU thresholds from 0.5 to 0.95 for evaluation. We choose ResNet-50 and CSPDarknet-53 as the backbone.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
training. Next, we added SENet, SA-Net, and Rega-Net to the backbone and trained to obtain the accuracy of each of the four detectors.

V. CONCLUSION

We propose a novel method for designing convolutional kernels based on the retina-like principle in this letter. And we design a state-of-the-art attention module named Rega-Net. Experimental results show that the proposed method increases Top-1 accuracy by up to 2.468% on image classification compared to the original network. The MAP is increased by up to 3.5% on object detection to the original network. The accuracy of CNNs is effectively improved when compared with SOTA attention networks. The proposed network is expected to be applied in the fields of military reconnaissance, remote sensing, and public safety. The convolution operation we propose can also be used as the basic convolution module of deep learning. Moreover, the attention module we propose is also a plug-and-play module, which can play an important role in networks that need to improve accuracy. However, Rega-Net still needs further optimization in terms of speed and computational complexity. In future work, we will do more optimization work in terms of parameter compression and acceleration of Rega-Net. Through these optimization strategies, the proposed Rega-Net is more suitable for embedded platforms.

REFERENCES

[1] X. Zeng, W. Wu, G. Tian, F. Li, and Y. Liu, “Deep superpixel convolutional network for image recognition,” IEEE Signal Process. Lett., vol. 25, pp. 922–926, 2021.
[2] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 10012–10022.
[3] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” 2022, arXiv:2201.03545.
[4] F. C. Akyon, S. O. Altinuc, and A. Temizel, “Slicing aided hyper-inference and fine-tuning for small object detection,” 2022, arXiv:2202.06934.
[5] X. Liu, Y. Ma, Z. Shi, and J. Chen, “GridDehazeNet: Attention-based multi-scale network for image dehazing,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Los Alamitos, CA, USA, Oct. 2019, pp. 7314–7323.
[6] J. Li et al., “Efficient center voting for object detection and 6D pose estimation in 3D point cloud,” IEEE Trans. Image Process., vol. 30, pp. 5072–5084, 2021.
[7] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, “Deep learning for 3D point clouds: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 12, pp. 4338–4364, Dec. 2020.
[8] D. He, Q. Shi, X. Liu, Y. Zhong, G. Xia, and L. Zhang, “Generating annual high resolution land cover products for 28 metabolises in China based on a deep super-resolution mapping network using Landsat imagery,” Gisci. Remote Sens., vol. 59, no. 1, pp. 2036–2067, Dec. 2022.
[9] D. He, Q. Shi, X. Liu, and Y. Zhong, “Deep subpixel mapping based on semantic information modulated network for urban land use mapping,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 12, pp. 10628–10646, Dec. 2021.
[10] D. He, Q. Shi, X. Liu, Y. Zhong, and L. Zhang, “Generating 2m fine-scale urban tree cover product over 34 metabolises in China based on deep context-aware sub-pixel mapping network,” Int. J. Appl. Earth Observ. Geoinf., vol. 106, Feb. 2022, Art. no. 102667.
[11] Q. Zhu et al., “Knowledge-guided land pattern depiction for urban land use mapping: A case study of Chinese cities,” Remote Sens. Environ., vol. 272, Apr. 2022, Art. no. 112916.
[12] Q. Zhu et al., “A spectral–spatial-dependent global learning framework for insufficient and imbalanced hyperspectral image classification,” IEEE Trans. Cybern., vol. 52, no. 11, pp. 11709–11723, Nov. 2021.

[13] W. Wang et al., “Exploring sequence feature alignment for domain- adaptive detection transformers,” in Proc. 29th ACM Int. Conf. Multimedia, Oct. 2021, pp. 1730–1738.
[14] X. Liang et al., “R-drop: Regularized dropout for neural networks,” in Proc. Adv. Neural Inf. Process. Syst., Montreal, QC, Canada, 2021, pp. 1–12.
[15] A. de Santana Correia and E. L. Colombini, “Attention, please! A survey of neural attention models in deep learning,” 2021, arXiv:2103.16775.
[16] Z. Pan, B. Zhuang, H. He, J. Liu, and J. Cai, “Less is more: Pay less attention in vision transformers,” 2021, arXiv:2105.14217.
[17] Y. Li, T. Yao, Y. Pan, and T. Mei, “Contextual transformer networks for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 2, pp. 1489–1500, Feb. 2023.
[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Miami, FL, USA, Jun. 2009, pp. 248–255.
[19] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.
[20] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, 2018, pp. 3–19.
[21] X. Li, X. Hu, and J. Yang, “Spatial group-wise enhance: Improving semantic feature learning in convolutional networks,” 2019, arXiv:1905.09646.
[22] Q.-L. Zhang and Y.-B. Yang, “SA-Net: Shuffle attention for deep convolutional neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Toronto, ON, Canada, Jun. 2021, pp. 2235–2239.
[23] H.-Y. Yang, S.-R. Qi, C. Wang, S.-B. Yang, and X.-Y. Wang, “Image analysis by log-polar exponent-Fourier moments,” Pattern Recognit., vol. 101, May 2020, Art. no. 107177.
[24] V. Javier Traver and A. Bernardino, “A review of log-polar imaging for visual perception in robotics,” Robot. Auto. Syst., vol. 58, no. 4, pp. 378–398, Apr. 2011.
[25] D. Li, R. Du, A. Babu, C. D. Brumar, and A. Varshney, “A log-rectilinear transformation for foveated 360-degree video streaming,” IEEE Trans. Vis. Comput. Graph., vol. 27, no. 5, pp. 2638–2647, May 2021.
[26] C. Liu and H. Wechsler, “Independent component analysis of Gabor features for face recognition,” IEEE Trans. Neural Netw., vol. 14, no. 4, pp. 919–928, Jul. 2003.
[27] F. Yu, Y. Koltun, and T. Funkhouser, “Dilated residual networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 472–480.
[28] R. Chakraborty, X. Zhen, N. Voigt, B. Bendlin, and V. Singh, “Dilated convolutional neural networks for sequential manifold-valued data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Los Alamitos, CA, USA, Oct. 2019, pp. 10621–10631.
[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[30] F. Yu, V. Koltun, and T. Funkhouser, “Dilated residual networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 472–480.
[31] R. Chakraborty, X. Zhen, N. Voigt, B. Bendlin, and V. Singh, “Dilated convolutional neural networks for sequential manifold-valued data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Los Alamitos, CA, USA, Oct. 2019, pp. 10621–10631.
[32] C. Liu and H. Wechsler, “Independent component analysis of Gabor features for face recognition,” IEEE Trans. Neural Netw., vol. 14, no. 4, pp. 919–928, Jul. 2003.
[33] F. Yu, Y. Koltun, and T. Funkhouser, “Dilated residual networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 472–480.
[34] R. Chakraborty, X. Zhen, N. Voigt, B. Bendlin, and V. Singh, “Dilated convolutional neural networks for sequential manifold-valued data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Los Alamitos, CA, USA, Oct. 2019, pp. 10621–10631.
[35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[36] F. Yu, V. Koltun, and T. Funkhouser, “Dilated residual networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 472–480.