CS-GRESNET: A SIMPLE AND HIGHLY EFFICIENT NETWORK FOR FACIAL EXPRESSION RECOGNITION

Shaoping Jiang[1] *Xiangmin Xu*[1*] *Fang Liu*[2] *Xiaofen Xing*[1] *Lin Wang*[1]

[1]South China University of Technology, China [2] Guangdong University of Finance, China

ABSTRACT

Facial expression recognition (FER) has recently attracted attention in computer vision. However, existing methods mostly focus on the explicit performance and overlook their computational resources and memory consumption. Hence, achieving promising performance while maintaining the efficiency of models is still a huge challenge. In this work, we propose a highly efficient Channel-Shift Gabor-ResNet (CS-GResNet) to capture the crucial visual properties in facial images. Concretely, we incorporate the Gabor Convolution (GConv) into ResNet to produce the significant GResNet as our backbone with limited memory cost. Furthermore, we adopt an extremely simple yet effective Channel-Shift Module inserted into the GResNet to obtain the facial informative representation via facilitating information exchanged among neighboring channels. We conduct extensive experiments on three wild datasets: RAF-DB, FER2013 and SFEW. The results show that our proposed CS-GResNet achieves superior performance against the state-of-the-art methods with less computational and memory cost. Codes are available at *https://github.com/jsesr/CS-GResNet-PyTorch*.

Index Terms— Facial Expression Recognition, Highly Efficient, Gabor Convolution, Channel Shift

1. INTRODUCTION

Facial expression recognition (FER) plays a significant role in human daily life. It has recently become a topic of great interest among researchers because of its applications in various fields, including but not limited to sociable robots, health care and social psychology.

Many researchers have focused on the FER and they carefully design the structure of models to achieve promising performance [1, 2, 3, 4, 5, 6]. For example, Zhao et al. [1] proposed a global multi-scale and local attention network (MA-Net), which can address the issues of occlusion and pose variation. Li et al. [4] devised a training framework named the emotional education mechanism (EEM) to transfer knowledge to obtain the facial features. However, these works are inefficient because they pursue the greatest accuracy with excessive computational budgets and memory usage.

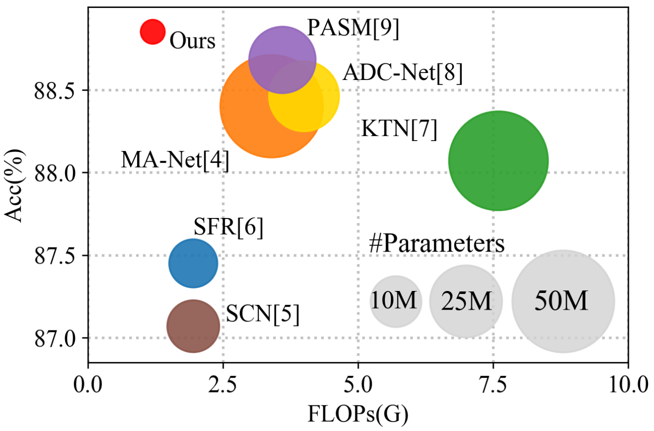

Fig. 1. Facial expression recognition performance comparison on RAF-DB dataset [7] in terms of accuracy, computational cost and model size. Our proposed CS-GResNet achieves the best trade-off between accuracy and efficiency compared with the latest state-of-the-art methods.

To improve the efficiency of the FER model, some researchers have attempted to produce some lightweight algorithms [8, 9]. For example, Barros et al. [8] presented a simple neural network (FaceChannel) that employed a VGG16 model based topology. Zhao et al. [9] proposed an expression detection model that can solve the delay problem. However, they have limitation in providing competitive performance because of their unsatisfied capacity in feature learning compared with the latest approaches.

To achieve superior performance while reducing computational cost and memory resources, we propose the GResNet via utilizing the Gabor Convolution (GConv) [10] to replace the raw convolution in ResNet. GConv is modulated filters which are incorporated the Gabor filters into the convolution filter. It not only enhances the robustness of facial features against the scale changes and rotations, but also significantly reduces the number of parameters due to the Gabor filters with pre-defined scales and orientations.

