electronics

MDPI

Article

A Multi-View Face Expression Recognition Method Based on DenseNet and GAN

Jingwei Dong *, Yushun Zhang and Lingye Fan

School of Measurement and Communication Engineering, Harbin University of Science and Technology, Harbin 150080, China; 1605030129@stu.hrbust.edu.cn (Y.Z.); 1605030304@stu.hrbust.edu.cn (L.F.)
* Correspondence: djw@hrbust.edu.cn

Abstract: Facial expression recognition (FER) techniques can be widely used in human-computer interaction, intelligent robots, intelligent monitoring, and other domains. Currently, FER methods based on deep learning have become the mainstream schemes. However, these methods have some problems, such as a large number of parameters, difficulty in being applied to embedded processors, and the fact that recognition accuracy is affected by facial deflection. To solve the problem of a large number of parameters, we propose a DSC-DenseNet model, which improves the standard convolution in DenseNet to depthwise separable convolution (DSC). To solve the problem wherein face deflection affects the recognition effect, we propose a posture normalization model based on GAN: a GAN with two local discriminators (LD-GAN) that strengthen the discriminatory abilities of the expression-related local parts, such as the parts related to the eyes, eyebrows, mouth, and nose. These discriminators improve the model's ability to retain facial expressions and evidently benefits FER. Quantitative and qualitative experimental results on the Fer2013 and KDEF datasets have consistently shown the superiority of our FER method when working with multi-pose face images.

Keywords: facial expression recognition (FER); DenseNet; depthwise separable convolution (DSC); posture normalization; generative adversarial network (GAN)

check for updates

Citation: Dong, J.; Zhang, Y.; Fan, L. A Multi-View Face Expression Recognition Method Based on DenseNet and GAN. *Electronics* **2023**, *12*, 2527. https://doi.org/10.3390/electronics12112527

Academic Editor: George A. Tsihrintzis

Received: 19 April 2023
Revised: 29 May 2023
Accepted: 1 June 2023
Published: 3 June 2023

1. Introduction

With the rapid development of computer technology and artificial intelligence technology, the demand for human–computer interaction is increasingly strong. The realization of the understanding and recognizing of human facial expressions by computers is valuable in the domains of intelligent robotics, intelligent monitoring, virtual reality, medical assisted diagnosis, and so on. Benefiting from the improvement of computer performance, algorithms based on deep learning have become the mainstream scheme of FER.

In the Large Scale Visual Recognition Challenge (ILSVRC) 2012, the AlxNet model [1] based on a convolution neural network (CNN) greatly improved the accuracy of FER. Since then, deeper CNNs have been proposed, such as VGGNet [2], GoogleNet [3], and ResNet [4]. Girshick et al. used Region CNN (R-CNN) to learn facial expression features and achieved good results [5]. Ren et al. used Faster R-CNN to generate high-quality features for FER [6]. Yao et al. used the CReLU activation function in ResNet, increased the network depth while ensuring the recognition rate, and designed a residual block, so that the FER system could capture expression changes by learning features of different scales [7]. However, these deep learning network models were complex and had a large number of parameters, making them unsuitable for embedded computers and mobile devices.

Researchers began to study lightweight CNN networks. In 2017, the densely connected convolution network (DenseNet) was proposed by Huang et al. [8]. DenseNets have several compelling advantages: they alleviate the vanishing gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. SqueezeNet, a lightweight model proposed by Han, uses a lot of 1 × 1 convolution kernels and achieves accuracy similar to that of AlexNet with fewer parameters [9]. Aiming at running on mobile terminals, ShuffleNet, a lightweight model proposed in [10], adds group convolution to the networks, and this makes the model smaller and faster. In the past two years, researchers have proposed many lightweight models that continuously improve FER accuracy [11–15].

The FER algorithms mentioned above focus on the front face image, but since the facial deflection at various angles cannot be avoided in natural environments, the accuracy of FER is more or less lowered [16]. Therefore, pose normalization is performed before FER; i.e., the face is corrected to the front view in the case of deflection.

