EAI Endorsed Transactions
on e-Learning

Research Article EAI.EU

Facial expression recognition via transfer learning

Bin Li[1,*]

[1]School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, P R China

Abstract

INTRODUCTION: With the development of artificial intelligence, facial expression recognition has become a hot topic. Facial expression recognition has been widely applied to every field of our life. How to improve the accuracy of facial emotion recognition is an important research content.
OBJECTIVES: In today's facial expression recognition, there are problems such as weak generalization ability and low recognition accuracy. Aiming to improve the current facial expression recognition problems, we propose a novel facial emotion recognition method.
METHODS: This paper focuses on the deep learning-based static face image expression recognition method, and combines transfer learning and deep residual network ResNet-101 to realize facial expression recognition.
RESULTS: The simulation results show that the overall accuracy of our method is 96.29± 0.78%.
CONCLUSION: The performance of this model is superior to the current mainstream face emotion recognition models. In the future research, we will try other methods based on deep learning.

Keywords: Deep residual network, Facial expression recognition, ResNet-101, Transfer learning.

Received on 29 January 2021, accepted on 05 April 2021, published on 08 April 2021

doi: 10.4108/eai.8-4-2021.169180

*Corresponding author. Email: 827843449@qq.com

1. Introduction

Facial expression plays an important role in the emotional expression of people's daily communication, and is one of the most important clues to identify human emotion and behavior. It is defined as the facial changes corresponding to people's inner emotional state, intention or social information [1]. As early as the 20th century, Ekman defined six basic facial expressions based on cross-cultural studies. These typical facial expressions are anger, disgust, fear, happiness, sadness and surprise. Facial expression recognition has a wide range of applications, such as human-computer interface, interactive games, online/distance education, criminal investigation and business analysis, etc. Facial expression recognition is a traditional problem in the field of computer vision. As an important part of intelligent human-computer interaction technology, it has received extensive attention in recent years, and many new methods have emerged [2].

According to different input resources, facial expression recognition system can be divided into two main types, namely input static image and dynamic image sequence. The method of static image only extracts the feature image from the current input, while the method of image sequence can extract the time information of the image sequence and the feature of each static image. According to the study, traditional hand-extracted features cannot address factors unrelated to facial expression. Traditional feature classifiers need to extract a large number of features manually. However, these features are not representative, and traditional classifiers require many parameters to be set manually and are subject to many manual influences. With the development of deep learning, its ability of automatic feature extraction has been recognized by many scholars, and the recognition task can be completed with the original image. This paper focuses on the deep learning based static face image expression recognition method.

With the development of machine learning and deep learning, there are more and more methods to realize facial

expression recognition. Drume and Jalal [3] combined principal component analysis (PCA) and support vector machine. Ali, et al. [4] employed high-order spectra (HOS) for face recognition. Shih and Chuang [5] proposed the use of support vector machine (SVM) method. Evans [6] presented to use Haar wavelet transform (HWT) method. Yang [7] utilized cat swarm optimization to recognize facial emotions. Li [8] chose to use biogeography-based optimization (BBO) for the same task. However, through the analysis of the above methods, it is found that there are some problems such as low accuracy and low robustness. It also has the problem of losing original information.

Therefore, in order to solve the above problems, we propose a method to improve facial expression recognition. We selected the deep residual network ResNet-101 to extract the features of the image to complete the classification task, and combined with transfer learning to improve the efficiency of training. Through comparative analysis, it can be found that our method is superior to existing methods.

2. Dataset

In the process of facial expression recognition training, it is very important to use enough effective label data for training. In order to make our experimental results more comparable and convincing, the data set used in this paper contains a total of 700 images. The data set adopted in this paper is a new data set obtained by using a face model in reference [9] data set. Among them, reference [9] is a facial model proposed by F.Y. Shih and C.F. Chuang. Figure 1 has showed the dataset we used. The data set in reference [9] was collected by an experienced photographer who used the canon digital camera to capture the facial expressions of each subjects ten times for 20 subjects of different ages, different careers and different race, including seven kinds of facial emotions pictures: happy, sadness, fear, anger, surprise, disgust, and neutral.

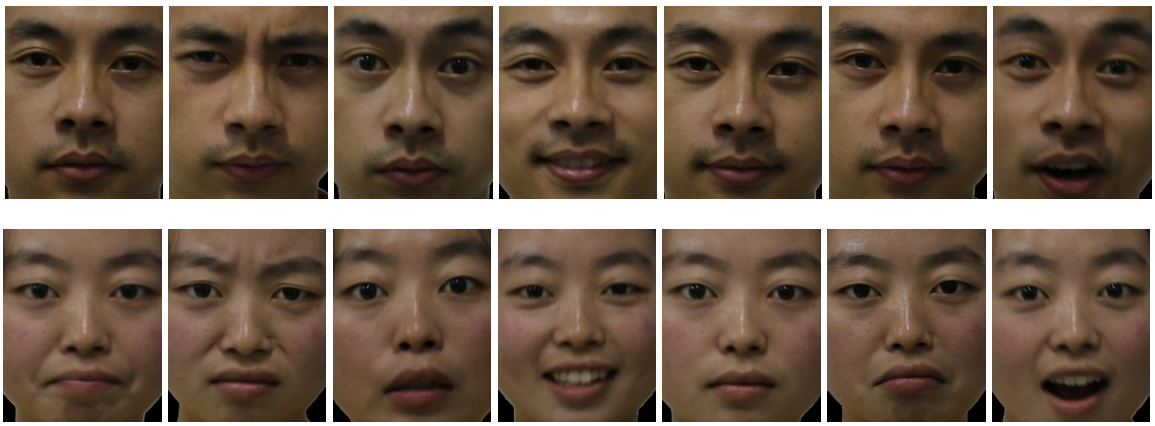

Figure 1. Samples of our dataset

3. Methodology

3.1. Convolution

Convolution has two typical application scenarios: 1) Signal analysis: an input signal f(t), whose characteristics can be described by the unit impulse response function g(t). After passing through a linear system, the output signal can be obtained through convolution operation [10]. 2) Image processing: input an image $f(x,y)$, after convolution processing with specially designed convolution kernel $g(x,y)$, the output image will get various effects such as blurring and edge enhancement [11]. The convolution of two functions is essentially flipping a function over and then sliding over it. In the continuous case, superposition refers to the integration of the product of two functions; in the discrete case, it is the weighted sum; for simplicity, it is called superposition.

$$\mathbf{y(t)} = \int_{-\infty}^{\infty} \boldsymbol{x(p)h(t-p)dp} = \boldsymbol{x(t)} * \boldsymbol{h(t)} \quad (1)$$

The $x(t)$ and $h(t)$ functions are the convolution variables, p is the integral variable, t is the amount that shifts $h(-p)$, asterisk $*$ denotes the convolution. In the operation of convolution, we first flip the h function [12], which is equivalent to crimping the h function from the right to the left on the number line, and that is where the "reel" of convolution comes from. And then we shift the h function to t, where we multiply the corresponding points of the two functions, and then add them, and this is the “product” of convolution [13].