To further obtain the informative representation for FER and maintain the efficiency of our model, we creatively propose an extremely simple yet effective Channel-Shift Module. This module facilitates information exchanged among

[*]Corresponding author. Email: xmxu@scut.edu.cn

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

neighboring channels through utilizing the depth-wise convolution. More importantly, the parameters in this module are frozen, which means the proposed module is computationally efficient during the back propagation process. Furthermore, the amount of its parameters and FLOPs can be ignored compared with the backbone due to its low computational resources and memory consumption.

Inspired by the aforementioned observations, we propose a simple and highly efficient CS-GResNet for FER. We conduct extensive experiments on three wild datasets: RAF-DB [7], FER2013 [11] and SFEW [12]. The results demonstrate the superiority of our proposed CS-GResNet against the state-of-the-art approaches with less computational and memory cost. We show the corresponding results on the RAF-DB in Fig. 1. The main contributions of this work are summarized as follows:

(1)We propose an effective GResNet to capture the visual salient properties in facial images while retaining computationally efficiency.

(2)We propose a flexible Channel-Shift Module to extract the discriminative information from adjacent channels with very limited extra computational and memory cost.

(3)We propose the CS-GResNet through inserting the Channel-Shift Module into GResNet, which outperforms the latest models and requires less computational resourses.

2. METHOD

This section describes the details of the proposed CS-GResNet for FER. Initially, we incorporate the Gabor filter into the learnable convolution filter to form GConv kernels. Then the GResNet is presented through utilizing the GConv to replace the raw convolution in the ResNet. Furthermore, we insert an innovative Channel-Shift Module into GResNet to produce an extremely simple yet efficient CS-GResNet model. And we feed the facial image into our proposed network to extract the discriminative features for FER.

2.1. GResNet

Gabor Convolution (GConv). To capture the visual salient properties in facial images, we apply the Gabor Convolution (GConv) in our model inspired by [10]. It is computed as follows:

$$C_{i,u}^{v} = C_i \cdot G\left(U, v\right) \tag{1}$$

$$\hat{F}_{i,j} = \sum_{n=1}^{N} F_i^{(n)} \otimes C_{i,u=j}^{v(n)} \tag{2}$$

Eq. 1 is used to obtain the learnable filter $C_{i,u}^{v}$ which is modulated by a group of Gabor filters $G\left(U, v\right)$ with a set of orientations U and a scale v, where C_i represents the traditional convolution filter of the i-th input channel whose shape is $[N, k, k]$. Here k means the kernel size and N is chosen as the length of U to keep the channels quantity consistent during the forward convolution process. $\cdot$ is the element-by-element product operation. Eq. 2 is employed to extract the facial features via leveraging the $C_{i,u}^{v}$. Initially, we will duplicate the i-th input feature N times to get feature map F_i whose shape is $[N, h, w]$, where h, w are the height and width of the feature. $\otimes$ is the convolution operation.

Structure. To avoid the degradation problem and enhance the discriminative features, we propose the GResNet by leveraging the GConv to replace the raw convolution in the ResNet. Specifically, the output of GConv has one more U dimension than Conv, hence we set its channel size in each GConv as $C_{GC\_out} = C_{out}/N$, where C_{out} means the output channel size in each Conv. This operation will greatly reduce the model parameters. In addition, we remove the max pool layer in ResNet, which increases the receptive field of the model. We set U as $[\frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi]$ and v as 3 respectively. The differences of ResNet18 and GResNet18 are shown in Table 1.

Table 1. The structure of ResNet18 and GResNet18. $[\ ]^p$ means the maxpool and the parameters in it represent the kernel size and stride. $[\ ]$ represents the convolution block. The parameters in each column represent the kernel size, output channel and stride (Default: 1). Here we set N as 4.

