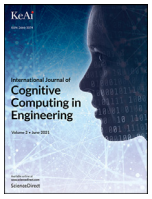
International Journal of Cognitive Computing in Engineering 2 (2021) 57–64

Contents lists available at ScienceDirect

International Journal of Cognitive Computing in Engineering

journal homepage: https://www.keaipublishing.com/en/journals/international-journal-of-cognitive-computing-in-engineering/



Facial expression recognition via ResNet-50

Bin Li [a], Dimas Lima [b,*]

[a] School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, PR China
[b] Department of Electrical Engineering, Federal University of Santa Catarina, Florianópolis, Brazil

ARTICLE INFO

Keywords:
Deep residual network
Facial expression recognition
ResNet-50

ABSTRACT

As one of the most important directions in the field of computer vision, facial emotion recognition plays an important role in people's daily work and life. Human emotion recognition based on facial expressions is of great significance in the application of intelligent human-computer interaction. However, in the current research on facial emotion recognition, there are some problems such as poor generalization ability of network model and low robustness of recognition system. In this content, we propose a method of feature extraction using the deep residual network ResNet-50, which combines convolutional neural network for facial emotion recognition. Through the experimental simulation of the specified data set, it can be proved that this model is superior to the current mainstream facial emotion recognition models in the performance of facial emotion detection.

1. Introduction

With the rapid development of artificial intelligence, people pay more and more attention to the emotional intelligence (Gonzalez-Yubero, Lazaro-Visa & Palomera, 2021) of machines, we are more eager to people's communication with the computer can be like the means of communication between people (Hanafi & Daud, 2021). This can be achieved only if computers can understand people's emotions, and facial expression recognition is an effective way to achieve it. Facial emotion recognition refers to the separation of specific facial states from a given static image or dynamic video sequence, so as to determine the psychological emotions of the object to be recognized. On the other hand, facial emotion is an important non-verbal expression in daily communication. By observing the changes of facial emotion (Furlong et al., 2021; Graumann et al., 2021; Staff et al., 2021), we can better recognize the emotional changes of the other party. Similarly, in the field of computer vision, facial emotion recognition is also an important research direction. Accurate recognition of facial emotions through computer vision recognition system will be conducive to the smooth progress of psychological recognition, human-computer interaction, assisted driving, station security and other work.

Facial emotion recognition is to analyze a given facial expression and use the analysis results to classify specific emotions (Hajarolasvadi & Demirel, Dec, 2020; Rajananda, Zhu & Peters, 2020). Facial emotions can be divided into seven categories: happy, sad, fearful, angry, surprised, disgusted and neutral. The first thing to do for facial expression recognition is to preprocess the collected images, then carry out feature extraction and classification recognition. With the emergence of the convolutional neural network, many scholars tend to use the convolutional neural network to extract image features. Ali, Hariharan, Yaacob and Adom, (2015) proposed the use of support vector machine (SVM) method. Evans (Evans, 2017) presented to use Haar wavelet transform (HWT) method. Phillips (Phillips, 2018) presented a novel method combining stationary wavelet entropy and Jaya algorithm. Through the analysis of the above literatures, it can be found that the facial emotional features extracted by the above methods have the problem that the original emotional information is easy to be lost. In addition, the generalization and robustness of these network models are also poor and the accuracy of facial expression recognition is not high.

For the above problems, we propose an improved facial emotion recognition model. We use ResNet-50 as the network infrastructure. Feature is extracted by convolutional neural network, BN and activation function ReLU are used to improve the convergence ability of the model. Through the simulation experiment below, it can be verified that the recognition performance of the proposed facial emotion recognition method is better than the most advanced method.

The structure of the rest part of this paper is as follows: the second part introduces the research object and data set; The third part introduces the concept and basic principle of the research method. The fourth part introduces the relevant experimental results and analysis. The fifth part summarizes and forecasts the full text.

2. Dataset

In order to make the experimental process easier to achieve and the experimental results more comparative, the data set adopted in this paper is a new data set obtained by using a face model. The data set was

* Corresponding author.
E-mail addresses: libin@home.hpu.edu.cn (B. Li), dimaslima@ieee.org (D. Lima).

https://doi.org/10.1016/j.ijcce.2021.02.002
Received 13 January 2021; Received in revised form 11 February 2021; Accepted 19 February 2021
Available online 23 February 2021

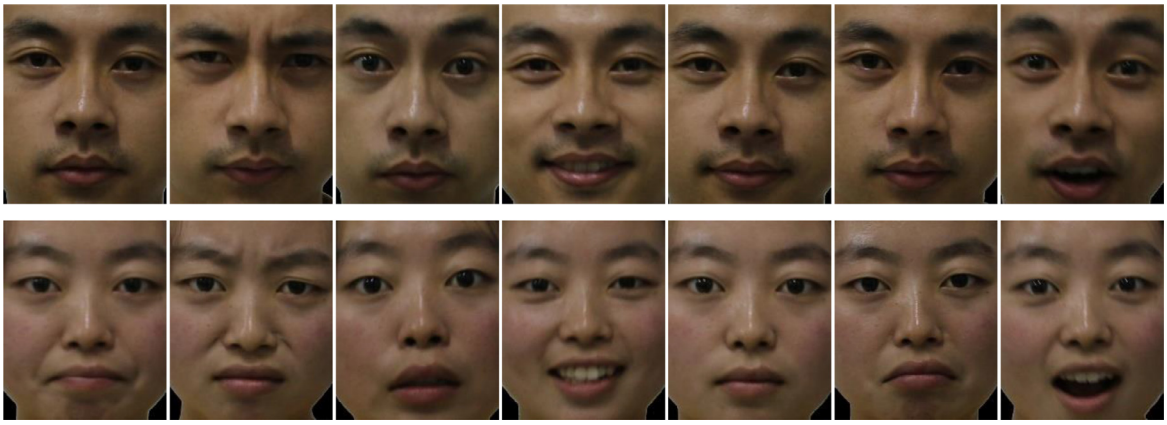


Fig. 1. Samples of our dataset (Lu, 2016).

collected by an experienced photographer who used 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. In the final, we have 700 images in total. Fig. 1 displays samples of our dataset from Ref. (Lu, 2016).

