International Journal of Innovative
Computing, Information and Control
Volume **19**, Number **5**, October **2023**

ICIC International ©2023 ISSN 1349-4198
pp. **1661–1678**

EFFICIENT CNN MODEL BASED ON COMBINING RESIDUAL NETWORK AND DENSE-CONNECTED NETWORK ARCHITECTURES FOR FACIAL EXPRESSION RECOGNITION

Duong Thang Long

Faculty of Information Technology
Hanoi Open University
B101 House, Nguyen Hien Street, Hanoi Capital City 11600, Vietnam
duongthanglong@hou.edu.vn

Received February 2023; revised May 2023

Abstract. *Facial expression recognition (FER) is an essential aspect of human communication and has many practical applications. Convolutional neural networks (CNNs) have been successful in FER by learning complex image representations. Residual Networks (ResNets) and Dense-Connected Networks (DenseNets) are state-of-the-art architectures of CNNs that can reduce the vanishing gradient problem and improve feature extraction from images. This paper proposes a combination of ResNet and DenseNet for FER that takes advantage of their strengths to reduce the model complexity in terms of pararmeters and improve the ability of good feature extraction for the FER task. Our model has 5.6 million parameters, which is less complex than many modern CNN models for FER. We run experiments on popular datasets such as JAFFE, CK+, OuluCASIA, KDEF, and a mix of the first three (COJ). The results show at least 99.92% accuracy on the testing data of KDEF. Our model has the highest accuracy on testing data among all methods compared.*

Keywords: Deep learning, Convolutional neural network, Residual Dense-Connectivity networks, Facial expression recognition

1. **Introduction.** The facial expressions of human beings play an important role in any interpersonal communication, and it can help others to understand one's emotions or even intentions, making it an indispensable communication element in human interaction. With the development of computer vision technology and its practical application, as referred in [1] the results of various studies on FERs have shown a lot of promising successful applications in the fields of human-machine interaction, animation, medicine and education. As mentioned in [1], P. Ekman and W. Friesen identified six basic human facial expressions which they believe these emotions are presented in all people regardless of nationality, ethnicity or religion. These expressions are happiness (Ha), sadness (Sa), surprise (Su), disgust (Di), anger (An) and fear (Fe). Some authors used one or two more emotions such as contempt (Co), and neutral (Ne) [2]. Figure 1 shows an example of seven basic facial expressions from the KDEF dataset [3].

In facial expressions recognition, specifically movements in parts of the face such as raised eyebrows, locked eyebrows, and movements in the corners of the mouth, are considered basic units of change. However, facial expressions can vary greatly between individuals, and the expressions shown by different faces can also be different. These factors can greatly affect the performance and efficiency of any FER system, including those based on computer vision techniques.

DOI: 10.24507/ijicic.19.05.1661

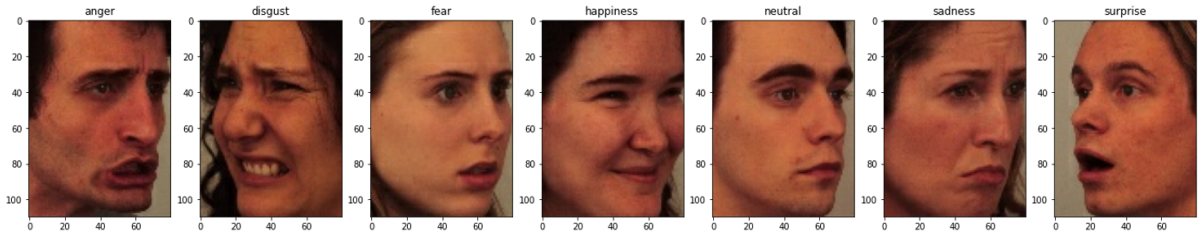


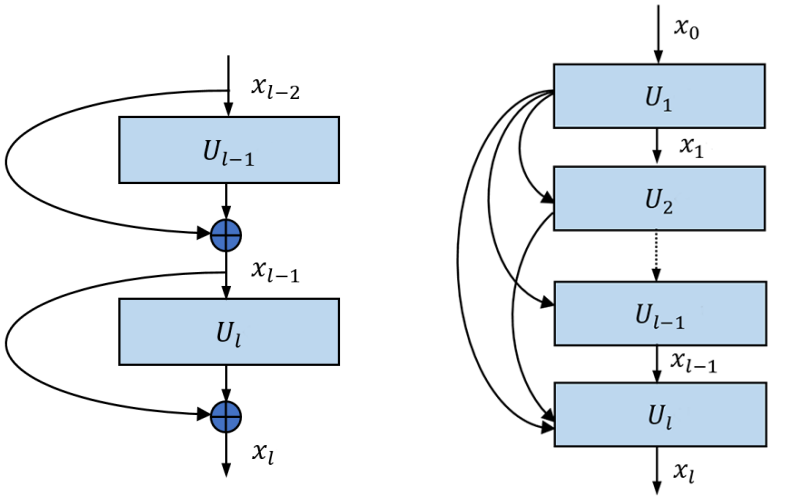
FIGURE 1. Example of seven basic facial expressions

Modern convolutional neural networks (CNNs) have become popular for FER tasks due to their ability to learn complex features from images, resulting in state-of-the-art performance on benchmark datasets [1-12]. CNNs can handle large amounts of data and learn hierarchical representations of the input, which makes them a powerful tool for this task. The typical approach for FER using CNNs involves several stages as the following.

- Pre-processing: The input images are pre-processed to remove any irrelevant information and normalize the data. This can include cropping the images to focus on the face region, and converting the images to grayscale or normalizing the color values.
- Feature extraction: In this stage, the CNN learns a rich, dense representation of the input data. This can be done using convolutional layers to capture local patterns in the images, and pooling layers to reduce the spatial dimensions while maintaining the important features.
- Classification: The extracted features are then passed through a series of fully connected layers to perform the final classification. This can be done using a softmax layer, which outputs a probability distribution over the different facial expression categories.
- Training: The CNN is trained using a large labeled dataset of facial expression images. The training process involves updating the weights of the network to minimize a loss function that measures the difference between the predicted and target facial expression labels.
- Evaluation: The trained CNN is then evaluated on a test dataset to assess its performance and identify any potential issues that may need to be addressed.

There are several popular CNN architectures that are well-suited for facial expression recognition tasks, including VGGNet, ResNet, InceptionNet and DenseNet [4,5]. VGGNet is a simple and plain architecture, while ResNet uses residual connections with shortcut connections to allow the network to learn residual functions and mitigate the problem of vanishing gradients. InceptionNet is a modular network architecture that uses multiple parallel branches to capture different scale and aspect ratios of features in the input images. DenseNet uses dense connections to concatenate all previous feature maps of a block to compute the output of the block; it can form a rich feature representation. Figure 2 illustrates these two networks, U is the abbreviation of a unit processing for several convolutional layers with activation layers and batch normalize layers, x presents the extracted feature map by a network layer, and $\oplus$ indicates element-wise addition.

Our motivation comes from the advantages of ResNet and DenseNet. The residual connections perform element-wise addition of the input and output of its block, allowing the network to learn residual functions that represent the difference between the input and desired output. While, dense connections provide a direct path for information to flow through the network. Thus, DenseNet can be effective in solving problems with limited data, as it allows the network to reuse feature maps and strengthen feature propagation.