As you can see from the process of the “product”, the sum that we get is a global concept. Taking signal analysis as an example, the result of convolution is not only related to the response value of the input signal at the current moment, [14] but also related to the response value of the input signal at all past times, considering the accumulation

of the effect of all past inputs [15]. In image processing, the result of convolution processing is to take into account the surrounding pixels of each pixel, or even the pixels of the whole image, and carry out some kind of weighted processing on the current pixel [16]. So, the product is a global concept, or a kind of "mix", where two functions are mixed in space or time. The purpose of "rolling" (flipping) is to impose a constraint that specifies what to look at when "integrating" [17]. In the case of signal analysis, it specifies which specific point in time to "accumulate" before and after, and in the case of spatial analysis, it specifies where the peripheral processing is to accumulate.

Standard Convolution

Think of it in terms of functions (or mappings, transformations).The convolution process is the process of linear transformation mapping into new values at each position of the image. The convolution kernel is taken as the weight, its vector is denoted as w, and the pixel vector at the corresponding position of the image is denoted as x [18]. Then the result of the convolution of this position is $y = w^T x + b$, that is, the inner product of the vector plus bias, and x is transformed into y. From this point of view, multi-layer convolution is a layer-by-layer mapping to form a complex function as a whole [19, 20]. The training process is to learn the weight required by each local mapping, and the training process can be regarded as the process of function fitting [21]. Think of it in terms of template matching. Convolution and correlation can be equivalent in calculation [22]. Correlation operation is usually used for template matching, that is, the convolution kernel defines a certain pattern. The convolution (correlation) operation is to calculate the degree of similarity between each location and the pattern, or the number of components of the pattern at each location [23]. For example, when the template was matched on an image layer with various animals, the dog's head position had the largest response after the response graph was applied. Of course, template matching can also be carried out at the feature level, and the hidden layer in the convolutional neural network can be regarded as template matching at the feature level. At this time, each element in the response graph represents the degree of similarity between the current position and the mode. Looking at the response graph alone, we can't actually see anything. It can be imagined that every position has a "dog head", the brighter the place is, the more like a "dog head". If given a template, the image can even be restored through de convolution [24, 25]. What we hope is to find the desired pattern in the image. If a nonlinear function is used to clear the part that does not look like the "dog's head" at all in the response graph, and keep the part that looks like the "dog's head", and then restore the image, and find that there is only one "dog's head" in the restored graph, it will be better [26]. Because we are clear about the image of the pattern, and reduce the interference of other information [27].

Convolution can extract features, and the above mentioned "dog's head" template, so what's the problem if we evaluate convolution as a "dog's head" template. It will lack flexibility, or generalization ability, because the dog's status is varied, if the convolution kernel is directly defined as "rigid", the dog or another dog will not recognize. Then, in order to adapt to the diversity of targets, the convolution kernel needs to be designed accordingly. Sobel operator, for example, to convolution of the image, obtain the image edge response figure, when we see the response, the response of the each position to know the picture represents the position in the original image has a similar to the edge of the sobel operator. The information is compressed, the response actually represents a number in the figure of this position has a corresponding strength of sobel edge model, we sampled with convolution characteristics. Artificial can define edge such simple convolution kernels to describe simple patterns, but more complex patterns to do, like a human face, cats, dogs and so on. Although every dog is different, but we even have never seen a dog, when you see the one will know it is a dog, so this group for the dog there must be some common patterns, let people can be identified, but the problem is how to define the model. We know that "rigid" definition of a dog template is not good, because it lacks generalization ability, so through multi-layer convolution, simple patterns are combined into complex patterns, through this flexible combination to ensure adequate expression ability and generalization ability. The features learned from the shallow layer are simple edges, corners, textures, geometric shapes, surfaces, etc., while the features learned from the deep layer are more complex and abstract, such as dogs, faces, keyboards, etc.

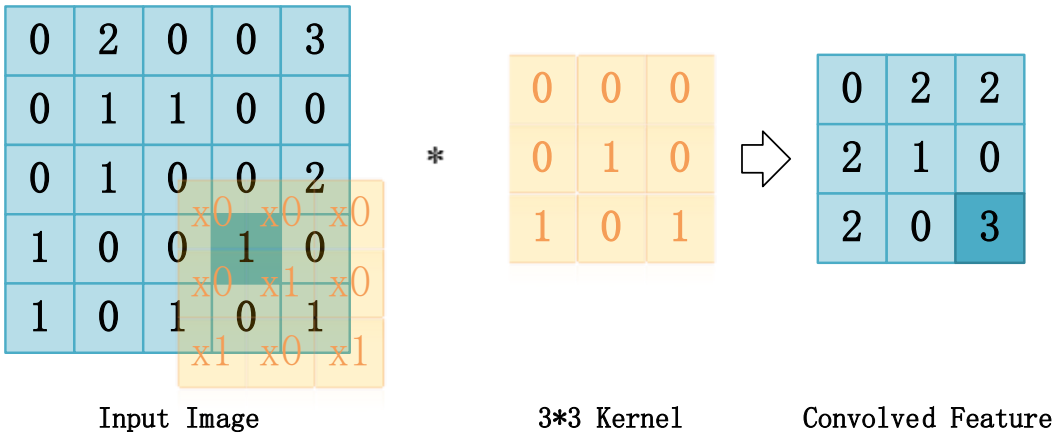

Figure 2. Standard Convolution Process

Here the kernel will convolve over each cell position of the input image and eventually we will get a convoluted image. For every single point in it, you can take the point and the 3x3 points around it out and do the convolution. For instance, for the lower right (row 4, column 4) point, the local convolution of the 5x5 image and the 3x3 matrix could be computed as shown in Figure 2.