3. Methodology

3.1. Convolution

Convolution (Hasebe & Ueda, 2021) is widely used in the field of image processing, such as filtering, edge detection, image sharpening, etc. (Belinschi, Bercovici & Liu, 2021; Tiwari, Jul-Sep, 2021), all of which are realized by different convolution kernels. In the convolutional neural network, features in the image can be extracted by convolution operation (Sahani & Dash, Apr, 2021). The lower convolutional layer can extract some features such as edges, lines and angles of the image (Guttery, 2021; Satapathy, 2021; Wang, 2021a, 2021b). The higher convolutional layer can learn more complex features from the lower convolutional layer, so as to realize image classification and recognition. As is shown in (1), convolution is a mathematical operator that generates a third function by two functions f and g, and $r(x)$ represents the integral of the overlap length of the product of the overlapping function values of the function f and g by flipping and shifting.

$$r(x) = \int_{-\infty}^{\infty} f(\tau)g(x-\tau)d\tau \tag{1}$$

The physical meaning of convolution is the weighted superposition of one function over another. The output of the system is the result of the superposition of multiple inputs. In the image analysis, $f(x)$ is the original pixel point, $g(x)$ is the action point. All the action points are combined into the convolution kernel. After all the action points on the convolution kernel are applied to the original pixel points in turn, the output of the final convolution is obtained.

3.1.1. Standard convolution

The element of convolution operation in the convolutional layer is called the convolution kernel, and its parameters need to be learned (Bister et al., 2021; Ganguly et al., 2021). The size of the convolution kernel should be smaller than the size of the input image. During the convolution process, each kernel is convolved with the input image to calculate a feature map (Satapathy & Wu, 2020; Wang, 2021a, 2021b; Zhang, Nayak, Zhang & Wang, 2020). In other words, the convolution kernel will slide over the input image and computes the dot product between the input and the convolution kernel at each spatial position. Then, the feature map of each kernel is superimposed along the depth dimension to obtain the output image of the convolutional layer.

In short, convolution is the multiplication and addition of the corresponding elements between the input matrix and the convolution kernel. Finally, the whole input matrix is traversed and the result matrix is obtained. Each kernel is smaller in width and height when compared to the input, each neuron in the activation map is associated only to a small local area of the input image, which means size of each neuron's receptive field is small, equals to size of kernel. From the perspective of template matching, the convolution kernel defines a certain pattern. The convolution 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. The more similar the current location is to the pattern, the stronger the response will be.

Here, the kernel convolves the position of each unit of the input image, and we end up with a convolved image. In Fig. 2, we use a kernel of 3 ×3 with a stride of 1.For each point, you can take this point and convolve it with the 3 by 3 points around it.

3.2. Pooling

After obtaining feature map through convolutional layer, the next step is to integrate and classify these features. Theoretically, all features extracted by convolution can be used as input to a classifier, such as the Softmax classifier (Ashiquzzaman et al., 2020; Satapathy & Zhu, 2020), but doing so has large amount of calculation. At this time, we will use pooling layer to get the feature dimension reduce. We can approximate the pixel point around some pixel points, count the eigenvalues of a certain position and its adjacent positions in the plane, and take the summarized result as the value of this position in the plane. Pooling is a process of abstracting information (Wang, 2020; Wu, 2020; Zhang, 2020). Pooling layer is also called the down-sampling layer, it aims to reduce the size of matrix generated by a convolutional layer. On the one hand, pooling reduces features and parameters, thus simplifying the complexity of convolutional network calculation. Pooling also prevents overfitting to a certain extent, making optimization more convenient. On the other hand, some invariance of features is maintained, such as rotation, translation and contraction, etc. Pooling can increase the invariance of the network against translation, which is very critical to the improvement of the network generalization ability. Pooling makes the model pay more attention to the existence of certain features rather than the specific location of the feature. Another use of pooling is to increase the receptive field, that is, the size of a pixel corresponding to the original image, which improves the capability of the model. There are two main pooling operations, Average Pooling and Max Pooling.

3.2.1. Average pooling

The error of feature extraction mainly comes from two aspects: The first error is that the estimated variance increases due to the limitation of

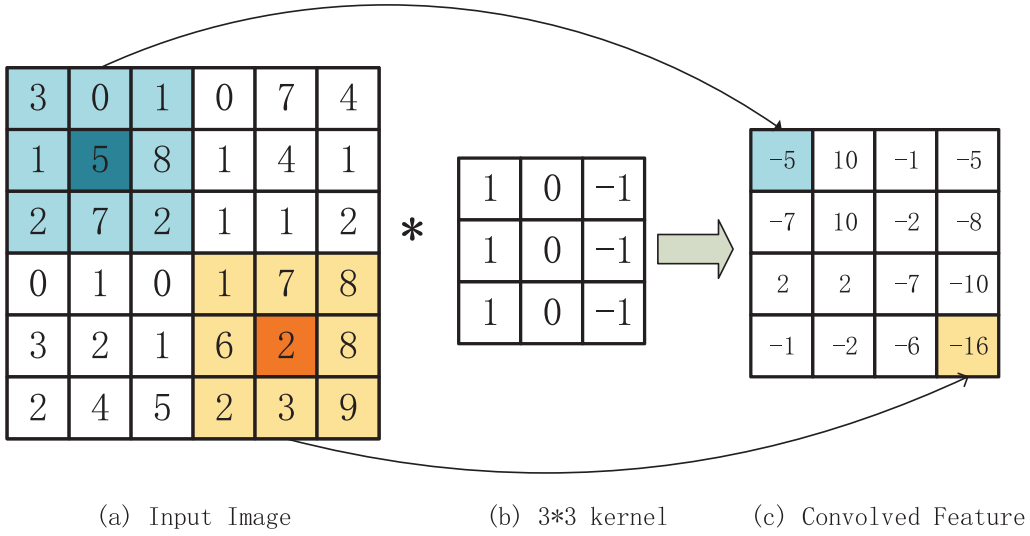


Fig. 2. Standard convolution.