(a) Residual connection (b) Dense connection

FIGURE 2. Illustration of residual connections and dense connections

Both of them help prevent the vanishing gradient problem and improve the flow of information through the network. We design two structures of residual connection blocks and densely connected blocks, which uses several unit processings (U) with convolutional layers, activations and batch normalization. We propose a combination of these two kinds of blocks for the architecture network of our model; it has earlier residual blocks for effectively extracting raw and low-level features. Then, following densely connected blocks is used to refine the extracted raw and low-level features. This combination offers several advantages for facial expressions recognition which helps to reduce the number of parameters in the network and improve its generalization ability, capturing both spatial and semantic information in the images, improving accuracy and it can lead to faster and more stable convergence during training. We evaluate the proposed model on some popular datasets of FER problems, which are in various poses and illuminations. These datasets are JAFFE, CK+, OuluCASIA and KDEF, which are widely used by authors in various articles [1-9,12,15].

The remainder of this paper is organized as follows. Section 2 introduces the related work on the combination of ResNet and DenseNet. Section 3 presents our network architecture and its detailed implementation. Section 4 shows the experimental results and compares our scheme with existing methods. Section 5 concludes the whole paper.

2. **Related Work.** The combination of ResNet and DenseNet (RDN) is a popular choice for many computer vision tasks [6-12] including facial expression recognition, because it combines the benefits of both architectures. ResNet helps to mitigate the vanishing gradient problem and allow for deep networks with many layers, while DenseNet provides a dense connectivity pattern that encourages feature reuse and can improve the representational capacity of the network.

There is a combination with embedding dense blocks inside residual blocks, i.e., the element-wise addition is applied to performing on the input and output of dense blocks. The dense blocks are designed to capture both low-level and high-level features of the image, while the residual connections allow for the network to better preserve the details of the original image. Zhang et al. [6] showed that their RDN model outperforms

existing state-of-the-art models on several benchmark datasets for the task of image super-resolution. This model utilizes the benefits of ResNet and DenseNet with allowing for easier optimization by alleviating the vanishing gradients problem by residual architecture and capturing local and global features effectively by dense-connected architecture.

Another combination uses multi element-wise additions and concatenations of previous feature maps in different orders between unit processing. Zhang et al. [11] used this kind to make a refined residual dense network to enhance global dense feature flow for character recognition. They show that their model can not only retain the advantages of residual dense blocks, i.e., local feature fusion and residual learning, but also refine the block structures to reduce the computing cost of inner layers. To ensure maximum global information flow between blocks, their model learns the global dense residual features fully. They also use two convolution layers with stride 2 and more channels to reduce the global feature size and extract more informative deeper features. This model experimentally shows that it outperforms the Random Forest or simple CNN methods.

Authors in [7-9] introduced modified RDN architectures for image enhancement. Song et al. [7] used a dynamic mechanism into the RDN architecture to improve its performance on image denoising tasks. A gate module is inserted into RDN for dynamically selecting one of two paths, one of the paths is an identity function, and the other consists of a dense block, local feature fusion, and local residual learning. The dynamic mechanism allows the network to adapt to different noise levels in the input image and adjust its parameters accordingly. This can better capture the inter-channel dependencies and improve the representation capability of the network. In [8,9], authors introduced efficient and advanced RDN for image super-resolution with good image reconstruction results in comparison. Wang et al. [8] proposed advanced deep RDN that can make full use of local hierarchical information and provide more hierarchical feature information, where its blocks can get an intermediate result, and the final reconstructed image can be obtained by weighted summation of all intermediate results. Song et al. [9] proposed three kinds of residual dense blocks (RDB): the first one is addition a 1×1 convolution into RDB to squeeze the channels of feature maps to construct a lean block (shrink RDB), the second one uses channel shuffle strategy before group convolution in RDB to offer a single output channel can see all preceding layers' features of the block (group RDB), and the last one contains four components including pooling, recursive convolution, upsample operator and local residual operator (contextual RDB). They also apply evolutionary algorithms to searching the neural architecture because of its excellent performance. They search for the best network model by selecting the kind of three RDBs, number of convolutional layers and some other parameters of the network. The authors show experimentally that their proposed method achieves good results in comparison with the original RDB.

Chen et al. [12] used three dense-connected blocks following two residual blocks for encoding feature interception, which connects each layer to every other layer in a feed-forward fashion. Their model has two phases, the earlier phase is an encoder and the after one is a decoder, the decoder has convolution operations and it is integrated with the encoder by attention gates. This model is designed for hepatocellular carcinoma segmentation. In [10], the authors used an integration of the ResNet and DenseNet models for dance action recognition. They used the ResNet model with good generalization performance to extract the deep action features and the DenseNet model was added to the multi-layer convolution layer to enhance the richness and effectiveness of the features. Their proposed method is tested on public datasets and compared with other existing methods, the experimental results verify the superior performance of the proposed method in practical application.

In the next section, we propose a model with new combination of residual blocks and dense-connected blocks to utilize advantages of them for effectively facial expression recognition.

3. **Proposed Method.** As mentioned, the residual connections allow networks to better preserve the details of the original image, and it can capture low-level features of the image. While the dense connections allow to capture high-level features, it can enhance the richness and effectiveness of the features [6,10]. Based on this, we propose a model for FER with two phases of extracting features: the first one is raw and low-level features extraction by residual blocks, and the second one is refining extracted features and making more and more high-level features by dense-connected blocks.

3.1. **Network architecture.** Our network uses two types of layer blocks: residual blocks (RBs) and densely connected blocks (DBs). The RBs are composed of three convolutional layers. In each layer, we apply batch normalization (BN) and 'relu' activation (RA) following the convolutional operation (CV) to compute output of the layer. The middle layer has a 3×3 convolution to capture local features, but its number of filters is a quarter of the desired number of output filters of the blocks, which leads to a reduction in the number of model parameters. The other two layers use 1×1 convolution to mainly change the dimensionality of the input, one is downsampling to feed into 3×3 CV and the other is upsampling for the output. We use BN before activation as it was originally suggested to reduce the amount by which the hidden unit values of the network shift around, also known as covariance shift. Normally, for the shortcut connection, we use a 1×1 CV on the input of the block to make the same number of filters in the last convolutional layer, then an addition ($\oplus$) is used for element-wise summation of the inputs and the output of the block before applying a 'relu' activation. This architecture of RBs is viewed as a pathway that learns to preserve the information being already present in the inputs and propagates it to the deeper layers of the network. We formalize the processing of this block as in Equation (1). Figure 3 shows the design of the residual blocks.

$$F^{RB} = f^{\oplus}\left(f^{3rd}\left(f^{2nd}\left(f^{1st}\left(F^{ip}\right)\right)\right), f^{S}\left(F^{ip}\right)\right), \tag{1}$$

where, $f^{\oplus}$, f^{1st}, f^{2nd}, f^{3rd} and f^{S} are functions as operations of addition with 'relu' activation, three convolutional layers (1st, 2nd and 3rd) and shortcut connection with convolution in case needing, respectively. Symbol "nf" denotes the number of filters or the output channels of operations. F^{ip} is input feature map and F^{RB} is output feature map of the residual blocks; we use $F^{RB(nf)}$ for the case where the desired output channel size is "nf".