3.2. Pooling

After the features are obtained through the convolutional layer, the next step is to use these features to integrate and classify [28, 29]. If all the features extracted through convolution are taken as the input of the classifier, it will face a huge amount of computation. In the convolutional neural network, we often encounter pooling operations [30], and the pooling layer is usually behind the convolutional layer. By pooling, the output feature vectors of the convolutional layer can be reduced, and the calculation amount can be reduced while the results can be improved, so that overfitting is not easy to occur [31, 32]. The reason why it is possible to do this by reducing dimensions is that images have a "static" nature. This means that features that are useful in one area of the image are more likely to be useful in another. Therefore, in order to describe large images, a natural idea is to aggregate statistics of features at different locations. Pooling refers to the operation which integrates each part of the input and then outputs a feature map with reduced size. For example, one can calculate the average or maximum value of a particular feature on a region of the image to represent the feature of that region [33].

Common Pooling methods include Max Pooling and Average Pooling. Pooling functions are used to further process the feature mapping results obtained from the convolution operation. Pooling will statistically summarize the eigenvalues of appointed location and its adjacent location in the plane. And take the summary result as the value of this appointed location in the plane. The use of pooling will not change the depth of the data matrix, but will only reduce the height and width to achieve the purpose of dimension reduction. The role of pooling: Suppress noise and reduce information redundancy; Scale invariance and rotation invariance of the model can be improved; Reduce the calculation amount of model; Prevent overfitting.

Max Pooling

Max Pooling refers to taking the point with the maximum value in the local receiving field. The convolutional layer parameter error causes the deviation of the estimated mean value, which leads to the error of feature extraction. Max Pooling can reduce this error and retain more texture information.

Figure 3 indicates an instance of max pooling utilizing a kernel of 2x2 with a stride of 2. After pooling, the maximum value in the 2x2 kernel will be left in a rectified feature map. Eventually, the 4x4 input image is shrunk to 2x2.

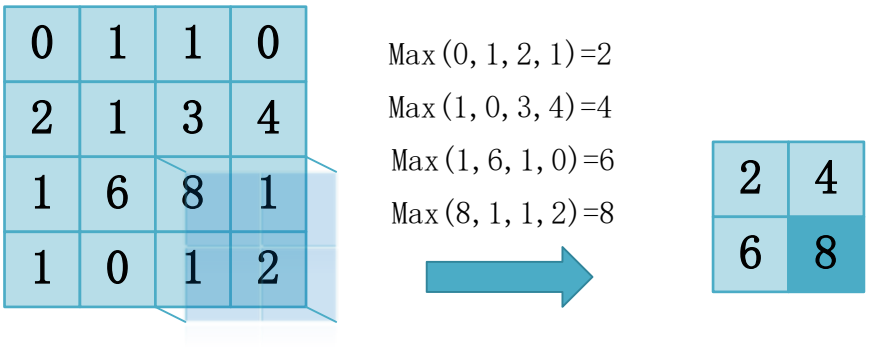

Figure 3. Max Pooling Process

Average Pooling

Average Pooling computes the average value for the position and its adjacent matrix regions and takes this value as the value for the position. The variance of the estimated value increases due to the limited size of the neighborhood, which leads to the error of feature extraction. Average Pooling can reduce the error and retain more background information of images.

Figure 4 indicates an instance of average pooling utilizing a kernel of 2x2 with a stride of 2. After pooling, the average value in the 2x2 kernel will be calculated and reserved in a rectified feature map. In this way, the 4x4 input is compressed to 2x2.

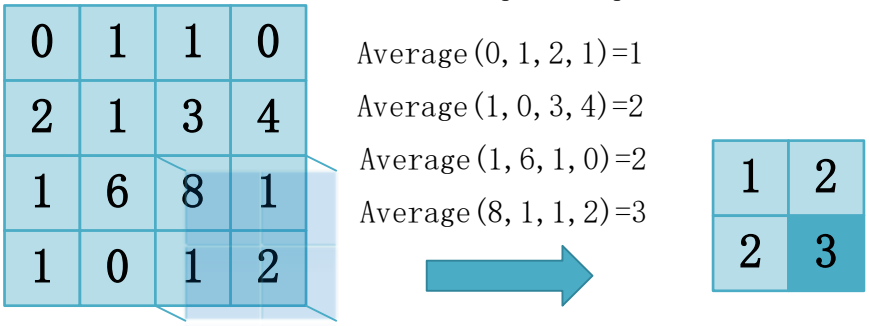

Figure 4. Average Pooling Process

3.3. Batch Normalization

Batch Normalization is to normalize each batch of data [34], indeed, for a batch of data $\{x_1, x_2, \ldots, x_n\}$, note that this data can be either the input or the output of a layer in the middle of the network. The first three steps of BN are as follows:

$$\boldsymbol{\mu} \leftarrow \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x_i} \tag{2}$$

$$\boldsymbol{\sigma^2} = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{x_i} - \boldsymbol{\mu}) \tag{3}$$

$$\widehat{\boldsymbol{x_i}} \leftarrow \frac{\boldsymbol{x_i}-\boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma^2}+\boldsymbol{\varepsilon}}} \tag{4}$$

As shown in the above formula, one layer has m dimensional input: $x = (x_1 \cdots x_m)$. μ is the average of the values of x. σ^2 is the variance of x. And ε is a small constant that keeps the denominator from being zero. So this is a standard data minus mean divided by variance normalization process. The effect of this normalization is shown in Figure 5.

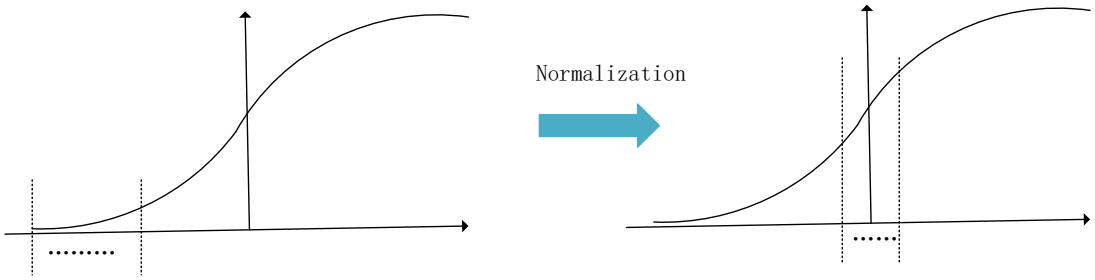

Figure 5. Normalization Process

On the left side of the Figure 5 is the input data without any processing, and the curve is the curve of the activation function, such as Sigmoid. If the data is in a very small gradient area as shown, then the learning rate will be slow or even stagnant for a long time. After subtracting the mean and dividing by the variance, the data is moved to the center, as shown on the right of the Figure 5 [35, 36]. For most activation functions, the gradient in this region is the largest or has a gradient (such as ReLU), which can be seen as a way to counter the gradient disappearance. If this is done for each layer of data, the data distribution is always in the sensitive area with the change of input, which is equivalent to no need to consider the change of data distribution. In this way, the training efficiency is much higher. But that's not the end of the problem, because subtracting the mean and dividing the variance is not necessarily the best distribution [37]. For example, the data itself is very asymmetric, or the activation function may not be the best effect of the data with the variance of 1, such as the Sigmoid activation function, the gradient between -1 and 1 does not change very much, then the function of nonlinear transformation may not be well reflected. So, after the first three steps, we'll add the last step to make it the real Batch Normalization.