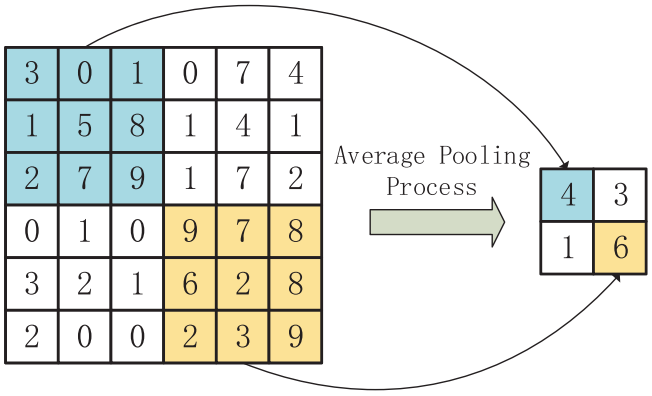


Fig. 3. Average Pooling.

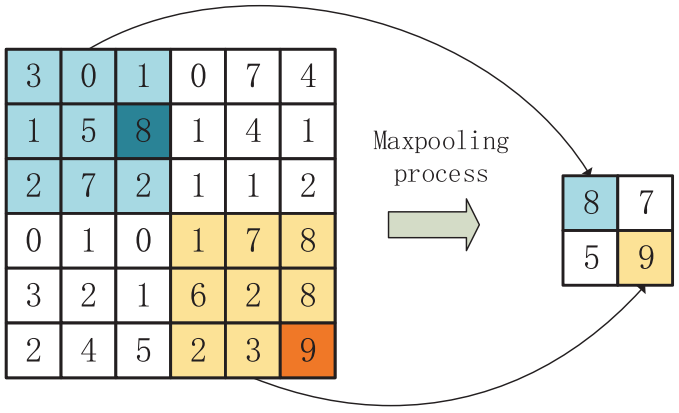


Fig. 4. Max Pooling.

neighborhood size. The second error is that the convolution layer parameter error causes the deviation of the estimated mean value. In general, Average Pooling can reduce the first error and retain more background information of the image. Average Pooling emphasizes the sublayer sampling of the overall feature information, which ensures the integrity of the transmitted information and reduces the parameter dimension. In large models such as Dense Net, Average Pooling is often used to reduce dimensions and facilitate information transfer to the next module to extract features.

In a simple term, Average Pooling means taking the average of a small range. Fig. 3 indicates an instance of average pooling utilizing a kernel of 3×3 with a stride of 3. After pooling, the average value in the 3×3 kernel will be calculated and reserved in a rectified feature map. In this way, the 6×6 input is compressed to 2×2.

3.2.2. Max pooling

Generally speaking, Max Pooling is more efficient and is a common method in image processing. It's kind of like making feature choices. Max Pooling selects features with good classification and recognition performance and has nonlinear characteristics. In addition, Max Pooling can handle the second error mentioned above and retain texture features well. For instance, if the feature we need to find is a cat, then as long as there is a cat in one area of the image, it means that there is a cat in the whole image. Therefore, we should take the maximum matching degree between all areas in the image and the features of the cat, that is, max pooling should be utilized.

In a simple term, Max Pooling means taking the maximum of a small range. Fig. 4 indicates an instance of max pooling utilizing a kernel of 3×3 with a stride of 3. After pooling, the maximum value in the 3×3 kernel will be left in a rectified feature map. Eventually, the 6×6 input image is shrunk to 2×2.

3.3. Batch normalization

Batch Normalization is a training optimization method. In CNN, batch is the number of images that are trained on the network. The reason why input data need normalization is that the learning process of neural network is essentially to learn data distribution (Choi & Jung, Mar, 2020; Garbin, Zhu & Marques, 2020). Once the distribution of training data is different from that of test data, the generalization ability of the network is greatly reduced. On the other hand, once the distribution of each batch of training data is different (batch gradient descent), the network has to learn to adapt to different distributions in each iteration (Govindaraj, 2019; Muhammad, 2019; Sangaiah, 2020). This will greatly reduce the training speed of the network, which is the reason why we need to do a normalized pretreatment of the data. The training of deep network is a complicated process (Nencka et al., 2021; Nishida et al., 2020). As long as the first few layers of the network change slightly, the changes in the next few layers will gradually increase. Therefore, if the distribution of training data keeps changing

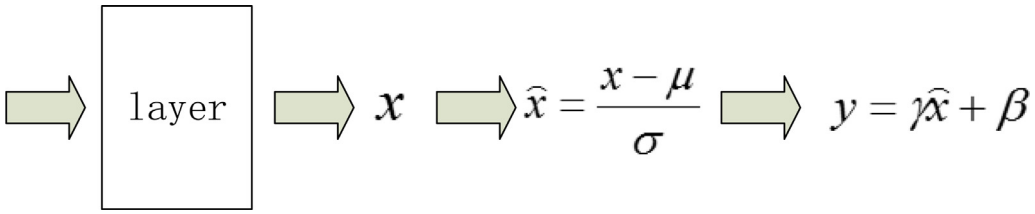


Fig. 5. Batch Normalization Process.

during the training process, the training speed of the network will be affected.

SGD is simple and efficient for training deep network, but it has a problem that it requires us to select parameters artificially (Benbahria, Sebari, Hajji & Smiej, 2021; Polat & Gungen, 2021), such as learning rate, parameter initialization, weight attenuation coefficient, Drop out ratio, etc. The selection of these parameters is so important to the training result that much of our time is wasted on these parameters. So with Batch Normalization, you don't have to adjust the parameters very slowly. In addition, once the neural network is trained, the parameters will be updated. Except the data of the input layer, the input data distribution of each layer of the back network is always changing. During training, the update of the training parameters of the previous layer will lead to the change of the input data distribution of the later layer. Take the second layer of the network as an example: the input of the second layer of the network is calculated by the parameters and input of the first layer. However, the parameters of the first layer are always changing during the whole training process, so it is inevitable to change the distribution of the input data of each subsequent layer. We refer to changes in the data distribution of the network middle layer during training as "Internal Covariate Shift". The Normalization is designed to deal with the change in the middle data distribution during training.