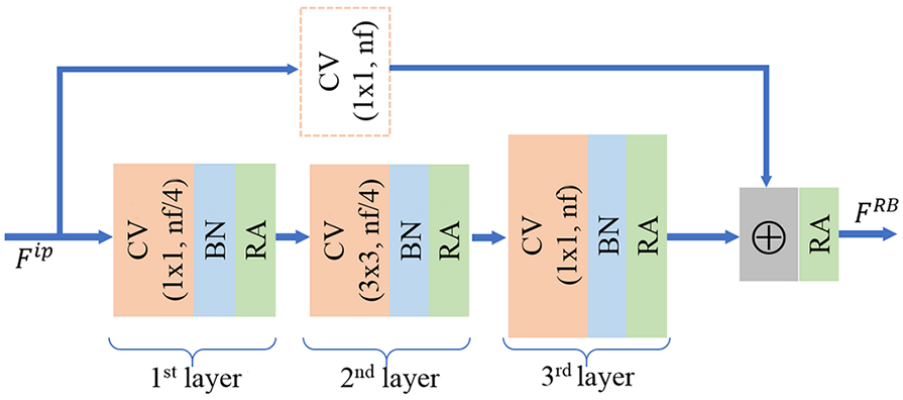


FIGURE 3. Design of our residual blocks (RBs)

For densely connected blocks (DBs), we apply several pairs of convolutional layers with two different kernel sizes, 1×1 in the first one and 3×3 in the second one. For the reasonability of high-level features extraction by the DBs, we use the first layer with 4 times as many filters as the second layer in order to upsample the input. This intends extracting not only high-level features but also getting more spatial and semantic information from the input. In addition, unlike RBs, we apply BN before CV in the DBs to keeping the refined features. At the end of each pair, a concatenation ([,]) of all previous feature maps is applied, instead of just the feature maps from the immediately preceding layer. This is a key function of the DBs. Transition layers (TLs) are applied after each DB to reducing the number of output filters, so it simplifiers the computational complexity of the network. The TLs have a 1×1 convolution operation, a 'relu' activation, and a 2×2 average pooling (AP) with strides 2 (denoted by 2×2-2) to make a transition and output the same number of filters as the immediately preceding layer. Figure 4 shows details of DBs. Equation (2) formalizes the processing of the blocks.

$$F^{DB} = f^{TL}\left(f^{[,]}\left(\cdots, f^{[,]}\left(f^{p2}\left(f^{p1}\left(F^{ip}\right)\right), F^{ip}\right)\right)\right), \tag{2}$$

where, f^{TL}, $f^{[,]}$, f^{p1} and f^{p2} are functions as layers of transition, concatenation, pair of convolutional layers (p_1 and p_2), respectively. F^{DB} is output feature map of the densely connected blocks. The number of used $f^{[,]}$ is the number repetition of pairs of convolutional layers; we also use $F^{DB(nf)}$ for the case where the desired output channel size is "nf".

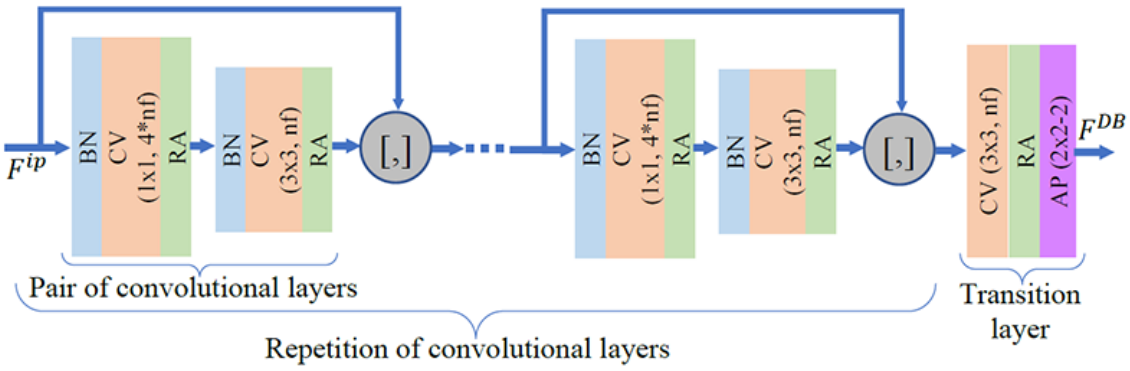


FIGURE 4. Design of our densely connected blocks (DBs)

Overall, our model consists of three residual blocks preceding three densely connected blocks consecutively. These blocks have different numbers of filters and repetition of convolutional layers in DBs. To effectively extract raw and low-level features from the RBs, a larger number of filters are utilized compared to those in the DBs. The filters in the RBs are 128, 256, and 512, while the filters in the DBs are 16, 32, and 64. The extracted features are then refined by the DBs through repeated pairs of convolutional layers, with 4, 8, and 16 repetitions. However, the features extracted by the RBs are also preserved and carried to the end of the model by concatenations in DBs for classification. We use global averaging pooling to aggregate and flatten the extracted features, followed by a fully connected layer with a 'softmax' activation at the end of the model, for facial expression recognition. Its name is RDFER.

As the network progresses through the layers, the features become more abstract and representative of higher-level structures in the image. This hierarchical feature extraction process is a key characteristic of deep convolutional neural networks and enables them to learn robust representations for image classification tasks. So, the beginning of the model

FIGURE 5. The structure of our model for FER (RDFER)

uses convolutional and pooling layers to extract raw and low-level features from images directly. These early layers typically have medium filter sizes and fewer filters, they are 7×7 and 64 in the model respectively, which enables them to capture the features such as edges, textures, and shapes. Figure 5 illustrates the entire model, where the numbers in RBs denote the number of filters and in DBs denote the number of filters and repetitions, respectively.

Finally, the RDFER model has a total of 66 convolutional layers which are divided into 6 blocks, serving as feature extractors. Despite having a relatively high number of layers, this model has only about 5.6 million parameters, making it less complex compared to other modern CNN models for image classification problems. The low number of parameters is achieved by using small kernel sizes for the convolutional operations. The details of the model's parameters with a specific input size in height $\times$ width $\times$ channels of $100 \times 90 \times 3$ are shown in Table 1, where the symbol $\otimes$ presents the repetition in DBs, "C" represents convolutional operations with specified kernel size, strides and number of filters, "P" presents pooling operations with specified window size and strides, and "$\rightarrow$" represents the forward connection between two layers.

3.2. **Image pre-processing and augmentation.** As in [1,2], input images are usually obtained from user devices in practical applications, and they may consist of a background with many objects present. To detect faces on the images, we can use the well-known model named MTCNN [13]. The background is then removed from the images, and face images are fed into the model for training or recognition. Furthermore, to improve the quality of prediction, input can be enhanced for inference phase by using pre-processing images as in [14].

