Journal of Imaging Science and Technology® 66(4): 040402-1–040402-9, 2022.

Efficient Recognition of Facial Expression with Lightweight Octave Convolutional Neural Network

Shou-Chuan Lai, Ching-Yi Chen, and Jian-Hong Li
Department of Information and Telecommunications Engineering, Ming Chuan University, Taoyuan, Taiwan, ROC
E-mail: chingyi@mail.mcu.edu.tw

Fu-Chien Chiu
Department of Electronic Engineering, Ming Chuan University, Taoyuan, Taiwan, ROC

Abstract. *In various social activities, people usually focus on the other person's facial expression, which is an important element in interpersonal communication. The facial expression typically reflects a person's current mood and conveys emotional information. Perceiving the facial expressions of people in images through cameras has always been a popular research topic. Previous studies have classified facial expressions into different categories, such as happy, sad, fear, angry, calm, disgust and surprise, and have identified them using image processing methods. However, traditional image processing methods have a low detection efficiency and low recognition accuracy due to variation of perspectives. As a result, most of them can only be applied to the front face and short distance situations. In this paper, we propose a lightweight deep learning framework for facial expression recognition using Octave convolutional neural network (FerOctNet). FerOctNet including multi-scale convolutional processing and residual learning is able to obtain multi-scale features with enriched representation ability by integrating the deep level features with rich semantic information with shallow details of the features. Compared with other deep learning networks, not only does the proposed method have a good recognition rate, but also contains fewer parameters in the network.*

[DOI: 10.2352/J.ImagingSci.Technol.2022.66.4.040402]

1. INTRODUCTION

The facial expressions of human beings can help others to understand one's emotions or even intentions, making it an indispensable communication element in human interaction. With the development of information technology (IT), using computer vision (CV) to analyze and identify specific objects has become an important research theme. Object identification can be achieved as the user captures images through cameras, and then processes and analyzes the image through computer vision techniques. These methods not only are accurate but also can effectively save the cost of labor. Automatic facial expression recognition enables computers to perceive human behavior and emotions, helpful for applications such as human-computer interaction, vehicle driving condition monitoring, emotion and stress resistance recognition in recruitment systems. In 1971, Ekman and Friesen [1] defined six universal human expressions that they believed were present in all humans. That includes happiness, sadness, surprise, disgust, anger, and fear. They further systematically established a database of facial expressions, describing the details of each, which lays a foundation for facial emotion recognition (FER). Facial action units such as eyebrows raised, brows locked, and corners of the mouth moved out are considered as the basic units of facial expressions. However, the production of people's facial expressions is an irregular change, and the expressions presented by different faces also vary from person to person. All these factors make it challenging to realize the task of FER with computer vision techniques.

Early FER studies were achieved by traditional image processing and machine learning techniques. Facial expression features can be divided into two categories: texture-based local features and geometry-based global features. Texture-based features mainly include scale-invariant feature transform (SIFT) [2], histograms of oriented gradients (HOG) [3], and local binary patterns (LBP) [4], etc. Geometric global features mainly use landmark points around the nose, eyes, and mouth to describe facial contour information [5]. This method utilizes a rather small amount of facial features and is less affected by personal appearance. After extracting effective facial features, the feature information are used as the input for the classifier established by machine learning methods such as BP [6] or SVM [7] for classification, so as to obtain the final identification results. However, traditional image analysis methods bear the problem of low efficiency due to difference of perspectives, and the recognition rate does not perform well due to the simplicity of its network architecture. As a result, most of these approaches can only be applied to frontal face emotion recognition with short distances. With the rapid development of GPU and the expansion of expression datasets, these engineered learning methods have been gradually replaced by deep learning algorithms dominated by convolutional neural networks (CNN) when processing large and complex data [8–10]. Deep learning approaches for facial expression approaches use the powerful feature detectors of deep CNN to extract facial expression features, and achieve high performance by deepening the layers of the network and developing effective learning mechanisms. Therefore, they are robust for various face

Received Aug. 8, 2021; accepted for publication Dec. 5, 2021; published online Jan. 20, 2022. Associate Editor: Jia-Shing Sheu.
1062-3701/2022/66(4)/040402/9/$25.00

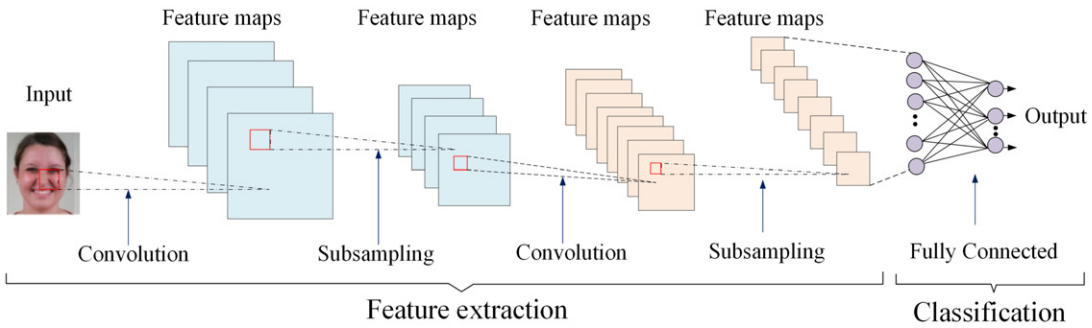

Figure 1. Typical CNN architecture.

positions and scale changes [11–13]. Though these methods are effective and have gained huge success, deep CNN models often carry significant computational cost and require large scale of parameters.