$$\widehat{y_t} \leftarrow \gamma \widehat{x_t} + \beta \tag{5}$$

Among them, γ and β are two parameters that need to be learned, so in fact, the essence of BN is to change the variance size and the position of the mean value by using optimization. It's called Batch Normalization because it counts variance and mean, and these values are calculated on each batch of data. When training the model, the mean value and variance of data distribution should be as close as possible to the distribution of all data. Therefore, the mean value and variance of a large number of data should be recorded in the training process to obtain the expected value of the mean value and variance of the whole training sample, which will be used as the final mean value and variance after the training.

Since the features of the convolutional neural network correspond to a whole characteristic response graph, BN is also made in response graph units rather than in various dimensions. For example, for a certain layer, the batch size is m and the response graph size is w*h, then the amount of data to do BN is m*w*h.

3.4. Rectified Linear Unit

As is shown in Figure 6, the part less than 0 is directly set to 0, and the part greater than 0 as the input of ReLU. In this way, the nonlinear transformation is realized, and the gradient of the part greater than 0 is 1. The gradient of the activation function is always 1 for the information that needs to be transmitted from the input all the time, and it will not become smaller even if multiplied continuously, thus solving the problem of the gradient disappearing. ReLU is a powerful tool to help train for rapid convergence.

$$\sigma_{ReLU}(x) = \max(0, x) \tag{6}$$

According to the definition of ReLU, information can only be transmitted in the region (forward and backward) where the input of ReLU is greater than 0, which brings another advantage that is sparsity. The sparsity is not only helpful to improve the performance of the network, but also from the perspective of neuroscience, the activation rate of neurons is very low, which is also a kind of bionic simulation.

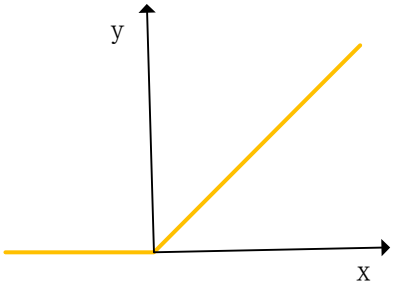

Figure 6. Rectified Linear Unit

3.5. ResNet-101

Another aspect of the training network is the problem of degradation. To put it simply, as the number of layers deepens to a certain extent, the deeper the network is, the worse the effect will be. Moreover, it is not because of the overfitting caused by the deeper network, nor necessarily because of the attenuation of gradient propagation, because there are many effective methods to avoid this problem. In order to solve the degradation problem, the residual module is proposed in ResNet-101.The general idea is that since unit mapping does not work in the framework of gradient descent, the input is directly exported to the output, "forced" as part of the unit mapping. And the learnable network as another part, this is the module of residual learning.

As we can see in Figure 7, the data travels through two straight lines, one through three convolution layers like a general network and then to the output, and the other is a direct connection route to achieve the unit mapping, which is called a shortcut. After doing this, if the parameters in the previous layer have reached a good level, as mentioned earlier, then the information entered in the basic building module is saved by shortcut to some extent.

This module addresses the degradation problem well and takes the number of network layers that can be effectively trained one step further. As for why it is effective, if the input-output relationship of a module in the network is regarded as y=H(x), then solving H(x) directly through the gradient method will face the mentioned degradation problem. So through this shortcut method, the optimization target of the variable parameter part is no longer H(x), as shown in the dashed box of Figure 7. If F(x) is used to represent the part to be optimized, then H(x)=F(x)+x, that is, F(x)=H(x)-x. Since y=x is equivalent to the observed value in the assumption of the unit mapping, F(x) corresponds to the residual, so it is called the residual network.ResNet-101 provides two options, namely identity mapping and residual mapping. If the network has reached the optimum, the residual mapping will be pushed to 0 and only identity mapping will be left if the network continues to deepen. In this way, the network is always in the optimal state theoretically, and the performance of the network will not decrease with the increase of depth.

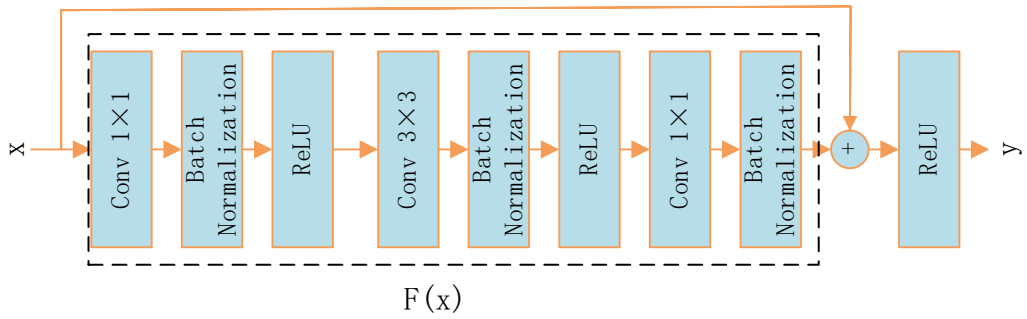

Figure 7. Residual Block

As shown in the Figure 7, in order to reduce calculation consumption, dimensionality reduction was first done through 1*1 convolution, then normal 3*3 convolution layer, and finally 1*1 convolution to match the dimension with shortcut.

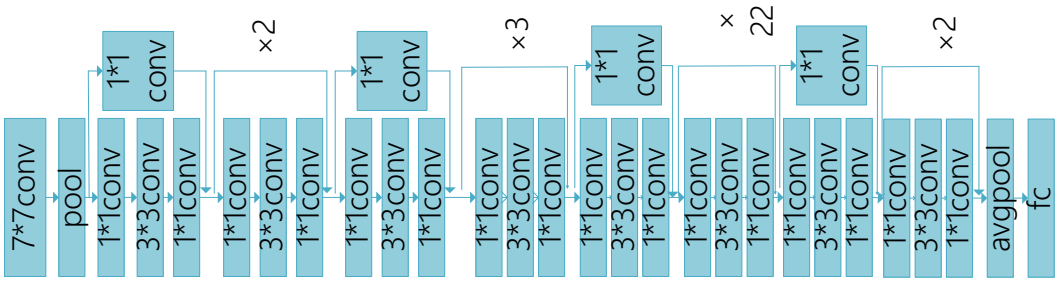

Figure 8. ResNet-101 Network Structure

In Figure 8,ResNet-101 first passes through the 7*7 convolutional layer, then through the 3*3 maximum pooled sub-sampling layer, then through a series of residual structures, and finally through the average pooled sub-sampling layer and the full connection layer.