In order to increase the robustness of the model and prevent overfitting during training, we augment the training images using 2D image processing techniques [1,2]. These techniques, such as noise addition, rotation, flipping, cropping, shifting, and color adjustment, aim to increase the diversity of the training data and make the model more resistant to variations in input images such as changing styles, illumination, positions, and perspectives. With a given face image a, the augmented results are formalized as follows:

$$\left\{\Im^{\alpha}\left(a, p^{\alpha}\right)\right\}, \tag{3}$$

where, $\Im^{\alpha}$ represents the augmenting operation with parameters p^{α}, $\alpha = \{noise, rotation, zoom, shifting, flipping, \ldots\}$. The value of p^{α} varies based on the type of α, and we also use any number of α-operations in practical running. For example, Figure 6 shows 18

TABLE 1. Parameters of the RDFER model

Type layers/blocks	**Operations (kernel size-strides, filters)**	**#Parameters (thousand)**
Input	–	–
1st convolutional layer	C(7×7-2, 64)	9.5
1st residual block	C(1×1-1, 32)	2.1
	→C(3×3-1, 32)	9.3
	→C(1×1-1, 128)	4.2
Shortcut connection	C(1×1-1, 128)	8.3
2nd residual block	C(1×1-1, 64)	8.3
	→C(3×3-1, 64)	36.9
	→C(1×1-1, 256)	16.6
Shortcut connection	C(1×1-1, 256)	33.0
3rd residual block	C(1×1-1, 128)	32.9
	→C(3×3-1, 128)	147.6
	→C(1×1-1, 512)	66.1
Shortcut connection	C(1×1-1, 512)	131.6
1st dense block	[C(1×1-1, 64)	137.6
	→C(3×3-1, 16)] ⊗ 4	36.9
Transition	C(1×1-1, 16)→P(2×2-2)	9.2
2nd dense block	[C(1×1-1, 128)	132.1
	→C(3×3-1, 32)] ⊗ 8	295.2
Transition	C(1×1-1, 32)→P(2×2-2)	8.7
3rd dense block	[C(1×1-1, 256)	2101.2
	→C(3×3-1, 64)] ⊗ 16	2360.3
Flatten	Average global pooling	–
Classifier (FC layer)	Softmax	7.4
Total		**5.6M**

augmented images from an original one of 'happiness' emotion in OuluCASIA dataset [15] with random parameters. The first row displays an original image and the following three rows showcase its augmented versions, labeled by their index. These augmented images are more diverse, providing the training model with increased stability for feature extraction despite variations in face poses, illumination, position, and perspectives of input images.

However, the parameters for these augmented operations are carefully selected to ensure that important information about facial expressions is preserved for feature extraction. For example, excessive rotation or shifting of an image could result in the loss of crucial facial expression information and make it difficult for the model to extract proper features and recognize. In Figure 6, the 16th augmented image in the 4th column of the last row is excessively shifting and rotating with a lot of noise, so it may be lost information. In this study, we randomly select values for the parameters within a suitable range for each enhancement operation, and in some cases, an image may undergo multiple enhancement operations simultaneously.

4. **Experimental Results.** In this section, we present the datasets and parameters used for training our RDFER model, and provide a comprehensive analysis of the training results. Additionally, we conduct a real-time demonstration experiment to demonstrate the efficacy of our model and compare its performance with other existing methods in the field.

FIGURE 6. Augmented images of an original one in OuluCASIA dataset

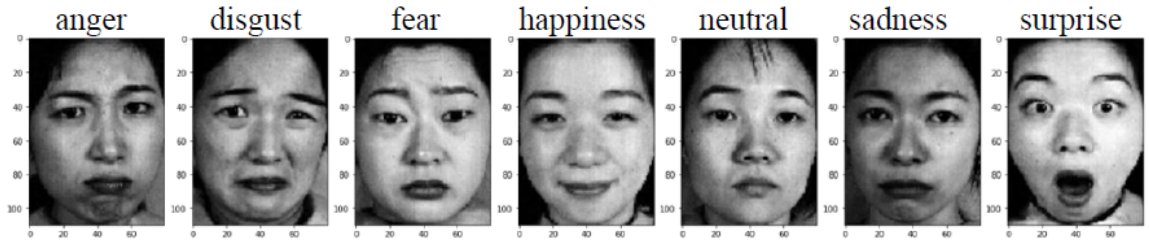
4.1. **Datasets and parameters.** For the experimental study, we utilized four datasets, namely, JAFFE [16], CK+ (Extended Cohn-Kanade) [17], OuluCASIA [15], and KDEF [3]. A combined dataset, referred to as COJ [1], was created by merging the first three datasets.

The OuluCASIA dataset has 1440 images from 80 people with 6 basic facial expressions under varying illumination and head poses in color. The JAFFE dataset comprises 213 images from 10 Japanese women, including 6 basic emotions and a 'neutral' emotion. The CK+ dataset contains 981 images from 118 individuals, each displaying 6 basic emotions plus the 'contempt' emotion. Both JAFFE and CK+ datasets are in grayscale. The COJ dataset has a total of 2634 images with 8 facial expression labels. The images of COJ were converted to the same size and grayscale. This dataset has imbalanced class distributions as shown in Table 2, where symbol '−' indicates no image. The 'neutral' emotion class has the lowest number of images (30), while the 'surprise' emotion class has the largest number of images (519), which is more than nearly twenty times the size of the smallest class. The KDEF dataset contains 4900 images with 6 basic emotions plus the 'neutral' expression from 5 different angles. It includes images of 70 individuals (35 females and 35 males) adults between 20 and 30 years old. Each image was taken without occlusions, such as mustaches, earrings, and eyeglasses. Figure 7 displays examples of the facial expressions in these datasets.

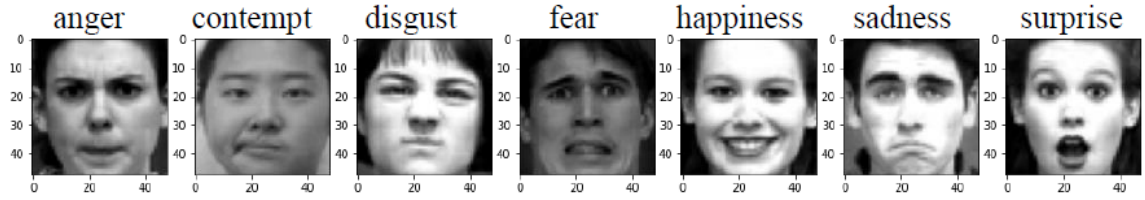
TABLE 2. Distribution of images in every facial expressions

Facial expressions	JAFFE	CK+	OuluCASIA	KDEF	COJ
anger	30	135	240	700	405
contempt	–	54	–	–	54
disgust	29	177	240	700	446
fear	32	75	240	700	347
happiness	31	207	240	700	478
neutral	30	–	–	700	30
sadness	31	84	240	700	355
surprise	30	249	240	700	519
Total	**213**	**981**	**1440**	**4900**	**2634**

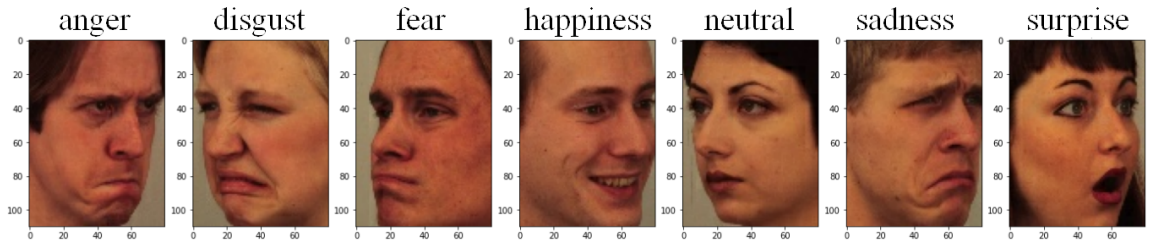
(a)



(b)



(c)



(d)

FIGURE 7. Example images from OuluCASIA (a), JAFFE (b), CK+ (c), and KDEF (d)

We utilized 5-folds cross-validation to conduct our experiments. The dataset was randomly divided into 5 equal-sized folds. In each iteration, one fold was used for testing (D^{te}), and the remaining folds were used for model training (D^{tr}) with using 20% of them for model selection evaluation (D^{va}). This process was repeated 5 times with each fold used for testing once, and the final results were reported as the mean and standard deviation of the 5 runs. To improve recognition accuracy, we augmented the training data by implementing the $\Im^{\alpha}$ operation as defined in Equation (3). The augmentation parameters were randomly selected from the specified range in Table 3, with the purpose of retaining the essential information of an image during the augmentation process. To augment the training dataset, each image was subjected to 10 augmentations, leading to a 5-folds increase in the size of the training data. This augmented data diversifies the dataset and mitigates overfitting.