To apply deep learning to image recognition more efficiently, some researchers view the image with different aspects. Each image can be decomposed into high-spatial frequency components and low-spatial frequency components according to the changes among its pixels. High spatial frequencies represent abrupt spatial changes in the image, such as edges, and generally correspond to feature information and fine detail; low spatial frequencies represent global information about the shape, such as general orientation and proportions. Deep network architectures applied in computer vision mostly focus on the design of the convolution architectures with multi-scale feature aggregation in spatial information in order to obtain different information in different receptive fields. However, few studies have explored how to integrate information in different frequency domains of images in convolutional neural networks. Octave Convolution (OctConv) is a new convolution architecture that decomposes and integrates information of different frequencies [14], which can achieve better recognition results while reducing the computation amount of image classification tasks. OctConv is very lightweight because of its decomposition mechanism. In light of this, we integrate OctConv with FER to obtain a lightweight efficient system for facial expression systems. In this study, a lightweight deep learning architecture for facial expression recognition based on OctConv (FerOctNet) is proposed for FER, which integrates two mechanisms including multi-scale OctConv and residual learning, and then fuses the extracted high-level features and low-level features. FerOctNet has the advantage of lower operational cost. Compared with the vanilla CNN-based methods, the proposed FerOctNet effectively reduces the redundancy in the spatial dimension of feature maps and the redundancy in dense model parameters, while obtaining good identification performance in FER tasks.

2. RELATED APPROACHES

2.1 *Convolutional Neural Network*

CNN has two key concepts: local connectivity and parameter sharing, which extract the features of an image. A CNN comprises convolution layers, pooling layers, and fully connected layers. As the main building block of a CNN, the convolution layer is composed of dozens of $N \times N$ filters. After the convolution operation, each filter will generate a corresponding feature map. The input feature map will then be down-sampled by the pooling layer connected after the convolution layer. Two or more fully connected layers will render the prediction results based on the features extracted by the convolution layer and pooling layer. Figure 1 demonstrates the architecture of a CNN. With the development of deep learning, researchers have adopted CNN approaches to enhance the performance of the facial expression recognition. However, vanilla CNN networks have the drawback of inefficiency, and large deep CNN models may be expensive and unsuitable for common applications.

2.2 *Octave Convolution*

A natural image can be decomposed into spatial frequencies, including low spatial frequency component and high spatial frequency component. A low spatial frequency component completely wipes out the sharp transitions of reflectance across a direction while preserving the slow transitions of reflectance in an image; a high spatial frequency preserves the edge of represented objects with a sudden change in light reflectance. Chen et al. [14] posited that the output feature map of a convolution layer could be regarded as the mixture of spatial frequencies, in their work, a convolution method named OctConv was proposed to process feature maps by their frequencies. In [14], the author believes that factorizing the mixed feature maps based on their frequencies, storing, and processing feature maps that are spatially slower at a lower spatial resolution, can reduce storage and computation cost.

Figure 2 shows how OctConv works. Fig. 2(a) shows that the output diagrams of the convolution layer can also be factorized and grouped according to their spatial frequency; the following Fig. 2(b) shows that we can store the low-frequency feature map with gentle changes in the low-resolution tensor to reduce spatial redundancy; Fig. 2(c) shows how to use OctConv to update the information of each group and further realize the information exchange between groups.

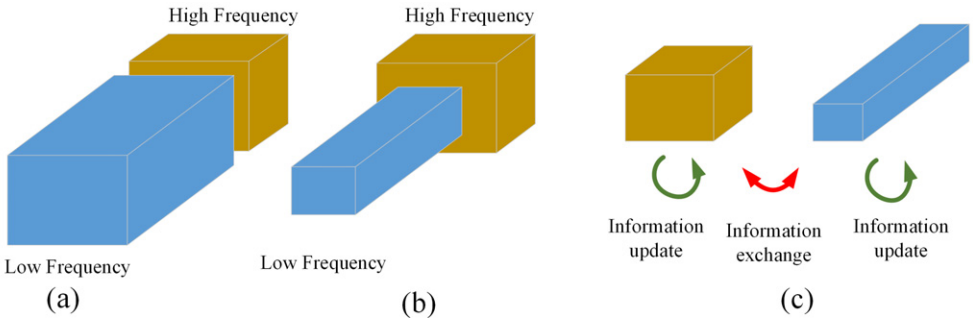

Figure 2. A demonstration of how OctConv works [14].

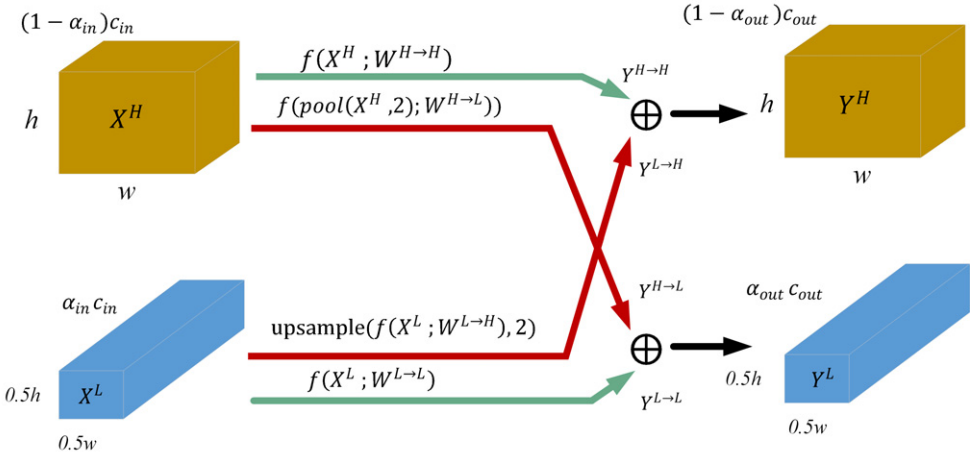

Figure 3. The detailed design of OctConv [14].