In order to correct face deflection, early researchers have proposed some 3D modeling methods. However, when the facial deflection angle is too large, the face normalization results of this type of methods are unsatisfactory. In 2014, Goodfellow et al. proposed the generative adversarial network (GAN) [17], which provides a new solution to the problem of missing features caused by face deflection. Up to now, a series of GAN variants have been developed to correct facial deflection. These GAN models for face normalization focus on preserving contour features in the process of synthesizing face images in order to facilitate identification. If the downstream task is FER, the synthesized frontal face does not meet the requirements very well because it does not focus on the preservation of local features related to facial expressions.

In this paper, we present an expression recognition method which combines the DenseNet FER model with the GAN-based posture normalization model. This method solves the problems of the large number of parameters in the FER model and low accuracy in multi-view face normalization. The contributions of this paper are summarized as follows:

1. A lightweight FER model, DSC-DenseNet, which reduces network parameters and computations by improving the standard convolution in DenseNet to DSC, is proposed. When the parameter is 0.16M, the FER rate of this model is 96.7% for frontal face input and 77.3% for profiles without posture normalization.
2. A posture normalization model, GAN, with two local discriminators (LD-GAN) based on the TP-GAN model, is proposed. The encoder–decoder structure implements a two-pathway generator, global pathway, and local pathway. In order to preserve more local features related to facial expressions in generated frontal faces, the discriminator was improved by adding two local discriminators besides the global discriminator to enhance its adversarial capability against the local pathway encoder. The loss functions are also improved to achieve better effects in network training.
3. The effectiveness of this method was verified on three public datasets. The validity of the lightweight FER model was verified on the CK+ and Fer2013 datasets, and the final effect of the combination of the posture normalization model and the FER model was verified on the KDEF dataset. Compared to the methods used in other representative models, this method effectively reduces the number of parameters of our model and has a higher FER rate (92.7%) under the condition of multi-angle deflection.

The remaining parts of this paper are organized as follows. Section 2 describes the previous related work. Section 3 describes the lightweight FER model and the posture normalization model that we propose in detail. Section 4 describes the experimental datasets, results, and related analysis. Section 5 gives conclusions and suggestions for future work.

2. Related Work

2.1. DenseNet

Although various CNN-based FER models have improved recognition rates, the consequent increase in the number of parameters has also resulted in more computational requirements. DenseNet is a model with a narrow network structure, as shown in Figure 1 [8].

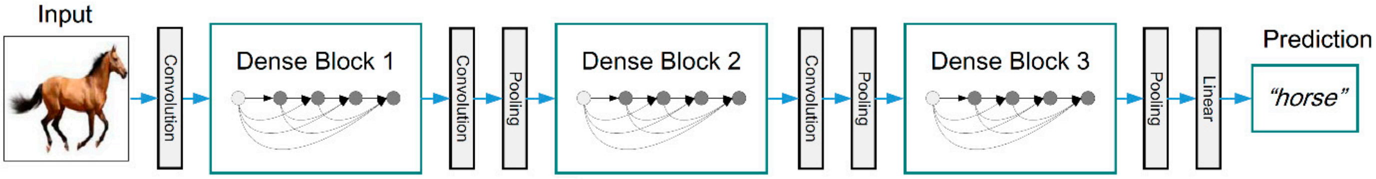

Figure 1. A deep DenseNet with three dense blocks.

DenseNet consists of dense blocks and transition layers, which lie between two adjacent blocks and change feature map sizes via convolution and pooling. The basic idea is that in a dense block, like in ResNet, direct connections from the preceding to the following layers are created. The difference in the dense block is that it establishes a dense connection between all the preceding layers to the followed layer; i.e., each layer takes all preceding feature-maps as input. The feature reuse of DenseNet improves the transmission ability of information throughout the entire network and reduces the number of parameters. To achieve the same accuracy as ResNet, DenseNet only needs about half of ResNet's parameters and half of its FLOPs (floating-point operations).