BN, like activation function layer, convolution layer, full connection layer and pooling layer, also belongs to a layer of network. The essential principle of BN is that when each layer of the network is input, another normalization layer is inserted. In other words, the normalization processing is performed first. The Normalization sets a batch of data with a mean of 0 and a variance of 1, and then goes to the next layer of the network.

One layer has d dimensional input:$x = (x^{(1)} \cdots x^{(d)})$, Normalize each dimension:

$$\hat{x}^{(k)} = \frac{x^k - E\left[x^{(k)}\right]}{\sqrt{Var\left[x^{(k)}\right]}} \tag{2}$$

If only the above normalization formula is used to normalize the output data of layer A of the network, and then it is sent to the next layer of network B, then the features learned at the layer of network A will be affected. For example, the network data in the middle of a layer is itself characterized by being distributed on both sides of the Sigmoid activation function, normalized to set its variance to 1, and converted its distribution to the middle of the Sigmoid activation function. This is equivalent to breaking the characteristic distribution learned by this layer of the network. Therefore, the learnable parameters γ,β are introduced to transform and reconstruct:

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \tag{3}$$

$$\gamma^{(k)} = \sqrt{Var\left[x^{(k)}\right]} \tag{4}$$

$$\beta^{(k)} = E\left[x^{(k)}\right] \tag{5}$$

The above formula shows that the learned reconstruction parameters can restore the features of the original layer. Fig. 5 displays the batch normalization process. Among them,μ is mean of x in mini-batch, σ is std of x in mini-batch, γ, β are parameters to be learned, analogous to weights..

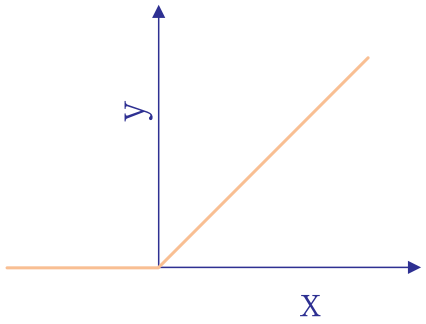


Fig. 6. Rectified Linear Unit.

3.4. Rectified linear unit

The activation function is used to add nonlinear factors, because the linear model is not expressive enough. Assuming that if there are no activation functions, the input of each node at each layer is a linear function of the output at the upper layer, it's easy to verify that no matter how many layers you have in your neural network, the output is a linear combination of the input. This is equivalent to no hidden layer, which means that each layer without activation function is equivalent to matrix multiplication (Huang, 2018; Pan, 2018). Even if you add layers, you're just multiplying matrices. Then the network approximation capability is quite limited. Because of the above reasons, we decided to use nonlinear function as activation function, so that the deep neural network expression ability is more powerful, no longer a linear combination of inputs, but can approximate almost any function. The expression for ReLU is as follows.

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

It is obvious from the expression (6) and the graph Fig. 6 that ReLU is a function to get a maximum value. ReLU is a piecewise function that turns all negative values into 0, but the positive values remain the same, and this operation is called unilateral inhibition. In other words, if the input is negative, it will output zero, so the neuron will not be activated. This means that only a few neurons are activated at the same time, making the network sparse and thus very efficient for computing (Nayak, Das, Dash, Majhi & Majhi, 2020; Olimov et al., 2021). Because of this unilateral inhibition, neurons in the neural network also have sparse activation. This is especially reflected in the deep neural network model such as CNN. When N layers are added to the model, the activation rate of neurons will theoretically decrease by 2 to the NTH power. The advantages of using the ReLU function are as follows: 1) There is no saturation region and no gradient disappearance. 2) Without complex exponential operation, the calculation is simple and the efficiency is improved 3) The actual convergence speed is much faster than Sigmoid/ Tanh. 4) It is more consistent with the biological neural activation mechanism than Sigmoid.

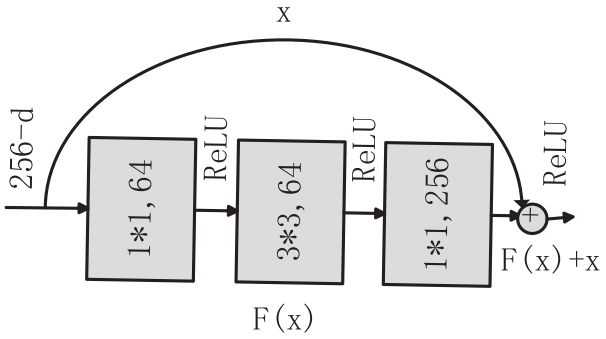


Fig. 7. Residual block.

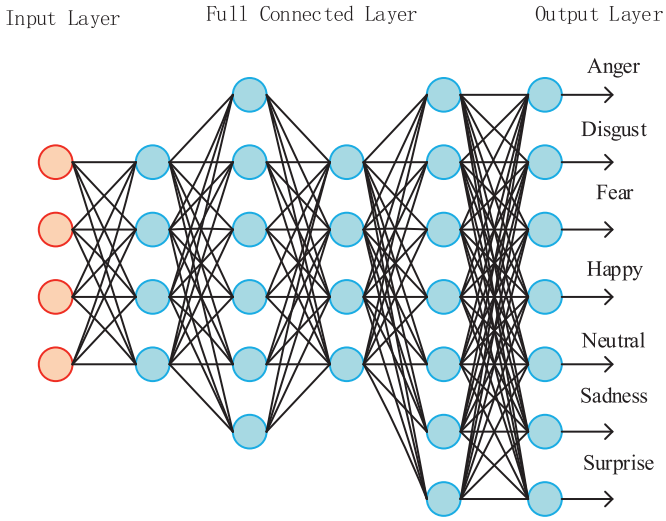


Fig. 9. The structure of FC layer.