The goal of OctConv is to effectively deal with low and high frequencies in the corresponding frequency tensor, at the same time allowing effective communication between the high and low frequency components of Octave feature representation.

Let $X \in R^{c\times h\times w}$ be the input feature tensor while the output feature tensor is denoted as Y, where h and w represents spatial dimensions, c is the depth of the channel. X can be factorized to $X = \{X^H, X^L\}$, where $X^H \in R^{(1-\alpha)c\times h\times w}$, $X^L \in R^{\alpha c\times h/2\times w/2}$. $\alpha \in [0, 1]$ indicates the ratio of channels allocated to the low-frequency part.

The high and low frequency feature maps of output $Y = \{Y^H, Y^L\}$ can be denoted as follows:

$$Y^H = Y^{H\to H} + Y^{L\to H} = f\left(X^H; W^{H\to H}\right) + \text{unsample}\left(f\left(X^L; W^{L\to H}\right), 2\right) \quad (1)$$

$$Y^L = Y^{H\to L} + Y^{L\to L} = f\left(X^L; W^{L\to L}\right) + f\left(pool(X^H, 2); W^{H\to L}\right), \quad (2)$$

where $Y^{A\to B}$ denotes the convolutional update from feature map group A to group B. $f(X; W)$ denotes a convolution with parameters W, $pool(X, k)$ is a pooling operation with kernel size $k \times k$ and stride k. upsample(X, k) is an up-sampling operation by a factor of k via nearest interpolation. Figure 3 shows the detailed design of OctConv, where the green arrow indicates the information updates, and the red arrow indicates the information exchange between two frequencies.

From the formula and flow chart of OctConv, it could be seen that Y^H and Y^L are composed of high and low frequencies respectively, which are the green arrow and the red arrow in Fig. 3. This enables the information change between two frequencies. An up-sampling operation is performed when the low frequency part is transformed into high frequency while average pooling is performed when high frequency is transformed into low frequency.

2.3 *Inception in GoogLeNet*

The most direct way to improve the performance of a neural network is to deepen it, having the drawback of dramatically increasing the number of parameters and computation cost. To enhance the performance of a deep neural network without increasing costs, Szegedy et al. [15] proposed GoogLeNet utilizing the Inception Modules in 2013. GoogLeNet does not enhance the performance by deepening the depth of the network. Instead, it extracts image information of different scales through different size convolutional kernels before merging the extracted features of these channels to obtain more features and details from

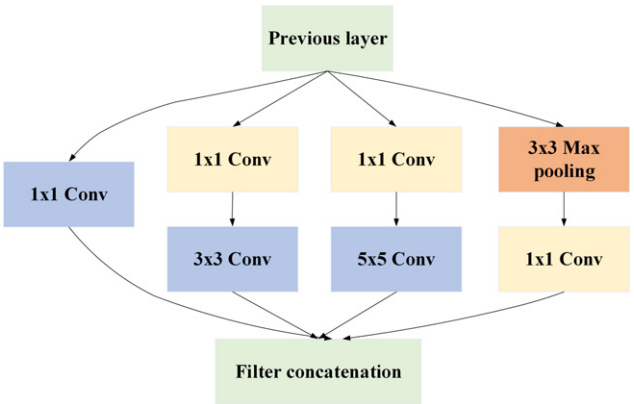

Figure 4. Inception module with dimension reductions.

an image. The concept of the Inception module is shown in Figure 4. In Fig. 4, 1×1 convolution operation is performed before 3×3 convolution and 5×5 convolution of the input image, which achieves dimension reduction. Utilizing the Inception module reduces the number of parameters required for the deep model. The Inception module is used on FerOctNet in order to enhance the performance of facial recognition systems as well as making FerOctNet more lightweight.

2.4 *Shortcut Connection in ResNet*

As the number of network layers increases, it can achieve better results theoretically, as the network can extract more complex feature patterns. However, when increasing the number of layers in a neural network, a deep network is generally accompanied by overfitting, gradient vanishing, and degradation. A residual network uses identity mapping to solve deep neural network degradation caused by too many layers in deep neural networks [16].

We can define the residual function $F(x) = H(x) - x$, as shown in Figure 5. When residual is 0, the residual block will only get the identity mapping, where the performance of the neural network can at least be guaranteed not to decrease after the addition of the blocks of the identity map. When the residual is not 0, the stacked layer learns new features from the input, so as to achieve better performance. This method is also integrated in FerOctNet to avoid gradient vanishing when building the model more deep and structured.

3. FACIAL EXPRESSION RECOGNITION USING OCTNET

In this study, we propose FerOctNet, a high performance facial expression recognition architecture based on OctNet. Our proposed FerOctNet system uses multi-scale OctConv to extract features of different scales in the image, so as to achieve the purpose of widening and deepening the network. Meanwhile, residual learning and batch normalization (BN) are used to avoid network degradation. The structure is described as follows.