2.2. *Depthwise Separable Convolution (DSC)*

Compared to standard convolution, DSC has much lower parameters and computational complexity. Thus, it has been successfully applied to two well-known models, Xception [18] and MobileNet [19], by the Google team. DSC splits the computation of standard convolution into two steps: depthwise convolution, which applies a single convolutional filter per each input channel, and, pointwise convolution, which creates a linear combination of the output of the depthwise convolution. For example, the depthwise conv applies N convolution kernels of size M × M × 1 to N input channels of size W × H, achieves N feature maps of size W × H × 1, concatenates N feature maps, and achieves one feature map of size W × H × N. In other words, the depthwise conv has the same number of channels for input and output feature maps. However, there has been no connection between the different channels in the process so far. Then, the pointwise conv, by applying K standard convolutions of size 1 × 1 × N, solves this problem. It weights the feature map depthwise to generate a feature map of size W × H × K; i.e., it has the ability to fuse channels. The ratio of DSC to standard convolution is $1/K + 1/M^2$.

2.3. *GAN and Its Variants*

The GAN is a deep learning model, and its framework is shown in Figure 2a [20]. This framework corresponds to a minimax two-player game. The GAN consists of two models: a generative model *G* that captures the data distribution and a discriminative model *D* that estimates the probability that a sample came from the training data rather than from G. The aim of the training procedure for *G* is to maximize the probability of *D* making a mistake.

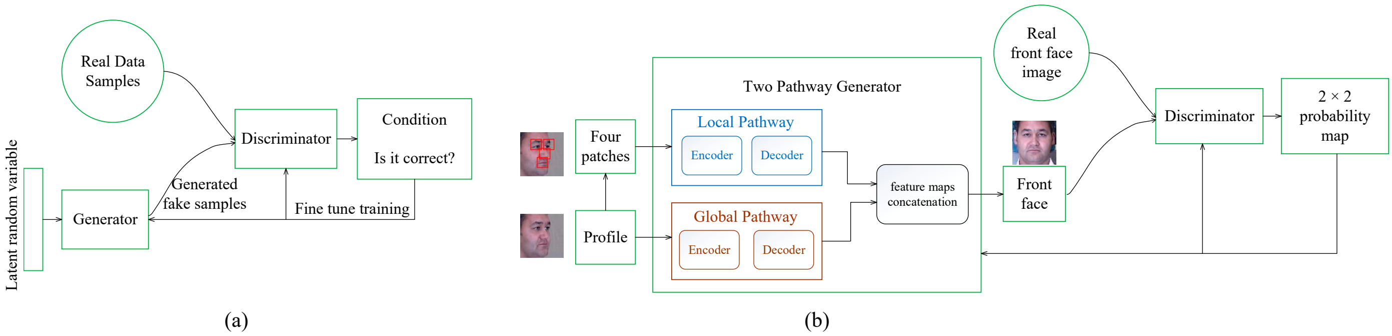

Figure 2. The frameworks of GAN and TP-GAN. (**a**) GAN framework; (**b**) TP-GAN framework.

Huang et al. proposed a two-pathway GAN (TP-GAN) [21] for photorealistic frontal view synthesis by simultaneously perceiving global structures and local details. As shown in Figure 2b, TP-GAN uses two pathways in G to perceive global structures and local details simultaneously. Four landmark-located patch networks, in addition to the commonly used global encoder–decoder network, are used to attend to local textures. Then, the positive synthesized image is used for the downstream task: identity recognition.

The training strategy and the loss function are challenging problems facing GANs. Combining a 3D-morphable model with a traditional GAN, FF-GAN [22] solves the problem of GANs being difficult to train by providing shape and appearance priors to guide the training on insufficient samples. Additionally, a new symmetry loss is introduced into the loss function. Similarly providing additional information to assist in training, the disentangled representation learning GAN (DR-GAN) [23] introduces a pose code to G and a pose estimation to D. Hu et al. proposed a couple-agent pose-guided GAN (CAPG-GAN) [24]. In the learning process for this network, the pose-guided G uses posture information provided by landmark heatmaps of input profile images and ground truth images. The couple-agent D essentially consists of two independent discriminators: one for rotation angle discriminating and the other for texture discriminating. Differing from the approaches above, Hardy et al. proposed a learning procedure for distributed GANs, MD-GAN [25], which can be trained over datasets that are spread across multiple servers.