Table 3. Limitation ranges for randomly selecting augmentation parameters

No	Parameters of data augmentation	Limitation range
1	Variance of Gaussian noise addition	$[0, 0.1]$
2	Rotation relative to original image (radian, negative is counter-clockwise)	$[-0.1\pi, 0.1\pi]$
3	Shifting relative to size of original image (percentage, negative is left or up shifting, both width and height)	$[-10\%, 10\%]$
4	Scaling relative to size of original image (negative is downscaling, both width and height)	$[-10\%, 10\%]$
5	Horizontal flipping image	True/False

We utilized the widely-used Adam optimization method [18], expressed in Equation (4), to update the model parameters during training progress.

$$w_t = w_{t-1} - \eta \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon}, \tag{4}$$

where w_t presents the model parameters at iteration t, $\widehat{m_t}$ is the aggregated gradients, $\widehat{v_t}$ is the sum of square of previous gradients, η is the learning rate and ϵ is a hyperparameter that sets the precision threshold. The detailed computation for the optimization can be found in [18]. For this study, we set a learning rate of $\eta = 10^{-5}$, a batch size of 128, and trained the model for 150 epochs.

The experiments were conducted on a computer system equipped with TPU and 32 Gb RAM. We developed the proposed model using the Python programming language in the TensorFlow platform, which is a widely used deep learning framework known for its powerful features in image processing and CNN modeling.

4.2. **Results and discussion.** The training loss and accuracy of the RDFER model were evaluated over 5 runs using a 5-folds cross-validation approach as illustrated in Figure 8. Each subfigure displays the loss and accuracy of both the training data (solid line) and validation data (dotted line). Due to the small number of images and similarity in facial expressions classes within the JAFFE dataset, there were fluctuations of validation loss in the training process. The model achieved high accuracy and low loss on the training data starting from around the 25th epoch, and these metrics continued to improve throughout the remaining training epochs.

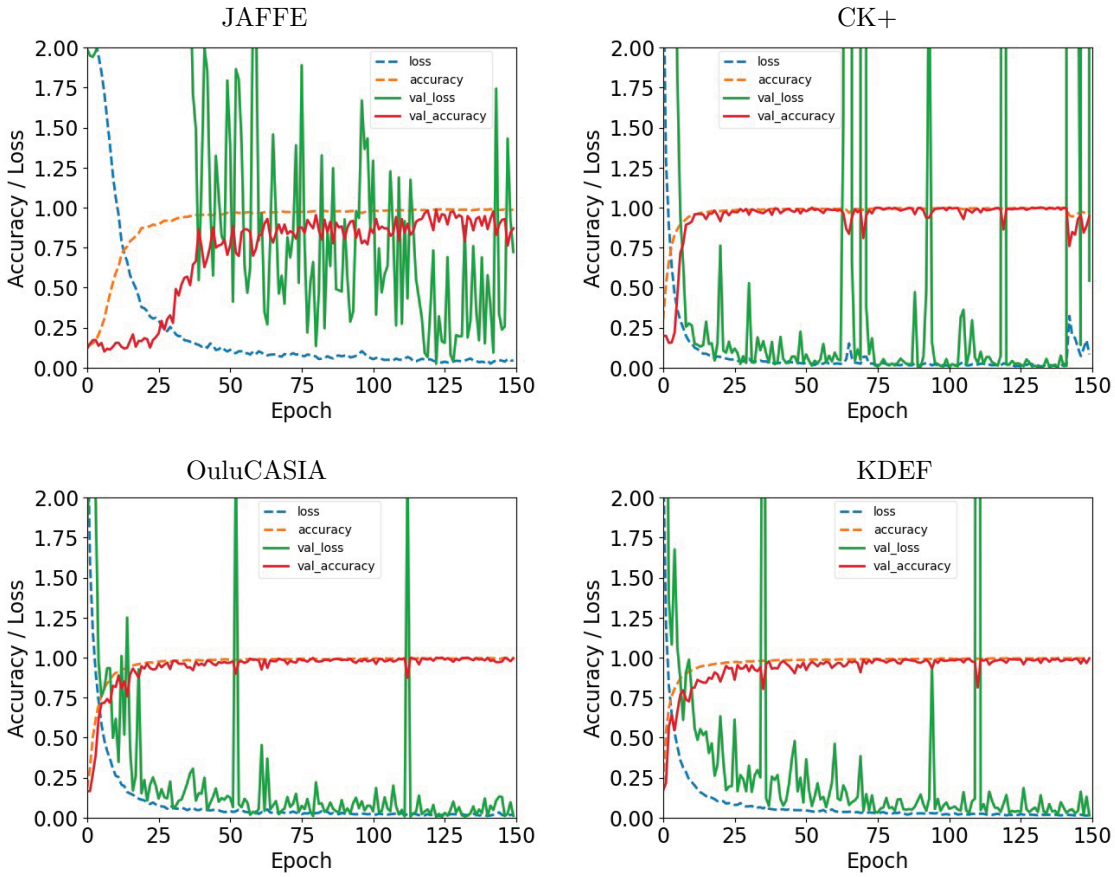
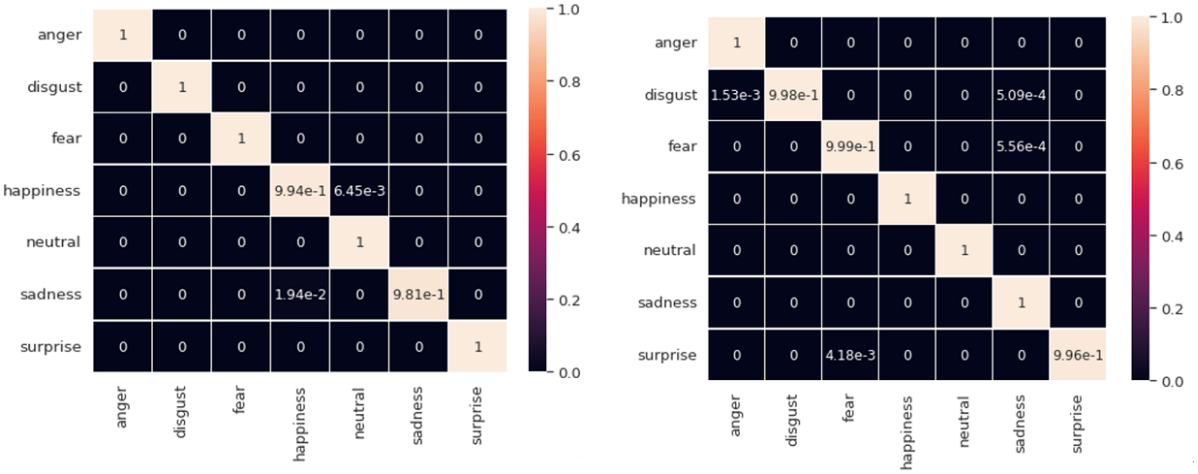


FIGURE 8. Loss and accuracy of training data and validation data



(a) (b)

FIGURE 9. Confusion matrices of RDFER model on JAFFE (a) and KDEF (b) datasets

To provide an overview of the performance of the RDFER model, we generated confusion matrices based on 5 runs of the model on the datasets, as shown in Figure 9. Figures 9(a), and 9(b) correspond to the JAFFE, and KDEF datasets, respectively. The two datasets, CK+ and OuluCASIA, have achieved perfect accuracy for all facial expressions; therefore, their confusion matrices contain only diagonal values of 1 and are not shown. In

these matrices, each row corresponds to facial expression in the dataset (target), and each column corresponds to a facial expression which is predicted by the model. We apply the trained model to recognizing all images in the dataset, including the training, validation, and testing data. The numbers in the confusion matrices represent the ratio of accuracy and were averaged over the 5 runs of the model.