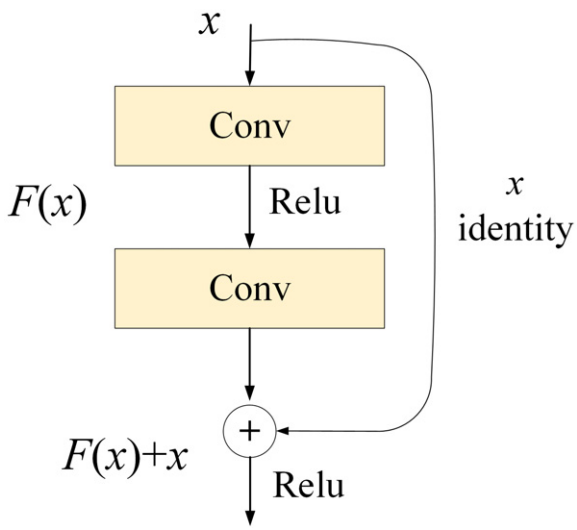

Figure 5. The structure of a basic residual block.

3.1 *The Multi-scale OctConv Module with Shortcut Connection*

In order to extract more detailed information and features from the image, three different channel ratios of $\alpha = 0.125$, $\alpha = 0.25$ and $\alpha = 0.5$ is used respectively to build multi-scale OctConv modules. In the module, each OctConv connects to a BN. The residual learning mechanism is also added to retain the information of the previous layer feature map. Finally, we concatenate all the output features OctConv and BN with the shortcut connection of the residual learning mechanism, as shown in Figure 6.

3.2 *The Framework of FerOctNet*

We use multi-scale OctConv module to stack the main architecture of the deep model. In addition to standardizing the data using BN, dropout is also added to the model to prevent overfitting during training. At the output of the network, softmax is used to transform the output value of neurons into the probability distribution of input samples belonging to each category. The overall framework of FerOctive is shown in Figure 7.

(1) Input flow: After entering Block1, the input image will enter multi-scale OctConv module1 (kernels = 64; α = 0.5, 0.25, and 0.125). The input image will be converted from $48 \times 48 \times 3$ to $12 \times 12 \times 64$ feature map after entering Block2 and Block3. Block3 contains a 1×1 Conv, which reduces the dimension of the feature map to reduce the number of parameters.

(2) Middle flow: Different from 64 kernels for all convolution operations in Input flow, 128 kernels are set for all convolution operations in Middle Flow. Middle Flow receives the feature map sent from the Input flow. After the convolution operation of Block4, the Middle flow enters the multi-scale OctConv Module2 (Kernels = 128; $\alpha = 0.5$, 0.25, and 0.125). The output features will then pass through Block5 and Block6 ($1X1$ Conv, BN,

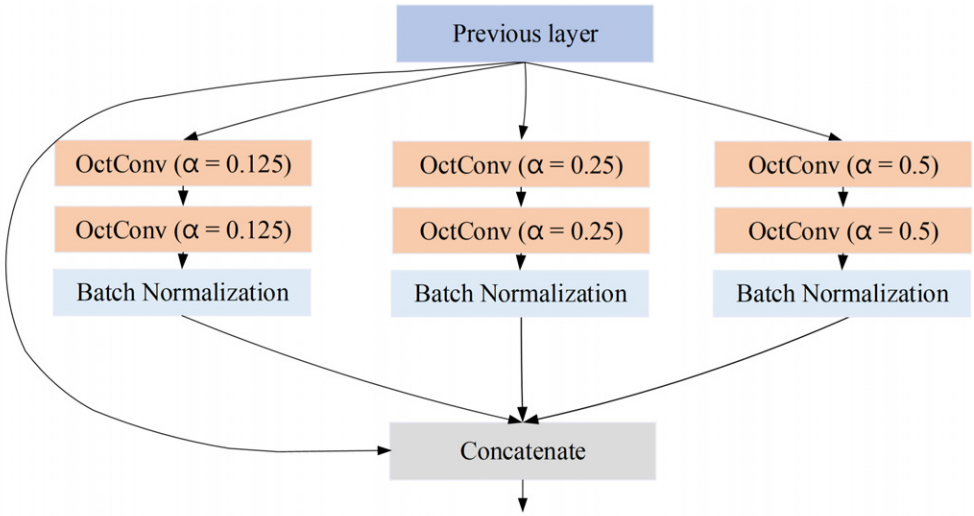

Figure 6. The multi-scale OctConv module.