3. Proposed Approach

3.1. Lightweight FER Model: DSC-DenseNet

3.1.1. The Framework of Dense Block

The lightweight FER model in this paper is based on DenseNet's feature reuse strategy, which is shown in Figure 3. In a dense block, the original feature x_0 is inputted into the layer, h_1, and x_1 is the output. The input of the layer h_2 includes not only x_1, the output from the layer h_1, but also the original feature x_0. The input of the layer h_3 includes not only x_2, the output of layer h_2, but also x_1 and x_0.

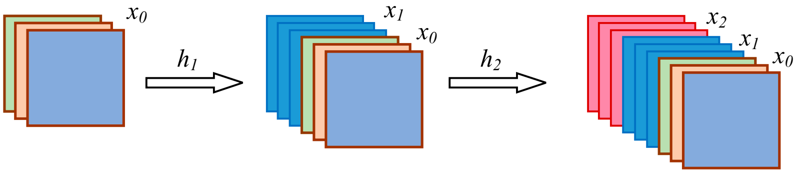

Figure 3. The sketch map of feature reuse in dense block.

To further lighten the FER model, we improve the feature map extracting in the dense block by using DSC instead of standard convolution to simplify the calculation.

The non-linear transformation function H(·) in dense block is defined as batch normalization(BN) + rectified linear unit (ReLU) +3 × 3 convolution. As the number of layers increases, the number of input channels increases dramatically with the number of overlapping feature maps. For this reason, a 1 × 1 conv is used before the 3 × 3 conv to limit the number of input channels. Then H(·) is defined as BN + ReLU + 1 × 1 Conv + BN + ReLU + 3 × 3 Conv. Assume that the input size of layer i in a dense block is 48 × 48, the feature maps of all the preceding layers are concatenated with N channels, the bottleneck layer reduces the number of channels to 128, and the growth rate k (the number of output channels per layer; i.e., the increase number of input channels in the next layer after concatenation) is 32. Then, the operation of layer i is as shown in Figure 4.

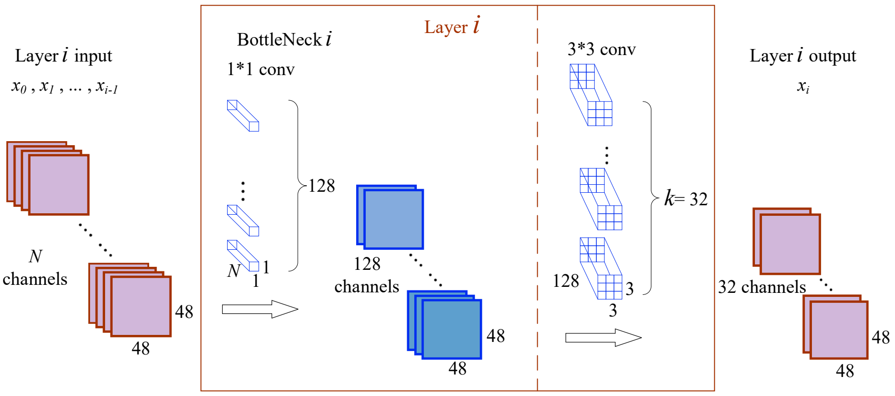

Figure 4. The framework of layer *i* in a dense block of DenseNet. 1 × 1 and 3 × 3 are the kernel sizes.

After replacing standard convolution with DSC, H(·) is defined as BN + ReLU + 3 × 3 DSC. Then, the operation of layer *i* is as shown in Figure 5. The number of pointwise conv, i.e., the number of channels of the feature map output by DSC, is the growth rate *k* of DenseNet.