This experiment is combined with transfer learning. In the Figure 9, Conv1, Conv2...ConvN refers to N convolutional layers used to extract features at different levels of an image. Among them, the shallow layer Conv1, Conv2 and so on to extract the shallow layer features of the image, such as: corner, texture, brightness and so on; Deep ConvN-1 and ConvN are used to extract more abstract features of images, such as eyes, nose, mouth and so on. The Dense layer refers to the whole connection layer, which is used to combine the features that have been learned, so as to learn how to distinguish various expressions.

For this classical classification network structure, there is one characteristic: the characteristics of shallow network recognition are universal. Because of this generality, we don't have to spend a lot of time and resources retraining these shallow features. Instead, we can take the previously trained model and fine-tune it to train the model parameters for specific tasks.

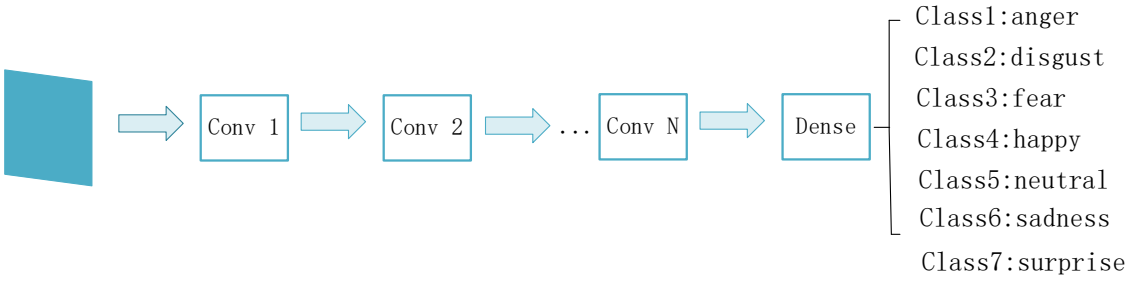

Figure 9. Classification Network Structure

3.6. 10-fold cross validation

N-fold cross validation has two purposes: model evaluation and model selection. N-fold crossing is a strategy for partitioning data set. It can avoid the limitations and particularity of fixed partitioned data set, and this advantage is more obvious in small-scale data set. If this strategy is used to divide training set and test set, model evaluation can be carried out. When this strategy is used to divide the training set and the validation set, model selection can be made.

The most important function of cross-validation is model selection, which can also be called hyperparameter selection. In this case, the data set needs to be divided into three parts: train set, validation set and test set, and the division of train set and validation set adopts the way of N-fold cross. The validation set is used to check the training condition of the model in the training process, so as to determine the appropriate hyperparameters. The test set is to test the generalization ability of the model after the training. The specific process is as follows: Firstly, a variety of model choices (hyperparameter selection) are validated on the train set and validation set, and the model with the minimum average error (hyperparameter) is selected. After selecting the appropriate model (hyperparameter), the train set and validation set can be combined, and the model can be trained again on the above to get the final model, and then the test set can be used to test its generalization ability.

The advantage of cross-validation is to avoid problems caused by unreasonable data set partition, such as over-fitting of the model on the training set, which may not be caused by the model, but caused by unreasonable data set partition. This situation is easy to occur when training models with small data sets, so it is more advantageous to evaluate models with cross-validation method on small data sets.

During the experiment, we use 10-fold cross validation. Each group contained 10 pictures of seven emotions: happy, sadness, fear, anger, surprise, disgust and neutral. Eight of these groups are used for training, one for validation, and the remaining one for testing.

For the sensitivity and overall accuracy (OA) of the network after the implement of r=10,g=10, we can obtain the following formula to define:

$$E(t) = \frac{M_{tt}(r=10,g=10)}{\sum_{i=1}^{7} M_{ti}(r=10,g=10)} \quad (7)$$

$$OA = \frac{\sum_{i=1}^{7} M_{ii}(r=10,g=10)}{\sum_{i=1}^{7}\sum_{j=1}^{7} M_{ij}(r=10,g=10)} \quad (8)$$

M is the confusion matrix, r is the number of iteratio ns, and g is the number of groups. M_{ij} represents the conf usion matrix representation of class i recognized as class j . In order to improve the accuracy of the experiment and r educe the error, we will implement 10 runs and summariz e the confusion matrix (CM). Here, $E(t)$ is the sensitivity of class $t(t \in [1,7], t \in N^{+})$, which means the tth element on the diagonal of $M(r = 10, g = 10)$ divided by the sum

of the tth row. OA is the overall precision, which means t ake the sum of the diagonal elements of $M(r = 10, g = 10)$ divided by the sum of $M(r = 10, g = 10)$.

4. Experiment Result and Discussions

4.1. Statistical Analysis

Table 1 is the statistical analysis of sensitivity of various types given by our system. C1-C7 represents the experimental data of the seven expressions of anger, disgust, fear, happy, neutral, sadness and surprise. Facial expression is the result of one or more movements or states of facial muscles. These movements express the individual's emotional state towards the observer. Table 1 shows the sensitivity analysis of 7 emotion classes running for 10 times. According to the data in Table 1 and Figure 10, the sensitivities of each expression were as follows: 95.30± 4.00%, 96.10± 1.66%, 97.50± 1.65%, 95.80± 1.69%, 96.50± 1.08%, 95.80± 1.87%, and 97.00± 1.05%. From the above data we can get: the expression of fear (C3) are the most sensitive and recognizable, followed by expressions of surprise (C7) and the third is the expression of neutral (C5). According to Table 2 and Figure 11, the overall average accuracy of the system after 10 times of operation is 96.29± 0.78%.