The confusion matrices reveal that the RDFER model achieved high accuracy on all datasets, with only a few pairs of labels that were frequently confused. In JAFFE, the 'sadness' emotion was the most frequently confused with 'happiness' (1.94%), followed by 'happiness' confused with 'neutral' (0.65%). In KDEF, 'disgust' was confused with 'anger' and 'sadness' at rates of 0.15% and 0.05%, respectively, while 'fear' was confused with 'sadness' (0.06%) and 'surprise' (0.42%). Notably, the model achieved perfect accuracy on three emotions ('anger', 'disgust', and 'fear') in JAFFE and two emotions ('happiness' and 'neutral') in KDEF, without any instances of confusion. In total, there are 16 images in Figure 10 that have confusion in the datasets. The title of each image shows a pair of labels separated by the ">" symbol, where the label on the left is the ground truth and the label on the right is the predicted label.

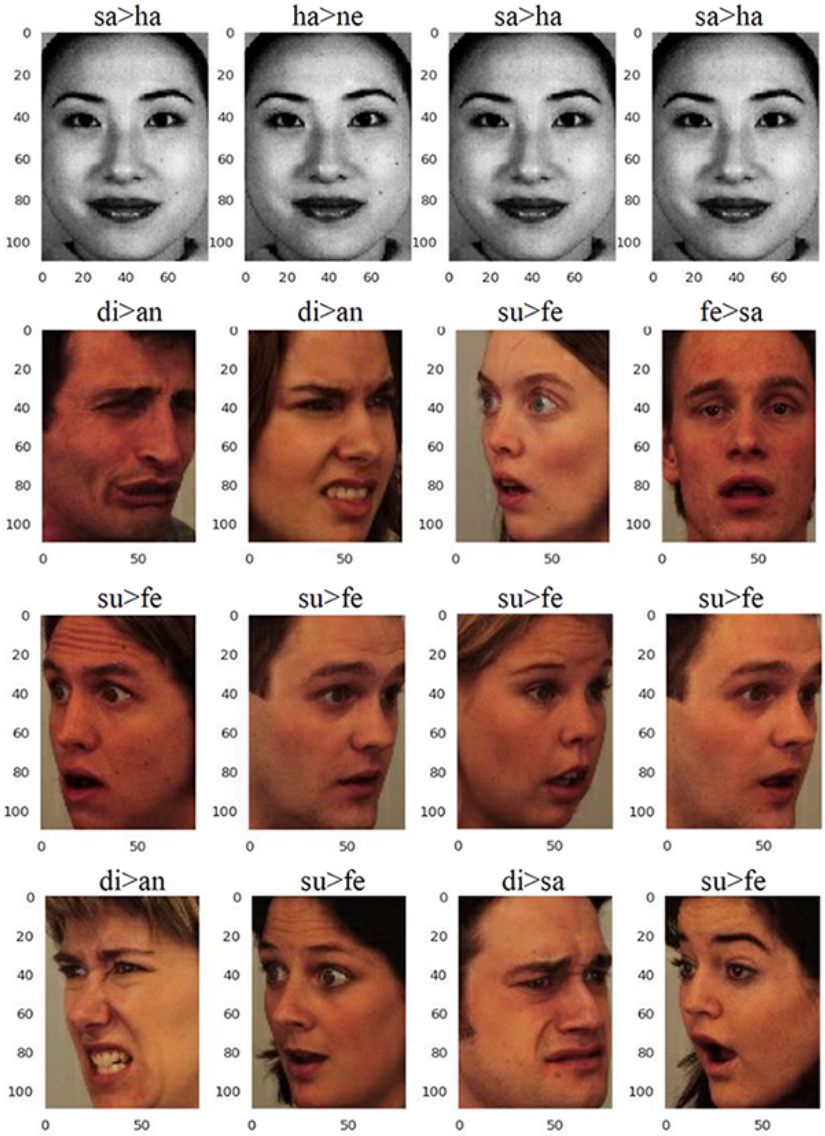
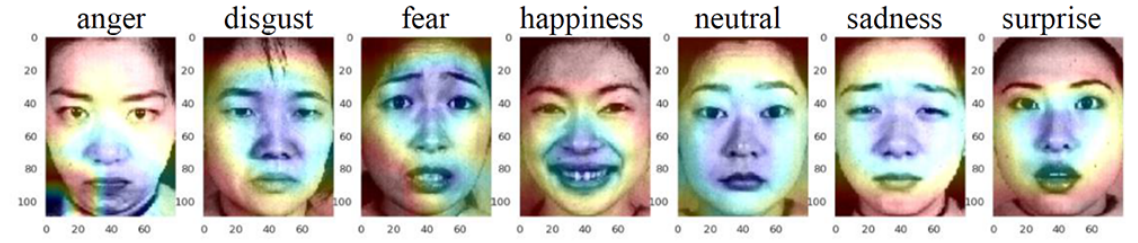
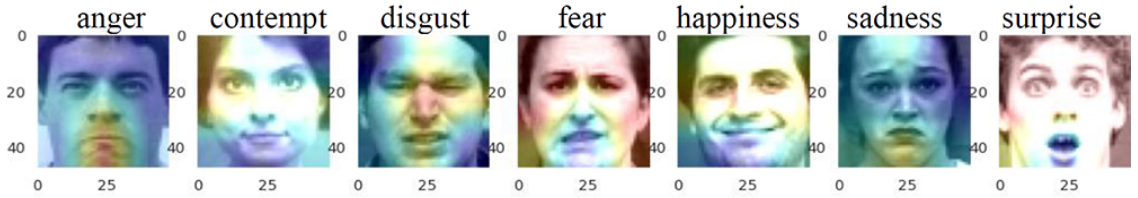


FIGURE 10. Images of confused recognition

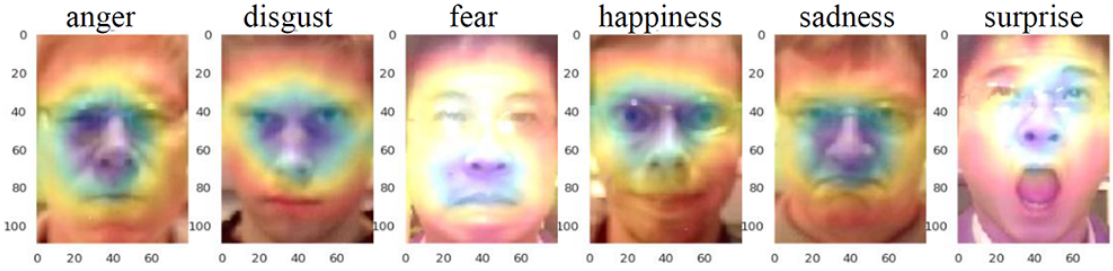
We visualize the model performing feature extraction on input images, i.e., convolutional layers in RDFER acting as feature extractors of facial expressions. The heatmap of the final convolutional layers with respect to input images using the Grad-CAM method [19] is shown in Figures 11(a), 11(b), 11(c) and 11(d) corresponding to JAFFE, CK+, OuluCASIA and KDEF datasets, respectively. This shows the interesting area of the convolutional layers on the input images for selecting features is called the 'gradient-based localization' method. The highlighted areas of input images are important in representing facial expressions, such as the mouth and eyes. This intuitive visualization reveals that the RDFER model focuses on these areas to extract descriptive features for facial expressions. Conversely, when these areas are not taken into account, it becomes difficult to accurately identify the correct facial expression. In all images, the heatmaps were mostly concentrated on the mouth area, with some variation on the eyes. This suggests that the convolutional layer in the RDFER model focuses on these regions to extract features of facial expressions, and ignoring them can lead to inaccurate facial expression recognition.