	ResNet18	GResNet18
Layer 0	$\begin{bmatrix}7*7, 64, 2\end{bmatrix}$	$\begin{bmatrix}4*5*5, 16, 2\end{bmatrix}$
Layer 1	$\begin{bmatrix}3*3, 2\end{bmatrix}^p$ $\begin{bmatrix}3*3, 64\\3*3, 64\end{bmatrix}*2$	$\begin{bmatrix}4*3*3, 16\\4*3*3, 16\end{bmatrix}*2$
Layer 2	$\begin{bmatrix}3*3, 128\\3*3, 128\end{bmatrix}*2$	$\begin{bmatrix}4*3*3, 32\\4*3*3, 32\end{bmatrix}*2$
Layer 3	$\begin{bmatrix}3*3, 256\\3*3, 256\end{bmatrix}*2$	$\begin{bmatrix}4*3*3, 64\\4*3*3, 64\end{bmatrix}*2$
Layer 4	$\begin{bmatrix}3*3, 512\\3*3, 512\end{bmatrix}*2$	$\begin{bmatrix}4*3*3, 128\\4*3*3, 128\end{bmatrix}*2$
Params	11.18M	**2.80M**

2.2. Channel-Shift Module

For capturing the salient representation from adjacent channels while retaining the original spatial information, we propose a simple yet effective Channel-Shift Module (illustrated in Fig. 2). We first divide the input features ($[bs, c, h, w]$) into three parts in the spatial dimension, where bs, c mean the batch size and channel size respectively. These three parts are Shift-right block, Shift-left block and Non-shift block. Their positions in the spatial dimension are $[i * h * w/\sigma : (i+1) * h * w/\sigma]$, $[(i+1) * h * w/\sigma : (i+2) * h * w/\sigma]$ and $[: i * h * w/\sigma \cup (i+2) * h * w/\sigma :]$, where σ and i control the size and position of the blocks respectively. Concretely, we utilize the depth-wise convolution and the one-dimension convolution kernel whose weights are frozen to achieve the above purpose. When the weights are [1, 0, 0], the convolution kernel can shift the channel features to the right. Similarly, when the weights are [0, 0, 1], it can shift them to the left. Noted that, this module is efficient during the back propagation process because its parameters don't need to be updated. Additionally the code about this module is extremely simple and it can be inserted into any models. We show its code based on the PyTorch 1.5.1 in Algorithm 1.

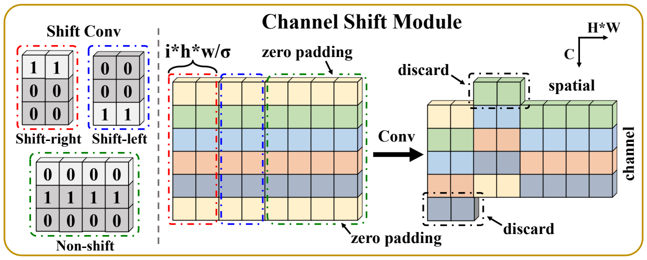

Fig. 2. Illustration of the Channel-Shift Module. We utilize the depth-wise convolution and the one-dimension convolution kernel to shift specified channels in order to enhance the feature from adjacent channels.

3. EXPERIMENTS

3.1. Experimental Setup

Datasets. The proposed method is evaluated on three popular and challenging datasets: RAF-DB [7], FER2013 [11] and SFEW [12]. The Real-world Affective Faces Dataset (RAF-DB) is a large-scale facial expression dataset. In our experiments, we only use the images with basic emotions, including 12,271 images as training data and 3,068 images as test data. The FER2013 dataset is collected from the internet and used for a challenge. It contains a total of 32,295 images of different identities and 3,592 for the test set. The Static Facial Expressions in the Wild (SFEW) dataset is built from AFEW5.0 by using a key-frame extraction method and it has been divided into the train set (958 images) and the test set (436 images). All the three datasets are assigned to the neutral expression or the six basic expressions.