Table 1. Statistical analysis on the sensitivities of each class

Run	C1	C2	C3	C4	C5	C6	C7
1	96.00	98.00	98.00	96.00	98.00	93.00	98.00
2	96.00	93.00	99.00	94.00	96.00	96.00	98.00
3	97.00	96.00	94.00	93.00	98.00	96.00	97.00
4	92.00	97.00	96.00	96.00	97.00	96.00	97.00
5	98.00	94.00	96.00	98.00	96.00	98.00	96.00
6	97.00	96.00	98.00	96.00	97.00	95.00	97.00
7	98.00	97.00	99.00	98.00	95.00	98.00	95.00
8	85.00	95.00	98.00	97.00	96.00	93.00	98.00
9	97.00	97.00	98.00	94.00	97.00	95.00	96.00
10	97.00	98.00	99.00	96.00	95.00	98.00	98.00
Mean+SD	95.30± 4.00	96.10± 1.66	97.50± 1.65	95.80± 1.69	96.50± 1.08	95.80± 1.87	97.00± 1.05

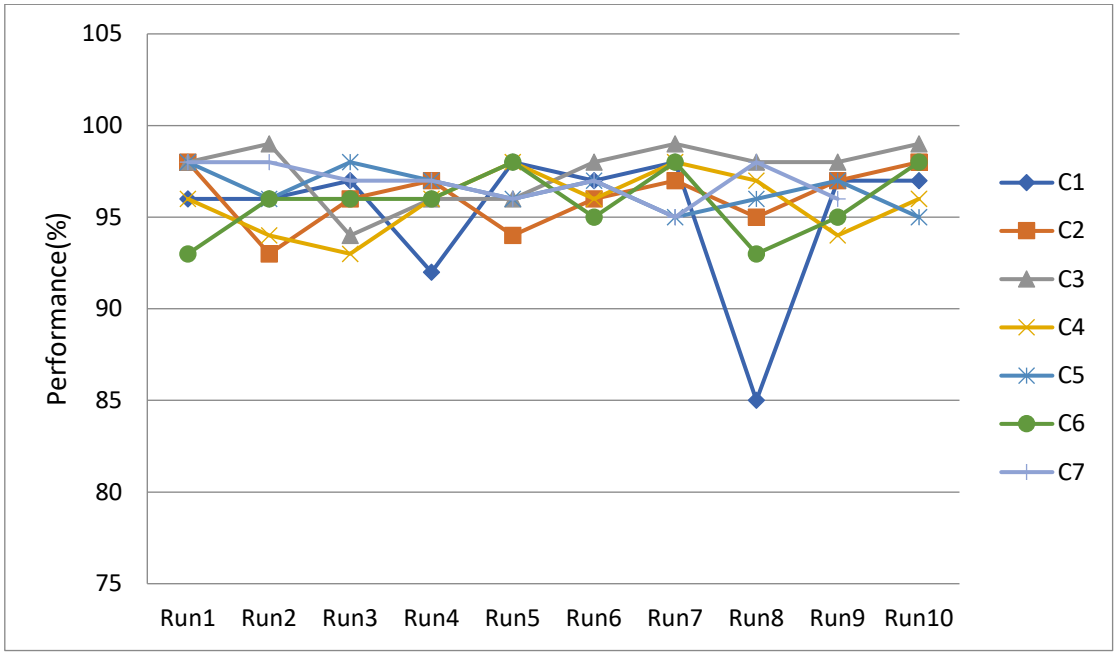

Figure 10. The trend of the sensitivities of each class

Table 2. Statistical analysis on the overall accuracies

Run	OA
1	96.71
2	96.00
3	95.86
4	95.86
5	96.57
6	96.57
7	97.14
8	94.57
9	96.29
10	97.29
Mean+SD	96.29± 0.78