3.5. ResNet-50

The first problem with increasing depth is gradient explosion/dissipation, which is due to the fact that as the number of layers increases, the gradient back propagating in the network will become unstable with the multiplications and become very large or very small. One of the problems that often arises is gradient dissipation (Kwon, Pellauer, Parashar & Krishna, 2021; Lee, Kim & Jung, 2021). In order to overcome gradient dissipation, many solutions have been found, such as using Batch Normalization, changing activation function to ReLU, and using Xaiver initialization, etc. It can be said that gradient dissipation has been well solved. Another problem with the network deepening is degradation, that is, the performance of the network is worse as the depth increases. From experience, the depth of the network is crucial to the performance of the model. When the number of network layers is increased, the network can carry out more complex feature pattern extraction, so better results can be obtained theoretically when the model is deeper. However, the experiment found that the deep network was degenerating. With the increase of network depth, the accuracy of the network tends to be saturated or even decreased. There is a decrease in the accuracy of the training set. We can determine that this is not caused by overfitting. Because the accuracy of the training set should be high in the case of overfitting. The residual network in ResNet is designed to solve this problem, and after solving this problem, the depth of the network rises by several orders of magnitude.

ResNet proposed two kinds of mapping: one is identity mapping, referring to the "curved curve" in Fig. 7, and the other residual mapping refers to the part except the "curved curve", so the final output is $y = F(x)+x$. Identity mapping, as the name implies, refers to itself, which is x in the formula, while residual mapping refers to "difference", that is, $y - x$, so residual refers to F(x). At first, ResNet-50 performed convolution operation on the input, followed by 4 residual blocks, and finally performed full connection operation to achieve classification tasks. The network structure of ResNet-50 is shown in Fig. 8, which has 50 Conv2D operations.

Fully connected (FC) layer usually appears at the end of the CNN to summarize the features of the previous layers (Ali, Janabi-Sharifi & Beheshti, 2021). If we take the previous convolution and pooling as the process of feature engineering, local amplification and local feature extraction, the latter FC layer can be thought of as feature weighting.

The structure of the FC layer shown in Fig. 9 is usually a way to quickly learn the nonlinear combinations of advanced attributes generated by the convolutional layer. The FC layer will learn a possible nonlinear function. The basic procedure of learning is as follows. First, the image, which has been converted into a form suitable for multilevel perceptron (Togacar, Ergen & Comert, 2020), is flattened into column vectors and fed back to the feed forward neural network. The flattened data is then applied to each iteration of the training. In this way, the model has the ability to distinguish between the major features in the image and some low-level features and classify them through classification techniques such as Softmax. Here we will output the classification results of the seven expressions (Figs. 10 and 11).

3.6. Measure

In the experimental process, in order to avoid the phenomenon of overfitting, we choose the cross-validation technique of 10 groups. Each group contained 10 images of seven emotions: happy, sadness, fear, anger, surprise, disgust and neutral. Eight of these groups were used for training, one for validation, and the remaining one for testing. For a more concise representation, we introduce the confusion matrix (CM). Therefore, the ideal $H(r = 1, g = 1)$ should be as follows:

$$\mathrm{H}(r=1,g=1)=\begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \tag{7}$$

Here, H is the confusion matrix, r is the number of iterations, and g is the number of groups. The above matrix is the representation of 1 group of ideal confusion matrices in 1 iteration. In particular, H_{ij} represents the confusion matrix representation of class i recognized as class j. To

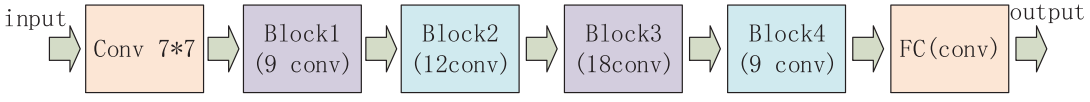


Fig. 8. ResNet-50.

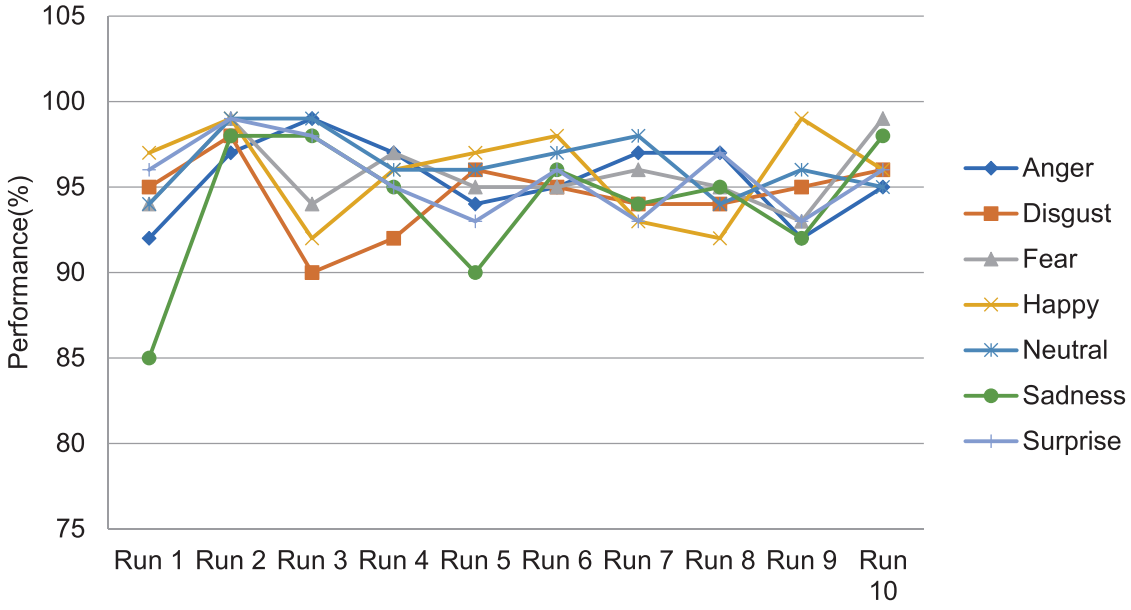


Fig. 10. The trend of the sensitivities of each class.