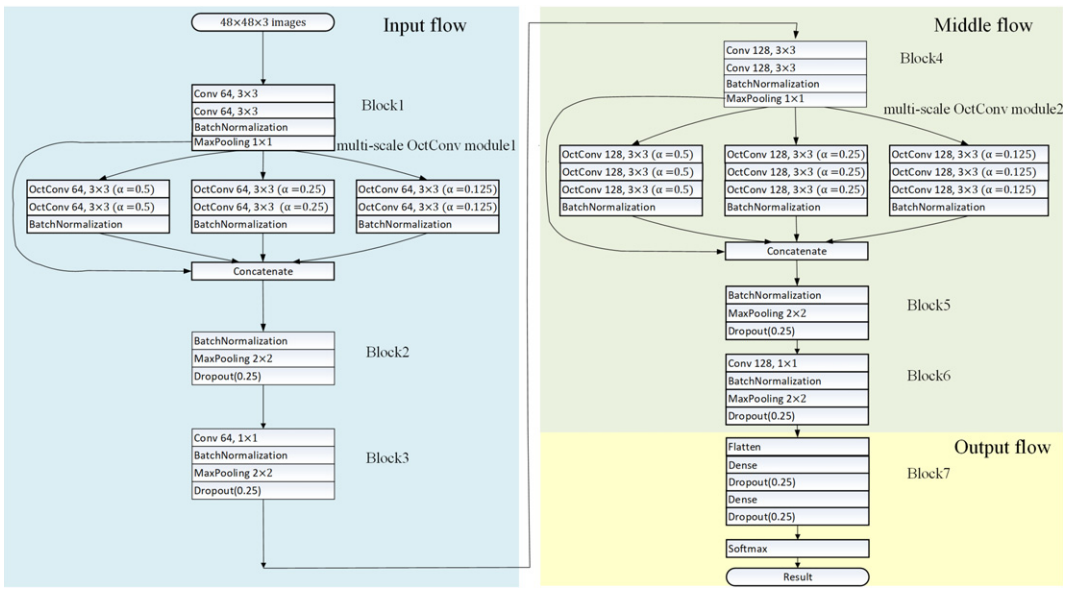

Figure 7. The framework of FerOctNet.

Maxpooling, and Dropout), the dimension of the feature map is transformed from $12 \times 12 \times 64$ to $3 \times 3 \times 128$.

(3) Output flow: The feature map transmitted from the Middle flow will be transformed into a one-dimensional vector after the Flatten layer, which can be used as the input value of the fully connected network. The output of the second hidden layer and Softmax layer (Block7) will be the final result.

4. EXPERIMENTAL RESULTS AND DISCUSSION

In this study, we use TensorFlow Core v2 as the deep learning framework. A computer with Core(TM) I5-6500 CPU, 8GB DDRIII RAM is used to conduct experiments in this study. We use a 3GB NVIDIA GeForce GTX 1060 to train our model.

4.1 *Performance on Public Datasets*

Fer2013 [17] and CK+ [18] are two datasets widely applied in FER research. Many studies in FER use these two datasets for verification. In this study too, these two datasets are used to conduct the training and testing processes to verify the performance of the FerOctNet for FER tasks.

4.1.1 *Performance on the Fer2013 dataset*

There are 28709 training data points and 3589 test data points in the Fer2013 dataset. Each data point is a 48×48 image of a human face. The facial expressions in the database

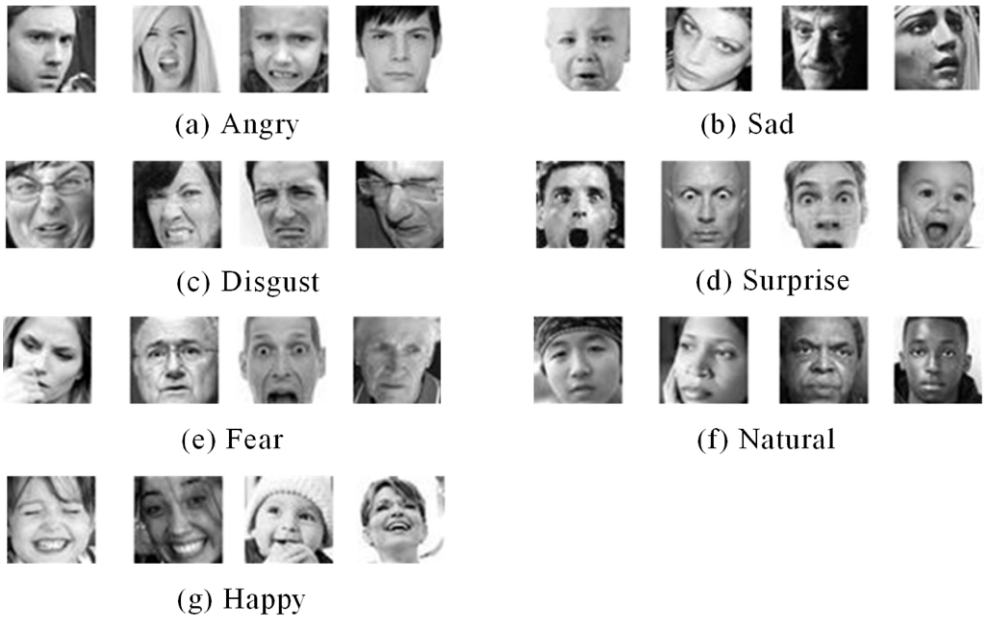

Figure 8. Partial pictures of the Fer2013 dataset.

Table I. Performance of different methods on the Fer2013 dataset.

Method	Accuracy
Pham & Won's model [11]	70.12%
Miao et al.'s model [19]	69.10%
Hung et al.'s model [20]	70.02%
Fei & Jiao's model [21]	71.19%
Our model	**72.78%**

are divided into seven categories, that include Angry, Sad, Disgust, Surprise, Fear, Neutral, and Happy, as shown in Figure 8.

(1) Evaluation of the identification rate on the Fer2013 dataset

In this study, we choose Adam as the optimizer for our experiment, and trained with data augmentation provided by the TensorFlow Keras API for 100 iterations. The identification rate of FerOctNet on Fer2013 testing data set is 72.78%, as shown in Table I. As can be seen from Table I, compared with many different deep learning models, our proposed FerOctNet have better identification performance for Fer2013.