Data Augmentation. To learn the robust features against slight pose changes, we resize the facial images to 130 × 130 and randomly crop them to 112 × 112 as input. Additionally, each image is rotated by random angles in $[-10°,10°]$ and randomly flipped horizontally with a probability of 0.5.

Training Strategies. Firstly, we pretrain the GResNet on the AffectNet dataset [13] and we fine-tune it on three datasets. We use PyTorch1.5.1 and train the whole model for 150 epochs on GTX-1080Ti. The batch size and the learning rate are 64 and 0.005. We use the SGD optimizer with default hyper-parameters. The loss function for FER is the standard cross entropy loss.

Algorithm 1 Channel-Shift Module

Input: The parameter to control the shift block position i; The parameter to control the shift block size σ; The input features x.

Output: The output features x_{out}.

```
% Initialization
$conv = Conv1d(h*w, h*w, 3, padding = 1, groups = h*w, bias = False)$

$b = h*w/\sigma$ % blocksize
$conv.weight.requires\_grad = False$
$conv.weight.data.zero\_()$
$conv.weight.data[i*b : (i+1)*b, 0, 0] = 1$
$conv.weight.data[(i+1)*b : (i+2)*b, 0, 2] = 1$
$conv.weight.data[: i*b, 0, 1] = 1$
$conv.weight.data[(i+2)*b :, 0, 1] = 1$

% Forward
$bs, c, h, w = x.size()$
$x = x.reshape(bs, c, h*w).permute([0, 2, 1])$
$x = conv(x)$
$x_{out} = x.permute([0, 2, 1]).reshape(bs, c, h, w)$
```

3.2. Ablation Study

Effectiveness of GResNet. To analyze the effectiveness of the GResNet, we build up the GResNet18/34 by following 2.1 and compare them with the traditional ResNet18/34 on three popular datasets. The results are illustrated in Fig. 3 and they indicate that GResNet18/34 can achieve the better performance on all datasets with less memory cost. Concretely, they can bring 0.26%/0.17% gains over the Resnet18/34 on the RAF-DB. Similarly, we can see 1.17%/0.17% and 2.07%/2.29% performance gaps over the Resnet18/34 on the FER2013 and SFEW respectively. In addition, our proposed GResNet18/34 reduce about 4 times memory costs comparing with the corresponding ResNet (2.80M vs. 11.17M, 5.33M vs. 21.28M). Noted that, under the comprehensive consideration of recognition accuracy and memory cost, we choose the GResNet18 as our backbone.

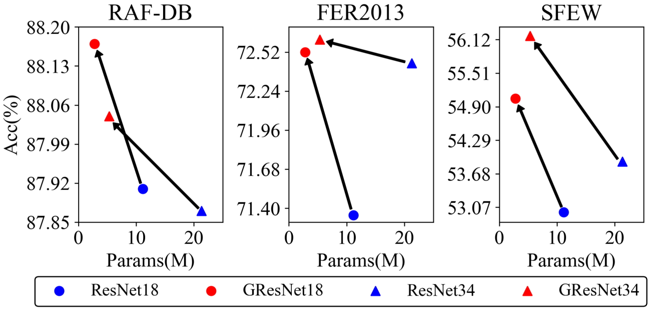

Fig. 3. Accuracy (%) and Parameter size (M) for ResNet and GResNet on three datasets.

Effectiveness of Channel-Shift Module. To study the effectiveness of the Channel-Shift Module and explore the impact of hyper-parameters i and σ, we insert this Module into our proposed GResNet18 to produce the CS-GResNet18 and conduct corresponding ablation studies. Initially, we set the σ in 2.1 as 8, 16, 32. Then we study different positions of the Channel-Shift Module as start, middle and end through setting i as some specified values. In addition, we also study the performance under randomly selecting $h * w/\sigma$ channels for shifting, named as stochastic. Finally, we consider the impact of this Module's position by inserting it into the beginning of the Layer1 or Layer3. The results are depicted in Fig. 4. We observe that almost all the CS-GResNet18 obtain the better performance than our backbone GResNet18. In the single dataset, the results indicate that there is a certain regularity in the setting of hyper-parameters. For example, CS-GResNet18 generally achieves the best results when σ is 16 and the position is Layer3 on three datasets. Specifically, the parameters in this Module do not need to be updated, and it brings few additional computational or memory cost. We depict the cost detail in Table 2.