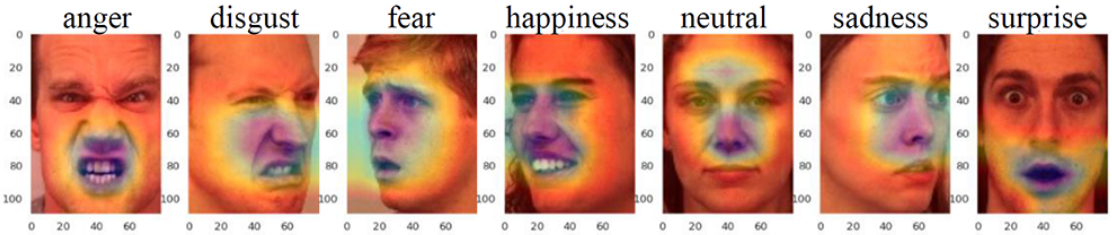
(a)



(b)



(c)



(d)

FIGURE 11. Heatmaps of the final convolutional layers on images of datasets

To evaluate the importance of the combination of residual and densely connected blocks, we conducted experiments using only one type of block with the same scenario and parameters as in the JAFFE dataset. Specifically, we replace the residual blocks (RBs) with densely connected blocks (DBs) to create the Dense-only Facial Expression Recognition (DoFER) model, and vice versa to create the Residual-only Facial Expression Recognition (RoFER) model. The running details of results are presented in Table 4. Despite having over 13.5 million parameters (241% more than RDFER), the DoFER model achieved lower accuracy (99.52%) compared to our RDFER model (100%). On the other hand, the RoFER model has 1.2 million parameters (21.4% of RDFER), so its accuracy (98.6%) is slightly lower than DoFER and significantly lower than the RDFER model. It is important to note that the accuracies on the training data are lower than those of the validation and testing data. This is due to the fact that the training data is augmented 10 times with increased diversity, resulting in a more challenging learning task for the model.

TABLE 4. Running accuracies on D^{tr}, D^{va} and D^{te} of our models in JAFFE

Run	RDFER (5.6M)			DoFER (13.5M)			RoFER (1.2M)		
	D^{tr}	D^{va}	D^{te}	D^{tr}	D^{va}	D^{te}	D^{tr}	D^{va}	D^{te}
1	97.97%	100%	100%	98.75%	100%	100%	98.75%	100%	97.62%
2	98.91%	100%	100%	95.94%	100%	100%	97.58%	100%	97.62%
3	98.75%	100%	100%	98.91%	100%	100%	95.86%	100%	100%
4	99.06%	100%	100%	99.14%	100%	97.62%	99.69%	100%	100%
5	99.30%	100%	100%	99.84%	100%	100%	99.22%	93.94%	97.78%
Mean	**98.80%**	**100%**	**100%**	**98.52%**	**100%**	**99.52%**	**98.22%**	**98.79%**	**98.60%**

Table 5 provides a comparison of our results with other state-of-the-art methods of convolutional neural networks. The '*' symbol denotes how the datasets were divided for testing, where 10F refers to 10-folds, 5F refers to 5-folds in the cross-validation scenario, and 0.5T means 50% of data samples were used for testing. Symbol '−' indicates having no experimental results. We chose the best-performing case from [4] as the authors conducted a survey of various methods. Some methods have total parameters of their models that it is shown in brackets '()'.

Our RDFER model achieves the highest accuracy among all methods in all datasets, with a perfect accuracy of 100% in JAFFE, CK+, OuluCASIA, and COJ, and a 99.92% accuracy in KDEF. The method described in [1] also achieved a perfect accuracy of 100% in OuluCASIA. All the best cases are bold in each column of the dataset. There are seven cases with high accuracy above 99% indicated by underlines. Some methods use very large models, such as in [2,20] while others use smaller models with 1.6M in [21] and 2.4M in [1], both of which are smaller than our model. In particular, our model uses a combination of residual and dense-connected blocks to achieve a trade-off between model complexity and performance. With a small number of parameters (5.6M), it takes up only 23.8% of the parameters used in [20] and 14.4% of those in [2], while still outperforming these methods with 0.5% and 1.6% better accuracy, respectively, on the OuluCASIA dataset. On the JAFFE, CK+, and COJ datasets, the method in [1] used only dense connectivity architectures with a smaller number of parameters than our model. However, it was not able to achieve perfect accuracy of 100%, while our model, which uses a combination of residual and dense connectivity architectures, achieved 100% accuracy on these datasets. Although our model has a larger number of parameters, it provides improved performance over the method in [1]. On the KDEF dataset, two methods in [21,22] have a smaller number of parameters compared to ours. However, their accuracy

Table 5. Comparison of results to other methods

Methods	JAFFE	CK+	OuluCASIA	KDEF	COJ
Deng and Li [4] (best case)	95.80%	99.60%	91.67%	–	–
Zhou et al. [22] (0.06M) *0.5Test	–	–	–	87.71%	–
Ming et al. [23] (39M) *10-folds	–	89.60%	99.50%	–	–
Zhao et al. [20] *10-folds	–	97.85%	89.23%	–	–
Devaram and Cesta [21] (1.6M) *5-folds	80.09%	84.27%	–	99.90%	–
Lai et al. [24] (2.5M) *5-folds	–	97.30%	–	–	–
Tang et al. [25] *10-folds	–	98.68%	–	–	–
Long [2] (23.5M) *10-folds	–	99.68%	98.47%	–	–
Long et al. [1] (2.4M) *5-folds	99.08%	99.90%	**100%**	–	99.92%
Proposed RDFER (5.6M) *5-folds	**100%**	**100%**	**100%**	**99.92%**	**100%**

is lower than ours, with 87.71% and 99.90% compared to our 99.92%. These percentages are also lower than our accuracy percentages of 87.78% and 99.98%, respectively. These results once again demonstrate that our proposed model, which combines residual and dense-connected blocks, harnesses the advantages of both architectures, resulting in a more powerful model.

5. **Conclusion.** In this paper, we proposed a facial expression recognition model that combines residual and dense-connected architectures to enhance accuracy while maintaining moderate model complexity. Our model consists of three residual blocks and three dense-connected blocks, totaling 66 convolutional layers. Despite its relatively high number of layers, our model has only 5.6 million parameters, making it less complex than many modern CNN models for FER. The proposed model achieved significant results in facial expression recognition on all datasets. It achieved the highest accuracy of 100% on JAFFE, CK+, and OuluCASIA, and the lowest accuracy of 99.92% on the KDEF dataset. In comparison to other models, our proposed model had the highest accuracy across all datasets. When the proposed model was tested without the combination of residual and dense-connected architectures, it did not achieve the same accuracy as the original model on JAFFE, and had a larger number of parameters. Our model is suitable for practical applications as it is moderate in complexity and can be integrated into systems with computational resource limitations.