(2) Performance comparison analysis on the Fer2013 dataset

Table II is the confusion matrix generated by the proposed FerOctNet tested on the Fer2013 dataset. The y-axis indicates the ground truth and the x-axis is the predictions made by our proposed model. According to the confusion matrix, it can be seen that the proposed method can achieve good performance for faces expressing Disgust, Happy, and Surprise, but fail to recognize Angry, Fear, Sad and Neutral facial expressions. Figure 9 lists few incorrectly

Table II. The confusion matrix of the proposed method tested on the Fer2013 dataset (Accuracy : 72.78%). Note that the y-axis is the ground-truth label, while the x-axis indicates the predicted labels.

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	76%	2%	4%	5%	6%	1%	7%
Disgust	2%	93%	0%	1%	1%	0%	1%
Fear	12%	2%	37%	12%	17%	9%	11%
Happy	2%	0%	1%	90%	3%	1%	2%
Sad	9%	1%	6%	8%	64%	2%	10%
Surprise	0%	0%	2%	3%	1%	92%	2%
Neutral	5%	0%	4%	13%	14%	3%	61%

recognized facial expression images. It can be seen that our model makes the most mistakes when predicting the label Fear, with an accuracy of 63%. For emotions with more dramatic facial expressions, such as Disgust, Happy, and Surprise, the model has a very high recognition rate. The potential reason for the failure of recognition is that the Fer2013 dataset contains many images with various face angles. In addition, many facial expression category labeling is based on personal subjective judgment, which is not easy to distinguish. All these factors lead to the low recognition rate of existing research results for this dataset.

4.1.2 *Performance on the CK+ dataset*

CK+ Dataset, released in 2010, is an extension of the Cohn-Kanade Dataset. The dataset contains facial images recorded by 118 participants, including 123 subjects with 7 emotions. Established under laboratory conditions, the CK+ Dataset is regarded as a relatively rigorous and standard dataset. Therefore, many researchers use the CK+ dataset

Figure 9. Images that our model failed to recognize in the Fer2013 dataset. The words in black font are the actual label of the image, while the red fonts are the false predicted labels.

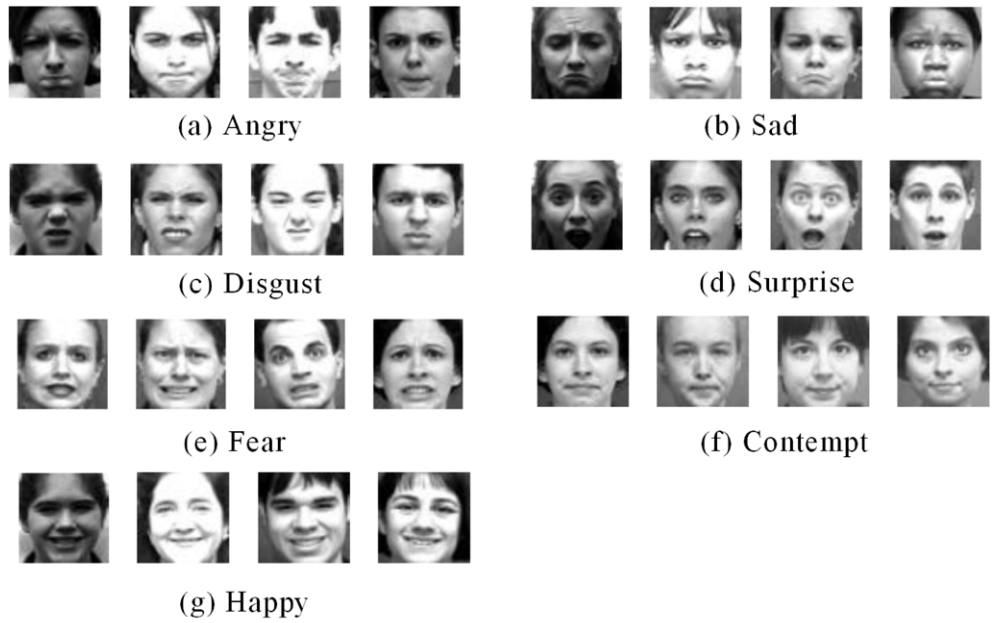

Figure 10. Partial pictures of CK+ database.

for FER testing and comparison. FER. Figure 10 shows few samples of images from the CK+ Dataset.

(1) Evaluation of the identification rate on the CK+ dataset

In the CK+ DATASET experiment, we also chose Adam as the optimizer and trained with data augmentation provided by the TensorFlow Keras API. The identification rate of FerOctNet against the CK+ dataset obtained after the training is 97.30%, as shown in Table III. According to the data listed in Table III, the identification rate of the CK+ dataset in the general research literature is about 94%–96%, which shows that the FerOctNet model proposed in this study has a better identification rate among many methods listed in Table III.

(2) Performance comparison analysis on the CK+ dataset

Table IV is the confusion matrix generated by the proposed FerOctNet model tested on the CK+ dataset. Since the data points of the CK+ dataset was obtained under laboratory conditions, the dataset is relatively rigorous and reliable. Note that the y-axis indicates the ground truth and the x-axis is the predictions made by our proposed model. Therefore, we can see from the analysis results

Table III. Performance of different methods on the CK+ dataset.