Table 2. ΔParams(M) and ΔFLOPs(G) on CS-GR18 compared with our backbone GResNet18, where CS-GR18 denotes CS-GResNet18. L1, L3 denotes inserting the Channel-Shift Module into Layer1, Layer3 respectively.

	CS-GR18(L1)	CS-GR18(L3)
ΔParams(M)	$3.8 * 10^{-2}$	$2.0 * 10^{-3}$
ΔFLOPs(G)	$6.0 * 10^{-4}$	$1.5 * 10^{-4}$

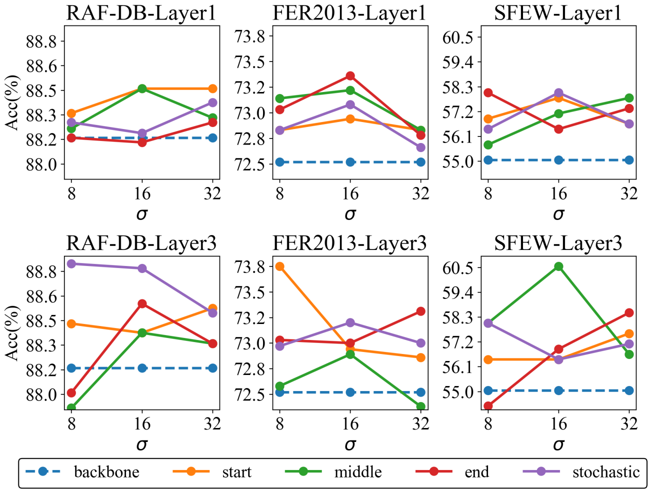

Fig. 4. Accuracy (%) with various settings on three datasets. Layer 1, Layer 3 mean the position of Channel-Shift Module. Start, middle, end, stochastic denote the position of shift block. Backbone represents the performance of the GResNet18. σ controls the block size.

3.3. Comparisons to the State-Of-The-Art Methods

We compare our proposed CS-GResNet with the state-of-the-art methods on RAF-DB, FER2013 and SFEW datasets. The results are shown in Table 3, Table 4 and Table 5. On three wild datasets, our model outperforms the latest methods with very limited extra computational and memory cost.

Table 3. Classification accuracy comparison against state-of-the-art methods on the RAF-DB dataset.

Methods	Year	Params/FLOPs	Accuracy(%)
DLN[14]	2021	–	86.40
SCN[2]	2020	>11.17M/1.80G	87.03
SFR[3]	2021	>11.17M/1.80G	87.35
KTN[4]	2021	>47.04M/7.60G	88.07
MA-Net[1]	2021	50.54M/3.65G	88.40
ADC-Net[5]	2021	>23.6M/3.08G	88.46
PASM[6]	2021	>21.28M/3.60G	88.68
CS-GResNet		**2.80M/1.72G**	**88.85**

Table 4. Classification accuracy comparison against state-of-the-art methods on the FER2013 dataset.

Methods	Year	Params/FLOPs	Accuracy(%)
DAM[15]	2019	>0.13G/11.28G	66.20
SNNs[16]	2021	–	73.00
PASM[6]	2021	21.28M/3.60G	73.59
CS-GResNet		**2.80M/1.72G**	**73.75**

Table 5. Classification accuracy comparison against state-of-the-art methods on the SFEW dataset.