For future research, we plan to explore the design of multi-task models based on the combination of residual and dense-connected architectures for image classification problems. We also aim to integrate this model into practical systems for applications such as online training and examination monitoring.

Acknowledgment. This work is supported by the Hanoi Open University, Vietnam under the Grant No. MHN2022-01.21.

REFERENCES

[1] D. T. Long, T. T. Tung and T. T. Dung, A facial expression recognition model using lightweight dense-connectivity neural networks for monitoring online learning activities, *International Journal of Modern Education and Computer Science*, vol.6, pp.53-64, 2022.

[2] D. T. Long, A facial expressions recognition method using residual network architecture for online learning evaluation, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol.25, no.6, pp.1-10, 2021.

[3] G. Ellen, D. R. Rudi, L. Lemke and V. Bruno, The Karolinska directed emotional faces: A validation study, *Cognition & Emotion*, vol.22, no.6, pp.1094-1118, 2008.

[4] W. Deng and S. Li, Deep facial expression recognition: A survey, *IEEE Transactions on Affective Computing*, vol.13, pp.1195-1215, 2022.

[5] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. V. Essen, A. A. S. Awwal and V. K. Asari, A state-of-the-art survey on deep learning theory and architectures, *Electronics*, vol.8, no.3, 292, https://doi.org/10.3390/electronics8030292, 2019.

[6] Y. Zhang, Y. Tian, Y. Kong, B. Zhong and Y. Fu, Residual dense network for image super-resolution, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.2472-2481, 2018.

[7] Y. Song, Y. Zhu and X. Du, Dynamic residual dense network for image denoising, *Sensors*, vol.19, no.17, 3809, https://doi.org/10.3390/s191738093809, pp.1-14, 2019.

[8] W. Wang, Y. Jiang, Y. Luo, J. Li, X. Wang and T. Zhang, An advanced deep residual dense network (DRDN) approach for image super-resolution, *International Journal of Computational Intelligence Systems*, vol.12, no.2, pp.1592-1601, 2019.

[9] D. Song, C. Xu, X. Jia, Y. Chen, C. Xu and Y. Wang, Efficient residual dense block search for image super-resolution, *The 34th AAAI Conference on Artificial Intelligence (AAAI-20)*, pp.12007-12014, 2020.

[10] X. Yang, Y. Lyu, Y. Sun and C. Zhang, A new residual dense network for dance action recognition from heterogeneous view perception, *Frontiers in Neurorobotics*, vol.15, pp.1-8, 2021.

[11] Z. Zhang, Z. Tang, Y. Wang, Z. Zhang, C. Zhan, Z. Zha and M. Wang, Dense residual network: Enhancing global dense feature flow for character recognition, *Neural Networks*, vol.139, pp.77-85, 2021.

[12] W.-F. Chen, H.-Y. Ou, H.-Y. Lin, C.-P. Wei, C.-C. Liao, Y.-F. Cheng and C.-T. Pan, Development of novel residual-dense-attention (RDA) U-Net network architecture for hepatocellular carcinoma segmentation, *Diagnostics*, vol.12, no.8, 1916, https://doi.org/10.3390/diagnostics12081916, 2022.

[13] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, *IEEE Signal Processing Letters*, vol.23, no.10, pp.1499-1503, 2016.

[14] K. R. Ummah, T. Karlita, R. Sigit, E. M. Yuniarno, I K. E. Purnama and M. H. Purnomo, Effect of image pre-processing method on convolutional neural network classification of COVID-19 CT scan images, *International Journal of Innovative Computing, Information and Control*, vol.18, no.6, pp.1895-1912, 2022.

[15] G. Zhao, X. Huang, M. Taini, S. Z. Li and M. Pietikäinen, Facial expression recognition from near-infrared videos, *Image and Vision Computing*, vol.29, pp.607-619, 2011.

[16] M. Lyons, S. Akamatsu, M. Kamachi and J. Gyoba, Coding facial expressions with Gabor wavelets, *Proc. of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, pp.200-205, 1998.

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

[18] D. P. Kingma and J. L. Ba, Adam: A method for stochastic optimization, *Proc. of the 3rd International Conference on Learning Representations (ICLR)*, pp.1-15, 2015.

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, *IEEE International Conference on Computer Vision (ICCV)*, pp.618-626, 2017.
[20] R. Zhao, T. Liu, J. Xiao, D. P. Lun and K.-M. Lam, Deep multi-task learning for facial expression recognition and synthesis based on selective feature sharing, *The 25th International Conference on Pattern Recognition (ICPR)*, pp.4412-4419, 2020.
[21] R. R. Devaram and A. Cesta, LEMON: A lightweight facial emotion recognition system for assistive robotics based on dilated residual convolutional neural networks, *Sensors*, vol.22, no.9, 3366, https://doi.org/10.3390/s22093366, 2022.
[22] N. Zhou, R. Liang and W. Shi, A lightweight convolutional neural network for real-time facial expression detection, *IEEE Access*, vol.9, pp.5573-5584, 2020.
[23] Z. Ming, J. Xia, M. Luqman, J.-C. Burie and K. Zhao, Dynamic multi-task learning for face recognition with facial expression, *Lightweight Face Recognition Challenge Workshop during the 2019 International Conference on Computer Vision (ICCV2019)*, Seoul, Korea, 2019.
[24] S.-C. Lai, C.-Y. Chen and J.-H. Li, Efficient recognition of facial expression with lightweight octave convolutional neural network, *Journal of Imaging Science and Technology*, vol.66, no.4, DOI: 10.2352/J.ImagingSci.Technol.2022.66.4.040402, 2022.
[25] X. Tang, S. Liu, Q. Xiang, J. Cheng, H. He and B. Xue, Facial expression recognition based on dual-channel fusion with edge features, *Symmetry*, vol.14, no.12, 2651, https://doi.org/10.3390/sym14122651, 2022.
[26] A. Greco, N. Strisciuglio, M. Vento and V. Vigilante, Benchmarking deep networks for facial emotion recognition in the wild, *Multimedia Tools and Applications*, https://doi.org/10.1007/s11042-022-12790-7, 2022.
[27] G. Huang, Z. Liu, L. V. D. Maaten and K. Q. Weinberger, Densely connected convolutional networks, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1-9, 2018.
[28] Y. Nan, J. Ju, Q. Hua, H. Zhang and B. Wang, A-MobileNet: An approach of facial expression recognition, *Alexandria Engineering Journal*, vol.61, pp.4435-4444, 2022.
[29] Z. Zhao, Q. Liu and F. Zhou, Robust lightweight facial expression recognition network with label distribution training, *The 35th AAAI Conference on Artificial Intelligence (AAAI-21)*, pp.3510-3519, 2021.

Author Biography

Duong Thang Long is a lecturer of Information Technology major at Hanoi Open University. He received the Ph.D. degree of Information Technology from Vietnam Academy of Science and Technology (VAST) in 2011. His research interests are machine learning, artificial intelligence, deep learning, computer vision, fuzzy logic and soft computing with practical applications.