Method	Accuracy
Happy & Routray's model [22]	94.09%
Sun & Wen's model [23]	94.87%
Lopes et al.'s model [24]	95.75%
Yan's model [25]	96.60%
Sariyanidi et al.'s model [26]	96.02%
Our model	**97.30%**

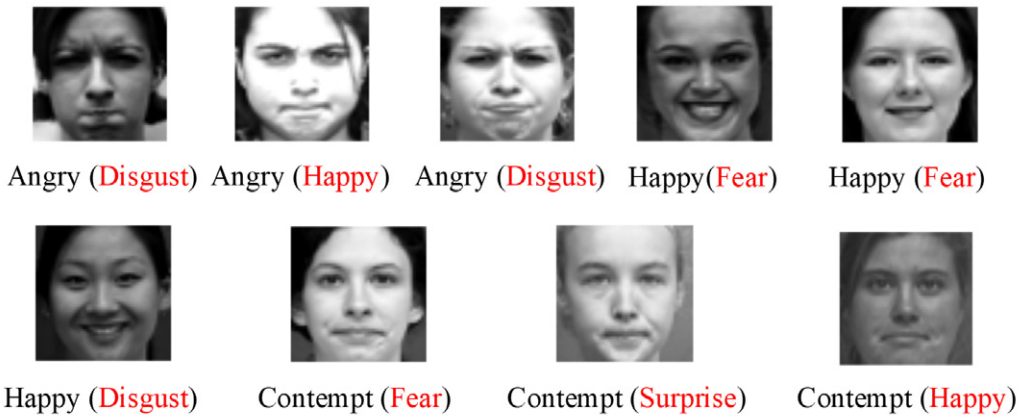

Figure 11. Images that our model failed to recognize in the CK+ dataset. The words in black font are the actual label of the image, while the red fonts are the false predicted labels.

Table IV. The confusion matrix of the proposed method tested on the CK+ (Accuracy: 97.3%). Note that the *y*-axis is the ground-truth label, while the *x*-axis indicates the predicted label.

	Angry	Disgust	Fear	Happy	Sad	Surprise	Contempt
Angry	95%	0%	0%	5%	0%	0%	0%
Disgust	0%	100%	0%	0%	0%	0%	0%
Fear	0%	0%	100%	0%	0%	0%	0%
Happy	0%	10%	0%	90%	0%	0%	0%
Sad	0%	0%	0%	0%	100%	0%	0%
Surprise	0%	0%	0%	0%	0%	100%	0%
Contempt	0%	0%	0%	0%	0%	4%	96%

of the confusion matrix that the proposed method has ideal performance on all 7 categories of facial expressions. Occasional errors were noted in identifying facial images with Angry, Happy, and Contempt labels. Figure 11 shows few facial expression images that failed to be recognized, including those images that our model misjudged Anger as Disgust and Happy as Fear.

4.2 *Ablation Study on the Performance of FerOctNet and Vanilla CNN*

Table V shows the comparison of identification rates between FerOctNet and Vanilla CNN. Here, we converted all OctConvs in the FerOctNet model shown in Fig. 7 into general convolution layer, and built a Vanilla CNN model with the exact same structure as FerOctNet. For the ablation study, we compare the identification performances of FerOctNet and Vanilla CNN on the Fer2013 dataset and the CK+ dataset. The Vanilla CNN model has an identification rate of 71.12% on the Fer2013 and 92.56% for the CK+ dataset, while FerOctNet achieves 72.7% and 97.30% on the two datasets, respectively. The result shows that OctConv performs better than traditional Conv in the same operating environment. During the training process, OctConv can achieve better identification performance in the FER task than Vanilla CNN, at the same time

Table V. The performance comparison of Vanilla CNN and OctConv.

	Fer2013	CK+
Vanilla CNN-based method	71.12%	92.57%
FerOctNet	72.78%	97.30%

effectively reduce the redundancy in the spatial dimension of feature maps and the parameter redundancy in dense model parameters. This is because OctConv is able to deal with the low and high frequencies from an image in the corresponding frequency tensor, and since it allows the communication between the two frequencies component, adapting this module for facial expression recognition allows the model to capture more useful information. Experimental results demonstrates the effectiveness of the proposed method, and show the capability of improving the performance on the CK+ dataset.

4.3 *Parameter Efficiency Evaluation of* 1×1 *Conv*

When 1×1 Conv is applied to a multi-channel feature map, it works as a linear combination of each pixel of different channels to achieve the purpose of dimensionality reduction while retaining the original image structure. In Table VI, we discuss the impact of using 1×1 Conv on the amount of model parameters. Following the previous experimental settings in Sect. 4.1, Adam is used as the optimizer for the experiment of ablation studies experiment, trained for 100 iterations. The FerOctNet model mentioned in Fig. 7 is configured with a convolutional layer with a filter size of 1×1 in both the Input flow and the Middle flow. The number of parameters of the proposed model is 2,542,311. When the 1×1 Conv of the Input flow is removed, the parameter amount of the model increased to 4,516,263; if the 1×1 Conv in the Input flow and Middle flow are removed, the parameter amount of the model becomes a huge amount of 39,839,527. Compared with the deep model architecture with similar purposes in the literature, the parameters used in the FerOctNet model are reduced by 63.7% compared to the

Table VI. Ablation study on the numbers of parameters of models using 1 × 1 Conv.

Deep model architecture	Total Params
Without 1 × 1 Conv	39,839,527
1 × 1 Conv in Middle flow only	4,516,263
1 × 1 Conv in both Input flow and Middle flow	2,542,311

traditional CNN model used for FER [27], and approximately reduced respectively compared to VGG19 and ResNet18 [28] 93.8% and 76.4%. It can be seen that the parameters are fewer and the model constructed is very effective.