Methods	Year	Params/FLOPs	Accuracy(%)
DDL[17]	2020	–	59.86
RAN[18]	2020	>95.13M/0.10T	56.40
Res-50IBN[2]	2021	>25.52M/3.80G	58.34
MA-Net[1]	2021	50.54M/3.65G	59.40
CS-GResNet		**2.80M/1.72G**	**60.55**

4. CONCLUSIONS

In this work, we propose a highly efficient Channel-Shift Gabor-ResNet (CS-GResNet) for FER. Based on the Gabor Convolution, we design a lightweight GResNet for exploiting the distinctive features of faces. Furthermore, Channel-Shift Module is adopted to extract the informative representation from neighboring channels with very limited computational resources. Extensive experiments on three wild datasets show that our proposed CS-GResNet achieves significant improvement compared to existing approaches, and verify the superiority and high efficiency of our model.

5. ACKNOWLEDGMENTS

The work is supported in part by the National Natural Science Foundation of China under Grant U1801262, 62102102; in part by Natural Science Foundation of Guangdong Province, China, under grant 2019A1515012146, 2020A1515010781; in part by the Science and Technology Project of Guangzhou under grant 202103010002; in part by the Key-Area Research and Development Program of Guangdong under grant 2019B010154003; in part by the Guangzhou Key Laboratory of Body Data Science under grant 201605030011; and in part by the Foundation for Characteristic Innovation in Higher Education of Guangdong Province, China, under grant 2019KTSCX112.

6. REFERENCES

[1] Z. Zhao, Q. Liu, and S. Wang, "Learning deep global multi-scale and local attention features for facial expression recognition in the wild," *IEEE Transactions on Image Processing*, vol. 30, pp. 6544–6556, 2021.

[2] K. Wang, X. Peng, J. Yang, S. Lu, and Y. Qiao, "Suppressing uncertainties for large-scale facial expression recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6897–6906.

[3] L. Lo, H. X. Xie, H.-H. Shuai, and W.-H. Cheng, "Facial chirality: Using self-face reflection to learn discriminative features for facial expression recognition," in *IEEE International Conference on Multimedia and Expo*, 2021, pp. 1–6.

[4] H. Li, N. Wang, X. Ding, X. Yang, and X. Gao, "Adaptively learning facial expression representation via cf labels and distillation," *IEEE Transactions on Image Processing*, vol. 30, 2021.

[5] H.-y. Xia, C. Li, Y. Tan, L. Li, and S. Song, "Destruction and reconstruction learning for facial expression recognition," *IEEE MultiMedia*, vol. 28, no. 2, pp. 20–28, 2021.

[6] P. Liu, Y. Lin, Z. Meng, L. Lu, W. Deng, J. T. Zhou, and Y. Yang, "Point adversarial self-mining: A simple method for facial expression recognition," *IEEE Transactions on Cybernetics*, pp. 1–12, 2021.

[7] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2852–2861.

[8] P. Barros, N. Churamani, and A. Sciutti, "The facechannel: A light-weight deep neural network for facial expression recognition." in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2020, pp. 652–656.

[9] G. Zhao, H. Yang, and M. Yu, "Expression recognition method based on a lightweight convolutional neural network," *IEEE Access*, vol. 8, pp. 38 528–38 537, 2020.

[10] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4357–4366, 2018.

[11] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee *et al.*, "Challenges in representation learning: A report on three machine learning contests," in *IEEE International Conference on Neural Information Processing*, 2013, pp. 117–124.

[12] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2011, pp. 2106–2112.

[13] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18–31, 2017.

[14] W. Zhang, X. Ji, K. Chen, Y. Ding, and C. Fan, "Learning a facial expression embedding disentangled from identity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6759–6768.

[15] S. Xie, H. Hu, and Y. Wu, "Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition," *Pattern Recognition*, vol. 92, pp. 177–191, 2019.

[16] W. Hayale, P. S. Negi, and M. Mahoor, "Deep siamese neural networks for facial expression recognition in the wild," *IEEE Transactions on Affective Computing*, 2021.

[17] D. Ruan, Y. Yan, S. Chen, J.-H. Xue, and H. Wang, "Deep disturbance-disentangled learning for facial expression recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2833–2841.

[18] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao, "Region attention networks for pose and occlusion robust facial expression recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 4057–4069, 2020.