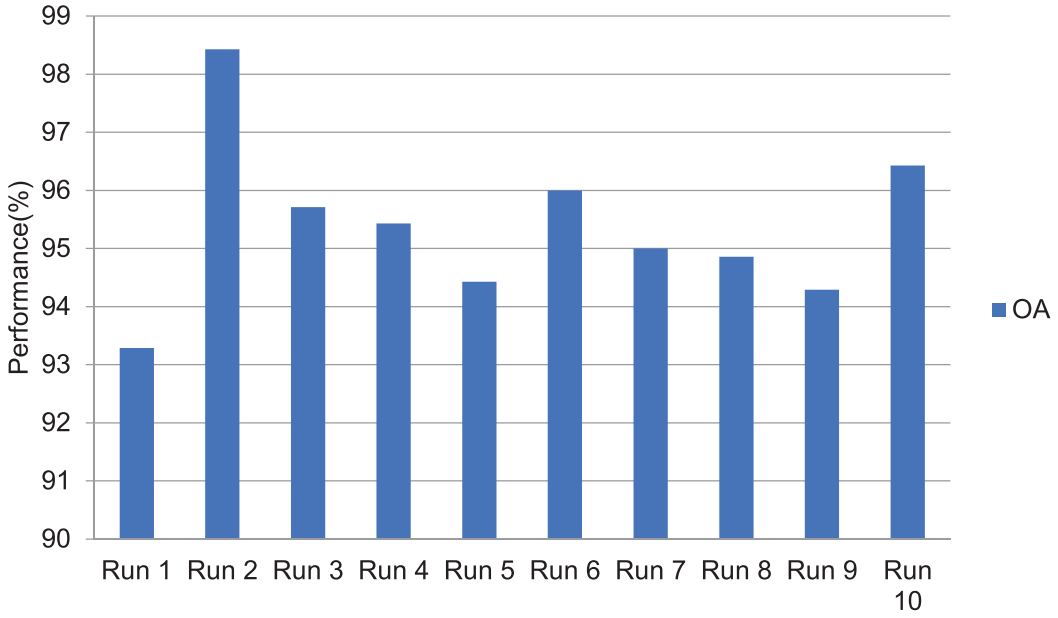


Fig. 11. Overall accuracy comparison.

sum up, the ideal $H(r = 1, g = 10)$ of 10x grouping cross validation is:

$$\mathrm{H}(r = 1, g = 10) = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad (8)$$

Here, the elements on the diagonal of $H(r = 1, g = 10)$ are the structure of matrix summation for the test sets of 10 experiment groups. In general, in order to improve the accuracy of the experiment and reduce the error, we will implement 10 runs and summarize CM. Thus, the ideal $H(r = 10, g = 10)$ can be obtained as:

$$\mathrm{H}(r = 10, g{=}10) = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \quad (9)$$

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:

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

Table 1
Statistical analysis on the sensitivities of each class.

Run	Anger	Disgust	Fear	Happy	Neutral	Sadness	Surprise
1	92.00	95.00	94.00	97.00	94.00	85.00	96.00
2	97.00	98.00	99.00	99.00	99.00	98.00	99.00
3	99.00	90.00	94.00	92.00	99.00	98.00	98.00
4	97.00	92.00	97.00	96.00	96.00	95.00	95.00
5	94.00	96.00	95.00	97.00	96.00	90.00	93.00
6	95.00	95.00	95.00	98.00	97.00	96.00	96.00
7	97.00	94.00	96.00	93.00	98.00	94.00	93.00
8	97.00	94.00	95.00	92.00	94.00	95.00	97.00
9	92.00	95.00	93.00	99.00	96.00	92.00	93.00
10	95.00	96.00	99.00	96.00	95.00	98.00	96.00
Total	95.50± 2.32	94.50± 2.22	95.70± 2.06	95.90± 2.69	96.40± 1.84	94.10± 4.15	95.60± 2.12

Table 2
Statistical analysis on the overall accuracies.

Run	OA
1	93.29
2	98.43
3	95.71
4	95.43
5	94.43
6	96.00
7	95.00
8	94.86
9	94.29
10	96.43
Total	95.39± 1.41

$$OA = \frac{\sum_{i=1}^{7} H_{ii}(r = 10, g = 10)}{\sum_{i=1}^{7}\sum_{j=1}^{7} H_{ij}(r = 10, g = 10)} \tag{11}$$

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

4. Experiment result and discussions

4.1. Statistical analysis

The sensitivity analysis is shown in Table 1 .These data correlate closely with the facial muscles that correspond to the expression, with funnel-shaped lips and nasal wrinkles making their early facial expressions similar, and expressions such as drooping jaws and lifting upper eyelids making expressions based on the same muscle movement characteristics similar.

According to the data in Table 1, the sensitivity analysis of the seven facial expressions running ten times is as follows:95.50±2.32%, 94.50±2.22%, 95.70±2.06%, 95.90±2.69%, 96.40±1.84%, 94.10±4.15%, 95.60±2.12%. From this we can conclude that neutral expressions are the most sensitive and easily recognized, followed by happy and fear. As can be seen from Table 2, the overall average accuracy of the system after 10 runs is 95.39±1.41%.

4.2. Comparison with state-of-the-art approaches

The OA of "ResNet-50″ method used in this experiment was compared with that of the other three methods, HWT (Evans, 2017), CSO (Yang, 2017) and BBO (Li, 2020). The results are shown in Table 3: The OA of HWT (Evans, 2017) is 78.37±1.50%;The OA of CSO (Yang, 2017) is 89.49±0.76%;The OA of BBO (Li, 2020) is 93.79±1.24%.We can clearly see that "ResNet-50″ method has the highest accuracy (95.39±1.41%), followed by BBO (Li, 2020), CSO (Yang, 2017), and HWT (Evans, 2017).