5. CONCLUSION

In this study, a parameter-efficient deep learning model for facial expression recognition, FerOctNet, was proposed with a high identification rate. The proposed architecture uses a multi-scale OctConv module to detect facial features of different scales, reducing redundancy in the spatial dimension of feature maps. In addition, we use the residual learning mechanism and 1 × 1 Conv to avoid the network degradation problem while effectively reducing the number of parameters of the model. FerOctNet achieves a good recognition performance for facial expressions recognition with parameter efficiency and low computation cost. Experimental results show that the identification rate of our method in the Fer2013 dataset and CK+ dataset reaches 72.43% and 97.3%, respectively. Moreover, the number of parameters of the proposed model is reduced by 63.7% when compared with the traditional CNN architecture used for facial expression recognition and reduced by about 93.8% and 76.4% compared with VGG19 and ResNet18.

REFERENCES

[1] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," J. Pers. Soc. Psychol. **17**, 124–129 (1971).

[2] P. C. Ng and S. Henikoff, "Sift: Predicting amino acid changes that affect protein function," Nucleic Acids Res. **31**, 3812–3814 (2003).

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05)* (IEEE, Piscataway, NJ, 2005), pp. 886–893.

[4] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," Image Vis. Comput. **27**, 803–816 (2009).

[5] F. Khan, "Facial expression recognition using facial landmark detection and feature extraction via neural networks," arXiv:1812.04510v3 (2018).

[6] L. Ma and K. Khorasani, "Facial expression recognition using constructive feedforward neural networks," IEEE Trans. Syst. Man. Cybern. B Cybern. **34**, 1588–1595 (2004).

[7] L. Chen, C. Zhou, and L. Shen, "Facial expression recognition based on SVM in e-learning," IERI Procedia **2**, 781–787 (2012).

[8] J. S. Chiang, C. H. Hsia, H. J. Chen, and T. J. Lo, "VLSI architecture of low memory and high speed 2D lifting-based discrete wavelet transform for JPEG2000 applications," *2005 IEEE Int'l. Symposium on Circuits and Systems* (IEEE, Piscataway, NJ, 2005).

[9] C. H. Hsia, J. M. Guo, and C. S. Wu, "Finger-vein recognition based on parametric-oriented corrections," Multimedia Tools Appl. **76**, 25179–25196 (2017).

[10] C. H. Hsia, "Improved finger-vein pattern method using wavelet-based for real-time personal identification system," J. Imaging Sci. Technol. **62**, 030402-1–030402-8 (2018).

[11] X. Chen, X. Yang, M. Wang, and J. Zou, "Convolution neural network for automatic facial expression recognition," *2017 Int'l. Conf. on Applied System Innovation (CASI)* (IEEE, Piscataway, NJ, 2017).

[12] T. T. D. Pham and C. S. Won, "Facial action units for training convolutional neural networks," IEEE Access **7**, 77816–77824 (2019).

[13] S. Miao, H. Xu, Z. Han, and Y. Zhu, "Recognizing facial expressions using a shallow convolutional neural network," IEEE Access **7**, 78000–78011 (2019).

[14] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng, "Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution," arXiv:1904.05049 (2019).

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," arXiv:1409.4842v1 (2014).

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv:1512.03385 (2018).

[17] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hammer, W. Cukierski, Y. Tang, D. Thaler, D. H. Lee, and Y. Zhou, "Challenges in representation learning: A report on three machine learning contests," Neural Netw. **64**, 59–63 (2015).

[18] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," *2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition - Workshops* (IEEE, Piscataway, NJ, 2010), pp. 94–101.

[19] S. Miao, H. Xu, Z. Han, and Y. Zhu, "Recognizing facial expressions using a shallow convolutional neural network," IEEE Access **7**, 78000–78011 (2019).

[20] J. C. Hung, K. C. Lin, and N. X. Lai, "Recognizing learning emotion based on convolutional neural networks and transfer learning," Appl. Soft Comput. **84** (2019).

[21] Y. Fei and G. Jiao, "Research on facial expression recognition based on voting model," IOP Conf. Ser.: Mater. Sci. Eng. **646** (2019).

[22] S. L. Happy and A. Routray, "Automatic facial expression recognition using features of salient facial patches," IEEE Trans. Affect. Comput. **6**, 1–12 (2015).

[23] Y. Sun and G. Wen, "Cognitive facial expression recognition with constrained dimensionality reduction," Neurocomputing **230**, 397–408 (2017).

[24] A. Lopes, E. Aguiar, A. Souza, and T. Olivera-Santos, "Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order," Pattern Recognit. **61**, 610–628 (2017).

[25] H. Yan, "Collaborative discriminative multi-metric learning for facial expression recognition in video," Pattern Recognit. **75**, 30–40 (2018).

[26] E. Sariyanidi, H. Gunes, and A. Cavallaro, "Learning bases of activity for facial expression recognition," IEEE Trans. Image Process. **26**, 1965–1978 (2017).

[27] J. Li and E. Y. Lam, "Facial expression recognition using deep neural network," *IEEE Int'l. Conf. on Imaging Systems and Techniques (IST)* (IEEE, Piscataway, NJ, 2015).

[28] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim, "Rotating your face using multi-task deep neural network," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015)* (IEEE, Piscataway, NJ, 2015) pp. 676–684.