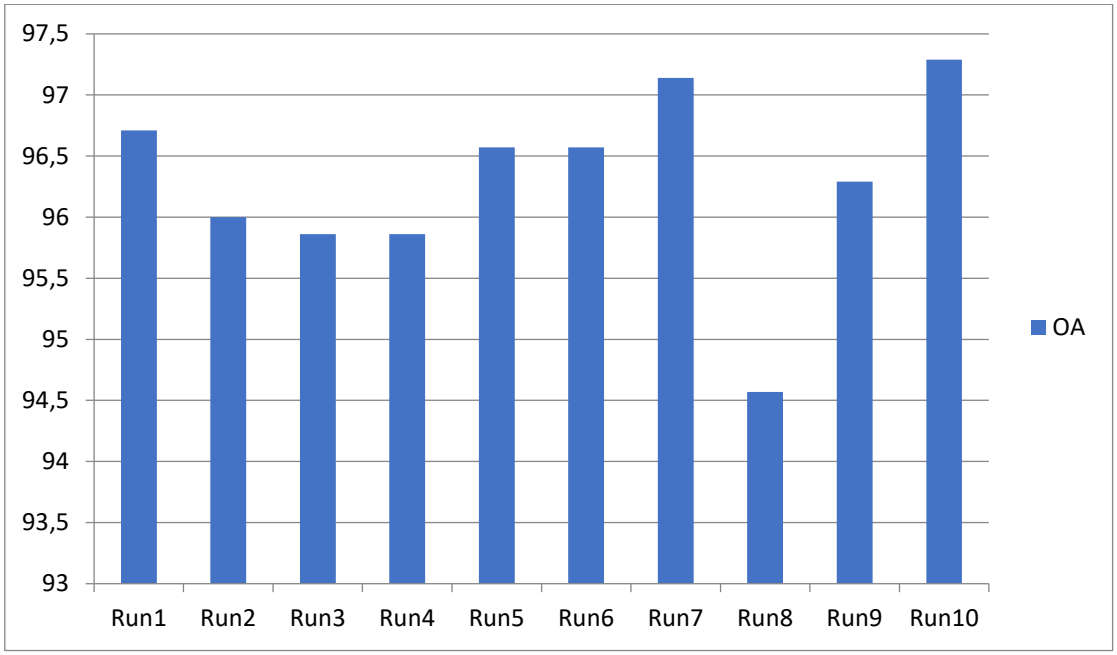

Figure 11. Overall accuracy comparison

4.2. Comparison with State-of-the-art Approaches

In this experiment, the OA of "ResNet-101" combined with transfer learning method was compared with that of PCA + SVM [3], HOS [4], CSO [7] and BBO [8]. The results were shown in Table 3 and Figure 12: the OA of PCA + SVM [3] was 89.14±2.91%, the OA of HOS [4] was 83.43±2.15%, the OA of CSO [7] was 89.49±0.76%, and the OA of BBO [8] was 93.79±1.24%.We can clearly see that the accuracy of "ResNet-101" combined with transfer learning method is the highest (96.29±0.78%), followed by BBO [8], CSO [7], PCA + SVM [3]. It can be seen from Table 3 that the "ResNet-101" method obtains the highest OA mainly depends on: the existence of ResNet-101 residual module solves the problem of network degradation; In the process of deep learning exploration, it is found that the expansion of depth is far better than the expansion of breadth. In the deep learning task, the depth of the network has a great impact on the effect of the final classification and recognition, so the deeper the network is designed, the better the effect is generally. The theory holds that different neurons in the same layer learn different features, and the deeper the neurons are, the more abstract the learning features are. Combined with convolution, ResNet-101 network is deep, so it has good feature extraction ability and good training ability. ResNet's superiority is that it solves the network degradation problem and gradient vanishing problem of deep network to a large extent. Using the residual network structure h(x)=F(x)+x instead of h(x)=x without a shortcut connection, it is much easier to learn F(x)=0 when updating the parameters of the redundant layer than it is to learn h(x)=x. And the structure of Shortcut connection also ensures that when the parameters are updated by backpropagation, it is difficult for the gradient to be 0 and the gradient will not disappear. The second best method is the BBO [8] algorithm, which is developed on the basis of genetic algorithm and particle swarm optimization. It is suitable for solving high-dimensional, multi-objective optimization problems. The third best method is CSO [7], a kind of swarm intelligence algorithm. It is a method to solve complex optimization problems by combining the two behaviors of cat searching and tracking.

In the following work, we will continue to explore ways to improve the accuracy of facial expression recognition. We will try more methods based on the deep residual network.

Table 3. Comparison with State-of-the-art methods

Approach	OA
PCA + SVM [3]	89.14±2.91
HOS [4]	83.43±2.15
CSO [7]	89.49±0.76
BBO [8]	93.79± 1.24
Ours	96.29± 0.78

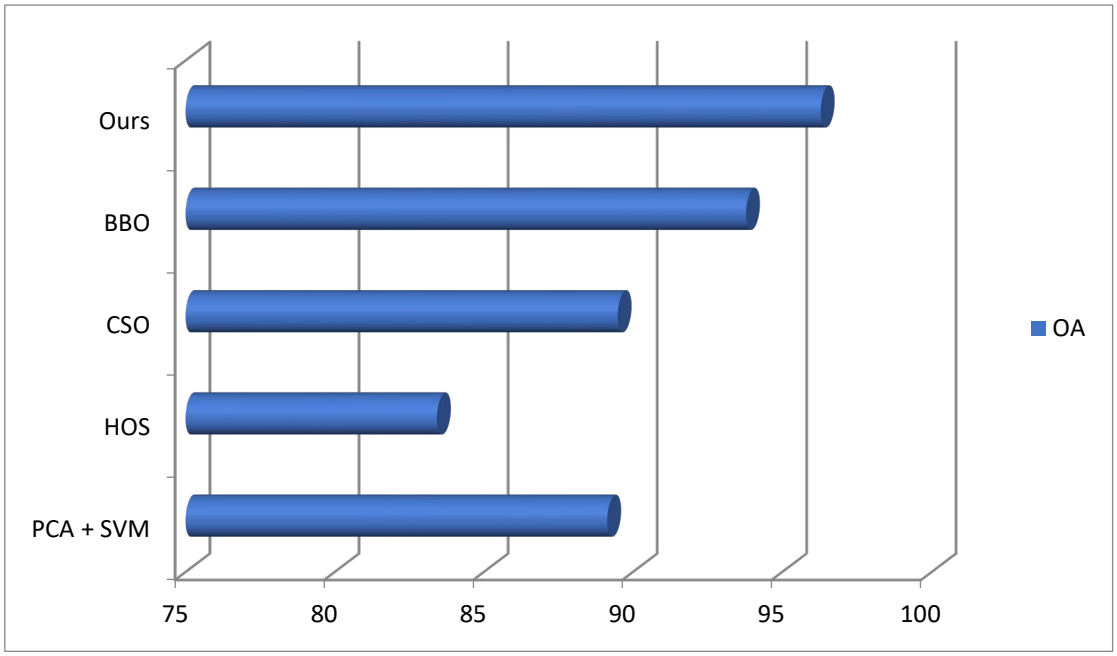

Figure 12. Comparison with each method

5. Conclusion

In this paper, we propose an improved facial emotion recognition system. We train the multi-layer neural network with transfer learning, and use the relatively deep network ResNet-101 for feature extraction. The trained deep neural network can abstract the data features layer by layer, and finally extract the features needed to complete the task. Then we integrate these features through the full connection layer, and finally use a classifier to complete the final task. Through the analysis of experimental results, the facial emotion recognition system has achieved a good recognition effect.

Through the construction of ResNet, the deeper the network is, the better the extraction of high-level abstract features and the better the network performance, and there is no need to worry about the degradation problem with the deepening of the network. Although the network can continue to grow deeper, it is sometimes necessary to double the number of network layers for a small increase in accuracy, which reduces feature reuse and slows down training speed. In the next work, we'll try to start from the width point of view and increase the width to improve performance.

In the future research, we will continue to focus on facial expression recognition and try to collect more facial expression images to optimize and propose a better algorithm to train the multi-layer neural network. We will also try better methods for facial expression recognition to increase the accuracy of recognition and improve the performance of multi-layer neural network.

References

[1] A. R. Hazourli, A. Djeghri, H. Salam, and A. Othmani. (2021). Multi-facial patches aggregation network for facial expression recognition and facial regions contributions to emotion display. *Multimedia Tools and Applications* [Article; Early Access]. doi: 10.1007/s11042-020-10332-7

[2] C. Faniku, W. Kong, L. He, M. Zhang, G. Lilly, and J. P. Pepper, "Hedgehog signaling promotes endoneurial fibroblast migration and Vegf-A expression following facial nerve injury," *Brain Research,* vol. 1751, p. 10, Article ID: 147204, Jan, 2021.

[3] D. Drume and A. S. Jalal, "A Multi-level Classification Approach for Facial Emotion Recognition," in *International*