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

Method	OA
HWT (Evans, 2017)	78.37±1.50
CSO (Yang, 2017)	89.49±0.76
BBO (Li, 2020)	93.79±1.24
ResNet-50 (Ours)	95.39±1.41

It can be seen from Table 1 that "ResNet-50″ method obtains the highest OA mainly depends on: the existence of residual module of ResNet-50 solves the problem of network degradation, and its network is relatively deep combined with convolution, which has good ability of feature extraction and excellent training ability. The second best method is the BBO (Li, 2020) algorithm, which is inspired by biogeography and uses migration and mutation to update data. Species will move from the habitat with higher HIS (habitat suitability index) values to the habitat with lower HIS values in order to improve the habitability of the habitat with lower HIS values. The core of BBO algorithm is the population migration and variation between different habitats. These two operations are also the main steps to achieve the optimization effect in the process of solving the problem. The third best method is CSO (Yang, 2017), which is a new group optimization algorithm based on the predator-prey strategy of cats, it mainly realizes global optimization through the combination of search mode and tracking mode. To solve the optimization problem by using cat swarm algorithm, the number of individuals involved in the optimization calculation should be determined first. After the cats have gone through the search mode and the tracking mode, their fitness is calculated based on the fitness function and the best solution in the current group is retained. Then, according to the MR (Mixture Ratio), the cat group was randomly divided into the searching part and the tracking part, and the iterative calculation was carried out by this method until the preset number of iterations was reached.

In this experiment, the ResNet-50 network has achieved good results compared with other methods. Next, we will try to use different layers of ResNet as well as several variants of ResNet, such as Wide Residual Network (WRN), ResNeXt and MobileNet to study facial emotion recognition. In the future work, we will test their performance.

5. Conclusion

In this paper, an improved facial expression recognition system is studied and a facial expression recognition method based on deep residual network is proposed. This paper focuses on the learning process of facial expression recognition. We use the current popular convolutional neural network algorithm, combined with the ResNet-50 residual network, which has achieved a good effect in the multi-classification task. Through the validation of the data set, the experimental results show

that the method proposed in this paper has good accuracy and good recognition effect in terms of average recognition accuracy.

In the future research, we will focus on the research of facial emotion recognition and try to collect more emotional images than in this experiment, so as to optimize and propose a better algorithm to train the hyperparameter of multi-layer feedforward neural network, such as the weights and biases. And we will also try such optimization algorithms based on the method mentioned above to improve the performance of multi-layer feedforward neural network. We will continue to explore ways based on deep residual network to improve the accuracy of facial expression recognition.

Declaration of Competing Interest

There is no conflict of interest.

References

Ali, H., Hariharan, M., Yaacob, S., & Adom, A. H. (2015, October). Facial emotion recognition based on higher-order spectra using support vector machines. *Journal Of Medical Imaging And Health Informatics, 5*, 1272–1277.

Ali, Y., Janabi-Sharifi, F., & Beheshti, S. (2021, February). Echocardiographic image segmentation using deep Res-U network. *Biomedical Signal Processing and Control, 64*(Article ID: 102248), 14.

Ashiquzzaman, A., Lee, H., Kim, K., Kim, H. Y., Park, J., & Kim, J. (2020, November). Compact spatial pyramid pooling deep convolutional neural network based hand gestures decoder. *Applied Sciences-Basel, 10*(Article ID: 7898), 22.

Belinschi, S. T., Bercovici, H., & Liu, W. H. (2021). The atoms of operator-valued free convolutions. *Journal of Operator Theory, 85*, 303–320 Win.

Benbahria, Z., Sebari, I., Hajji, H., & Smiej, M. F. (2021, February). Intelligent mapping of irrigated areas from landsat 8 images using transfer learning. *International Journal of Engineering and Geosciences, 6*, 41–51.

Bister, T., Erdmann, M., Glombitza, J., Langner, N., Schulte, J., & Wirtz, M. (2021, March). Identification of patterns in cosmic-ray arrival directions using dynamic graph convolutional neural networks. *Astroparticle Physics, 126*(Article ID: 102527), 10.

Choi, S. H., & Jung, S. H. (2020, March). Stable acquisition of fine-grained segments using batch normalization and focal loss with L1 regularization in U-Net structure. *International Journal of Fuzzy Logic and Intelligent Systems, 20*, 59–68.

Evans, F. (2017). Haar wavelet transform based facial emotion recognition. *Advances in Computer Science Research, 61*, 342–346 2017/03.

Furlong, L. S., Rossell, S. L., Caruana, G. F., Cropley, V. L., Hughes, M., & Van Rheenen, T. E. (2021, January). The activity and connectivity of the facial emotion processing neural circuitry in bipolar disorder: A systematic review. *Journal of Affective Disorders, 279*, 518–548.

Ganguly, B., Chaudhuri, S., Biswas, S., Dey, D., Munshi, S., Chatterjee, B., et al. (2021, March). Wavelet kernel-based convolutional neural network for localization of partial discharge sources within a power apparatus. *IEEE Transactions on Industrial Informatics, 17*, 1831–1841.

Garbin, C., Zhu, X. Q., & Marques, O. (2020, May). Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimedia Tools and Applications, 79*, 12777–12815.

Gonzalez-Yubero, S., Lazaro-Visa, S., & Palomera, R. (2021, January). How does emotional intelligence contribute to the study of personal protective factors for alcohol consumption in adolescence? *Psicologia Educativa, 27*, 27–36.

Govindaraj, V. V. (2019). High performance multiple sclerosis classification by data augmentation and AlexNet transfer learning model. *Journal of Medical Imaging and Health Informatics, 9*, 2012–2021.

Graumann, L., Duesenberg, M., Metz, S., Schulze, L., Wolf, O. T., Roepke, S., et al. (2021, January). Facial emotion recognition in borderline patients is unaffected by acute psychosocial stress. *Journal of Psychiatric Research, 132*, 131–135.

Guttery, D. S. (2021). Improved breast cancer classification through combining graph convolutional network and convolutional neural network. *Information Processing and Management, 58*(Article ID: 102439).

Hajarolasvadi, N., & Demirel, H. (2020, December). Deep facial emotion recognition in video using eigenframes. *IET Image Processing, 14*, 3536–3546.