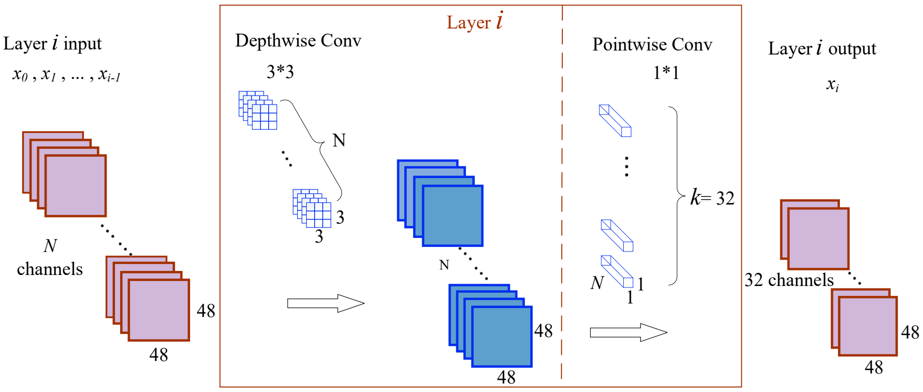

Figure 5. The framework of layer *i* in a dense block of DSC-DenseNet. 1 × 1 and 3 × 3 are the kernel sizes.

3.1.2. The Architecture of DSC-DenseNet

Because of its use of DSC, we refer to this network, which is shown in Figure 6, as DSC-DenseNet. The parameters of its components are given in Table 1.

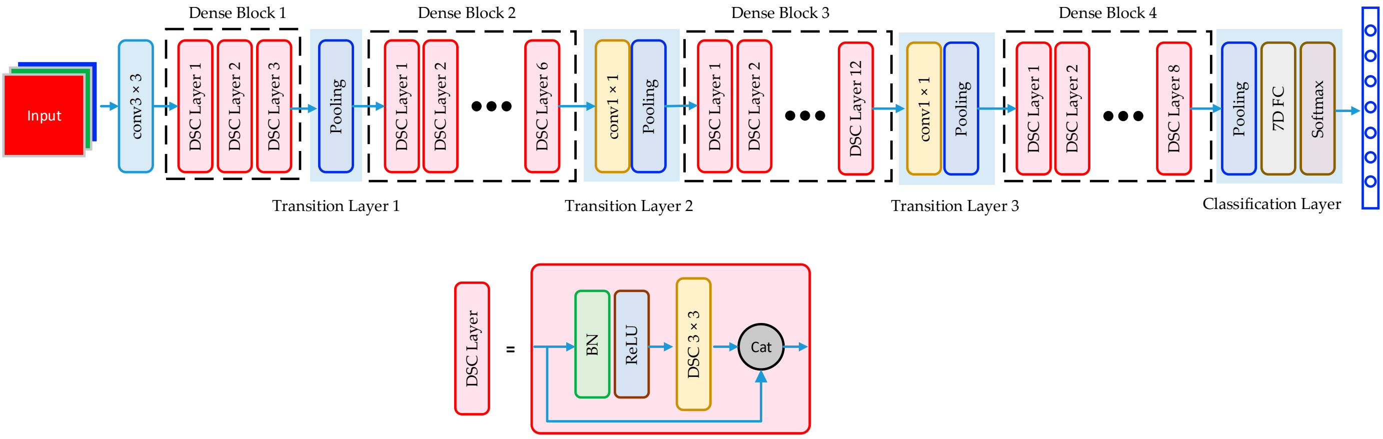

Figure 6. The framework of DSC-DenseNet.

Table 1. The architecture of DSC-DenseNet. The growth rate is k = 32.