Conference on Computational Intelligence And Computing Research, Coimbatore, INDIA, 2012, pp. 288-292.
[4] H. Ali, M. Hariharan, S. Yaacob, and A. H. Adom, "Facial Emotion Recognition Based on Higher-Order Spectra Using Support Vector Machines," *Journal Of Medical Imaging And Health Informatics,* vol. 5, pp. 1272-1277, Oct, 2015.
[5] F. Y. Shih and C. F. Chuang, "Automatic extraction of head and face boundaries and facial features," *Information Sciences,* vol. 158, pp. 117-130, Jan, 2004.
[6] F. Evans, "Haar Wavelet Transform Based Facial Emotion Recognition," *Advances in Computer Science Research,* vol. 61, pp. 342-346, 2017/03, 2017.
[7] W. Yang, "Facial Emotion Recognition via Discrete Wavelet Transform , Principal Component Analysis, and Cat Swarm Optimization," *Lecture Notes in Computer Science,* vol. 10559, pp. 203-214, 2017.
[8] X. Li, "Facial emotion recognition via stationary wavelet entropy and Biogeography-based optimization," *EAI Endorsed Transactions on e-Learning,* vol. 6, Article ID: e4, 2020.
[9] H. M. Lu, "Facial Emotion Recognition Based on Biorthogonal Wavelet Entropy, Fuzzy Support Vector Machine, and Stratified Cross Validation," *IEEE Access,* vol. 4, pp. 8375-8385, 2016.
[10] Y.-D. Lv, "Alcoholism detection by data augmentation and convolutional neural network with stochastic pooling," *Journal of Medical Systems,* vol. 42, Article ID: 2, 2018.
[11] C. Tang, "Twelve-layer deep convolutional neural network with stochastic pooling for tea category classification on GPU platform," *Multimedia Tools and Applications,* vol. 77, pp. 22821-22839, 2018.
[12] C. Pan, "Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling," *Journal of Computational Science,* vol. 27, pp. 57-68, 2018.
[13] A. M. Hamer, D. M. Simms, and T. W. Waine, "Replacing human interpretation of agricultural land in Afghanistan with a deep convolutional neural network," *International Journal of Remote Sensing,* vol. 42, pp. 3017-3038, Apr, 2021.
[14] C. Huang, "Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling," *Frontiers in Neuroscience,* vol. 12, Article ID: 818, 2018-November-08, 2018.
[15] C. Pan, "Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU," *Journal of Computational Science,* vol. 28, pp. 1-10, 2018/09/01/, 2018.
[16] G. Zhao, "Polarimetric synthetic aperture radar image segmentation by convolutional neural network using graphical processing units," *Journal of Real-Time Image Processing,* vol. 15, pp. 631-642, 2018.
[17] N. Saxena and R. Balasubramanian, "A pansharpening scheme using spectral graph wavelet transforms and convolutional neural networks," *International Journal of Remote Sensing,* vol. 42, pp. 2898-2919, Apr, 2021.
[18] K. Muhammad, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimedia Tools and Applications,* vol. 78, pp. 3613-3632, 2019.
[19] S. Xie, "Alcoholism Identification Based on an AlexNet Transfer Learning Model," *Frontiers in Psychiatry,* vol. 10, Article ID: 205, 2019-April-11, 2019.
[20] J. Llombart, D. Ribas, A. Miguel, L. Vicente, A. Ortega, and E. Lleida, "Progressive loss functions for speech enhancement with deep neural networks," *Eurasip Journal on Audio Speech and Music Processing,* vol. 2021, p. 16, Article ID: 1, Dec, 2021.
[21] C. Tang, "Cerebral Micro-Bleeding Detection Based on Densely Connected Neural Network," *Frontiers in Neuroscience,* vol. 13, Article ID: 422, 2019-May-17, 2019.
[22] V. V. Govindaraj, "High performance multiple sclerosis classification by data augmentation and AlexNet transfer learning model," *Journal of Medical Imaging and Health Informatics,* vol. 9, pp. 2012-2021, 2019.
[23] M. Sahani and P. K. Dash, "FPGA-Based Deep Convolutional Neural Network of Process Adaptive VMD Data With Online Sequential RVFLN for Power Quality Events Recognition," *IEEE Transactions on Power Electronics,* vol. 36, pp. 4006-4015, Apr, 2021.
[24] S.-H. Wang and J. Sun, "Cerebral micro-bleeding identification based on a nine-layer convolutional neural network with stochastic pooling," *Concurrency and Computation: Practice and Experience,* vol. 32, p. e5130, 2020.
[25] A. K. Sangaiah, "Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization," *Neural Computing and Applications,* vol. 32, pp. 665-680, 2020.
[26] S.-H. Wang, "DenseNet-201-Based Deep Neural Network with Composite Learning Factor and Precomputation for Multiple Sclerosis Classification," *ACM Trans. Multimedia Comput. Commun. Appl.,* vol. 16, p. Article 60, 2020.
[27] G. F. Roberto, A. Lumini, L. A. Neves, and M. Z. do Nascimento, "Fractal Neural Network: A new ensemble of fractal geometry and convolutional neural networks for the classification of histology images," *Expert Systems with Applications,* vol. 166, p. 11, Article ID: 114103, Mar, 2021.
[28] Y.-D. Zhang, "Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation," *Information Fusion,* vol. 64, pp. 149-187, 2020/12/01/, 2020.
[29] X. Wu, "Diagnosis of COVID-19 by Wavelet Renyi Entropy and Three-Segment Biogeography-Based Optimization," *International Journal of Computational Intelligence Systems,* vol. 13, pp. 1332-1344, 2020-09-17T09:29:20.000Z, 2020.
[30] S. Sharma, R. Mehra, and S. Kumar. (2021). Optimised CNN in conjunction with efficient pooling strategy for the multi-classification of breast cancer. *IET Image Processing* [Article; Early Access]. doi: 10.1049/ipr2.12074
[31] S.-H. Wang, "Covid-19 Classification by FGCNet with Deep Feature Fusion from Graph Convolutional Network and Convolutional Neural Network," *Information Fusion,* vol. 67, pp. 208-229, 2020/10/09/, 2021.
[32] S. C. Satapathy, "A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis," *Machine Vision and Applications,* vol. 32, Article ID: 14, 2021.
[33] T. Tuncer, A. Subasi, F. Ertam, and S. Dogan, "A novel spiral pattern and 2D M4 pooling based environmental sound classification method," *Applied Acoustics,* vol. 170, p. 11, Article ID: 107508, Dec, 2020.
[34] B. Olimov, K. Sanjar, S. Din, A. Ahmad, A. Paul, and J. Kim. (2021). FU-Net: fast biomedical image segmentation model based on bottleneck convolution layers. *Multimedia Systems* [Article; Early Access]. 14. doi: 10.1007/s00530-020-00726-w
[35] S.-H. Wang, "COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant

correlation analysis," *Information Fusion,* vol. 68, pp. 131-148, 2021.

[36] D. S. Guttery, "Improved Breast Cancer Classification Through Combining Graph Convolutional Network and Convolutional Neural Network," *Information Processing and Management,* vol. 58, Article ID: 102439, 2021.

[37] E. Rajbhandari, A. Alsadoon, P. W. C. Prasad, I. Seher, T. Q. V. Nguyen, and D. T. H. Pham. (2021). A novel solution of enhanced loss function using deep learning in sleep stage classification: predict and diagnose patients with sleep disorders. *Multimedia Tools and Applications* [Article; Early Access]. 24. doi: 10.1007/s11042-020-10199-8