Hanafi, W. N. W., & Daud, S. (2021). Managing sustainable development of government link companies (GLCs) in Malaysia through emotional intelligence and organisational politics. *International Journal of Innovation and Sustainable Development, 15*, 126–141.

Hasebe, T., & Ueda, Y. (2021). Unimodality for free multiplicative convolution with free normal distributions on the unit circle. *Journal of Operator Theory, 85*, 21–43 Win.

Huang, C. (2018). Multiple sclerosis identification by 14-layer convolutional neural network with batch normalization, dropout, and stochastic pooling. *Frontiers in Neuroscience, 12*(Article ID: 818) 2018-November-08.

Kwon, H., Pellauer, M., Parashar, A., & Krishna, T. (2021, January). Flexion: A quantitative metric for flexibility in DNN accelerators. *IEEE Computer Architecture Letters, 20*, 1–4.

Lee, D., Kim, J., & Jung, K. (2021, January). Improving object detection quality by incorporating global contexts via self-attention. *Electronics, 10*(Article ID: 90), 15.

Li, X. (2020). Facial emotion recognition via stationary wavelet entropy and biogeography-based optimization. *EAI Endorsed Transactions on e-Learning, 6*(Article ID: E4).

Lu, H. M. (2016). Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation. *IEEE Access, 4*, 8375–8385.

Muhammad, K. (2019). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications, 78*, 3613–3632.

Nayak, D. R., Das, D., Dash, R., Majhi, S., & Majhi, B. (2020, June). Deep extreme learning machine with leaky rectified linear unit for multiclass classification of pathological brain images. *Multimedia Tools and Applications, 79*, 15381–15396.

Nencka, A. S., Arpinar, V. E., Bhave, S., Yang, B. L., Banerjee, S., McCrea, M., et al. (2021). Split-slice training and hyperparameter tuning of RAKI networks for simultaneous multi-slice reconstruction. *Magnetic Resonance in Medicine, 9* [Article; Early Access]. 10.1002/mrm.28634.

Nishida, N., Oba, T., Unagami, Y., Cruz, J. P., Yanai, N., Teruya, T., et al. (2020, December). Efficient secure neural network prediction protocol reducing accuracy degradation. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, E103A*, 1367–1380.

Olimov, B., Karshiev, S., Jang, E., Din, S., Paul, A., & Kim, J. (2021). Weight initialization based-rectified linear unit activation function to improve the performance of a convolutional neural network model. *Concurrency and Computation-Practice & Experience*, 11 [Article; Early Access]. 10.1002/cpe.6143.

Pan, C. (2018). Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU. *Journal of Computational Science, 28*, 1–10 2018/09/01/.

Phillips, P. (2018). Intelligent facial emotion recognition based on stationary wavelet entropy and Jaya algorithm. *Neurocomputing, 272*, 668–676.

Polat, O., & Gungen, C. (2021). Classification of brain tumors from MR images using deep transfer learning. *Journal of Supercomputing*, 17 [Article; Early Access]. 10.1007/s11227-020-03572-9.

Rajananda, S., Zhu, J., & Peters, M. A. K. (2020, December). Normal observers show no evidence for blindsight in facial emotion perception. *Neuroscience of Consciousness, 6*(Article ID: Niaa023), 8.

Sahani, M., & Dash, P. K. (2021,). 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, 36*, 4006–4015.

Sangaiah, A. K. (2020). Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization. *Neural Computing and Applications, 32*, 665–680.

Satapathy, S. C. (2021). A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis. *Machine Vision and Applications, 32*(Article ID: 14).

Satapathy, S. C., & Wu, D. (2020). Improving ductal carcinoma in situ classification by convolutional neural network with exponential linear unit and rank-based weighted pooling. *Complex & Intelligent Systems*. 10.1007/s40747-020-00218-4.

Satapathy, S. C., & Zhu, L. Y. (2020). A seven-layer convolutional neural network for chest CT based COVID-19 diagnosis using stochastic pooling. *IEEE Sensors Journal* 1–1. 10.1109/JSEN.2020.3025855.

Staff, A. I., Luman, M., van der Oord, S., Bergwerff, C. E., van den Hoofdakker, B. J., & Oosterlaan, J. (2021). Facial emotion recognition impairment predicts social and emotional problems in children with (subthreshold) ADHD. *European Child & Adolescent Psychiatry [Article; Early Access], 13*. 10.1007/s00787-020-01709-y.

Tiwari, S. (2021, July). Dermatoscopy using multi-layer perceptron, convolution neural network, and capsule network to differentiate malignant melanoma from benign nevus. *International Journal of Healthcare Information Systems and Informatics, 16*, 58–73.

Togacar, M., Ergen, B., & Comert, Z. (2020, December). Classification of white blood cells using deep features obtained from convolutional neural network models based on the combination of feature selection methods. *Applied Soft Computing, 97*(Article ID: 106810), 10.

Wang, S.-. H. (2020). DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification. *ACM Transactions on Multimedia Computing, Communications, and Applications, 16*(Article 60).

Wang, S.-. H. (2021a). Covid-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network. *Information Fusion, 67*, 208–229 2020/10/09/.

Wang, S.-. H. (2021b). COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis. *Information Fusion, 68*, 131–148.

Wu, X. (2020). Diagnosis of COVID-19 by Wavelet Renyi entropy and three-segment biogeography-based optimization. *International Journal of Computational Intelligence Systems, 13*, 1332–1344 2020-09-17T09:29:20.000Z.

Yang, W. (2017). Facial emotion recognition via discrete wavelet transform, principal component analysis, and cat swarm optimization. *Lecture Notes in Computer Science, 10559*, 203–214.

Zhang, Y.-. D. (2020). Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation. *Information Fusion, 64*, 149–187 2020/12/01/.

Zhang, Y.-. D., Nayak, D. R., Zhang, X., & Wang, S.-. H. (2020). Diagnosis of secondary pulmonary tuberculosis by an eight-layer improved convolutional neural network with stochastic pooling and hyperparameter optimization. *Journal of Ambient Intelligence and Humanized Computing*. 10.1007/s12652-020-02612-9.