Layers	Operator	Output Size	Output Channels
Convolution	3 × 3 conv, stride 1	48 × 48	32
Dense Block1	3 × 3 DSC × 3 layers	48 × 48	32 + 32 × 3 = 128
Transition Layer1	2 × 2 average pool, stride 2	24 × 24	128
Dense Block2	3 × 3 DSC × 6 layers	24 × 24	128 + 32 × 6 = 320
Transition Layer2	1 × 1 × 128 conv	24 × 24	128
	2 × 2 average pool, stride 2	12 × 12	128
Dense Block3	3 × 3 DSC × 12 layers	12 × 12	128 + 32 × 12 = 512
Transition Layer3	1 × 1 × 256 conv	12 × 12	256
	2 × 2 average pool, stride 2	6 × 6	256
Dense Block4	3 × 3 DSC × 8 layers	6 × 6	256 + 32 × 8 = 512
Classification Layer	6 × 6 global average pool	1 × 1	-
	7D fully-connected, softmax	-	-

DSC-DenseNet consists of four dense blocks, three transition layers, and a classification layer. The four dense blocks contain three, six, twelve, and eight DSC layers, respectively. ReLU is used as the activation function. The transition layer uses 2 × 2 average pooling. If too many channels are output by the previous dense block, a 1 × 1 conv is added to reduce the number of channels and thus simplify operations. Examples of this addition include the 1 × 1 × 128 conv in transition layer 2 and the 1 × 1 × 256 conv in transition layer 3. In this situation, the transition layer is BN + 1 × 1 Conv + 2 × 2 average-pooling. Finally, the classification layer realizes the recognition of seven major types of facial expressions through SoftMax multiple classifiers.

Assuming that the input feature map's size and the channel of the convolution layer are $H_i \times W_i \times C_i$, and those of the output feature map are $H_o \times W_o \times C_o$, the convolution kernel of depthwise conv is $H_k \times W_k \times C_i \times 1$ and FLOPs is $H_k \times W_k \times C_i \times H_i \times W_i$, while the convolution kernel of pointwise conv is $1 \times 1 \times C_i \times C_o$ and FLOPs is $C_i \times C_o \times H_i \times W_i$. After summing two computational quantities and dividing them by that of standard convolution, a fraction can be obtained: $1/C_o + 1/H_k W_k$. As the number of feature maps increases, $1/C_o$ can be ignored. The size of the depthwise convolution kernel determines the computational quantity. Due to the use of DSC with 3 × 3 convolution kernels, the computational complexity in a dense block can be reduced to $1/32 + 1/3^2 \approx 14.2\%$. Due to the other layers in DSC-DenseNet, the computational complexity can actually be reduced to about 30%.

3.2. Frontal Face Normalization Model: GAN with Two Local Discriminators (LD-GAN)

Facial pose variations still remain a great challenge for FER models, especially for lightweight ones that sacrifice some accuracy. Therefore, facial pose normalization is a commonly adopted step. Synthesizing a frontal face from a profile image is a highly non-linear transformation.

3.2.1. The Framework of LD-GAN

Based on the idea of a two-pathway generator for TP-GAN, we propose LD-GAN, whose framework is shown in Figure 7. A global pathway is used to process facial contour features and a local pathway is used to process facial expression features in G. Two local feature discriminators are added in D to enhance the adversarial operation between G and D.

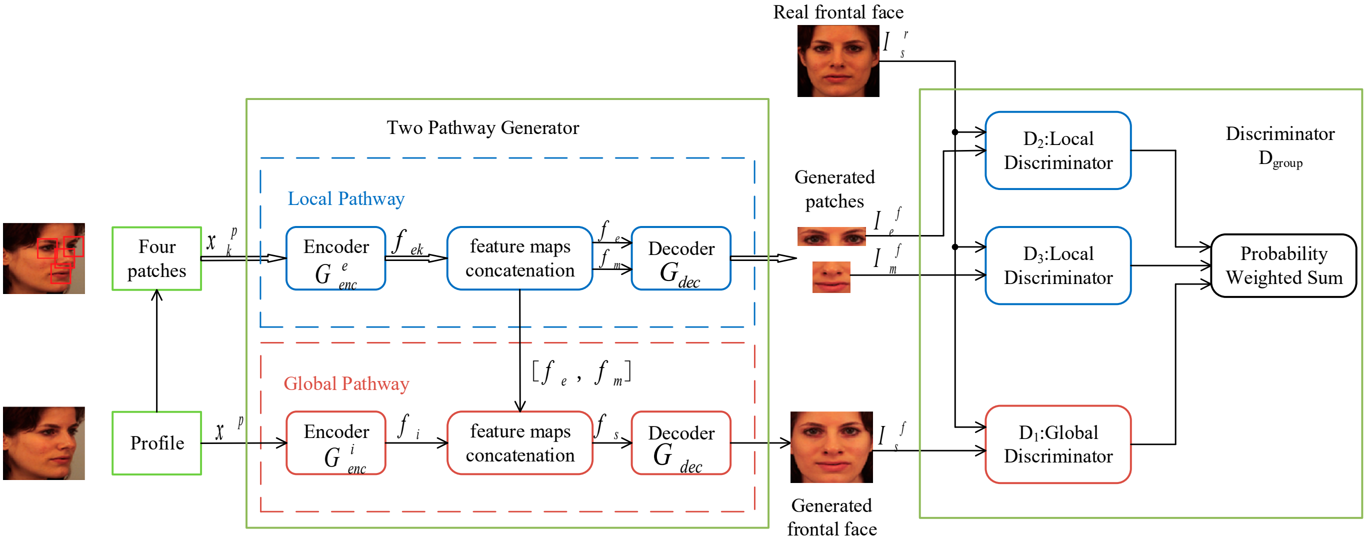

Figure 7. The framework of LD-GAN.

The profile x^p is input into the global pathway and the global encoder G_{enc}^i extracts the global contour features $f_i = G_{enc}^i(x^p)$. After landmarked allocating and cropping, four local patches of eyes, noses, and mouths x_k^p are input into the local pathway, and the local encoder G_{enc}^e extracts the facial expression features $f_{ek} = G_{enc}^e({x_k}^p)$. The purpose of cropping out expression-related parts is to reduce image noise by eliminating pixels that are less associated with expressions and to force G_{enc}^e to focus on extracting expression features. This way, the generated image can preserve the original expression better. To fuse the information from all pathways, feature maps need to be concatenated together, but only if they have the same spatial resolution. Thus, we first fuse two feature maps from eye patches together, specifically according to their landmarks, and then fuse those of the nose and mouth. At last, we obtain f_s simply by concatenating the global pathway feature map f_i with the two fused feature maps, f_e and f_m. f_s is decoded by G_{dec} and a frontal face I_s^f is generated. In the local pathway, f_e and f_m are decoded by G_{dec} and two frontal patches, eye patch I_e^f and mouth-nose patch I_e^f, are achieved in order to feed D. The reason we use only two patches instead of four is to simplify the framework of D.

Inspired by multi-discriminator strategy [26,27], we propose a D_{group} including a global discriminator D_1 and two local discriminators, eye discriminator D_2 and mouth-nose discriminator D_3. The input of D_{group} includes four images: the real frontal face I_s^r, generated frontal face I_s^f, generated eye patch I_e^f, and generated mouth-nose patch I_m^f. The next step is to combine these four images into three image pairs, $\left(I_s^r, I_s^f\right)$, $\left(I_s^r, I_e^f\right)$, and $\left(I_s^r, I_m^f\right)$, then input them into D_1–D_3 respectively.

3.2.2. The Architecture of LD-GAN

As a key component for extracting features from facial images, the core of the encoder is a CNN. We use Light-CNNs as encoders in both global and local pathways because of their advantages, i.e., having fewer parameters and better robustness [28]. As shown in Table 2, G_{enc}^i and G_{enc}^e have the same architecture of Conv0, Conv1–Conv4, fc1, and fc2. The activation function of Conv1–Conv4 is Maxout. The size of the input RGB image is 128 × 128 and the size of the final feature map obtained is 1 × 1 × 256. The decoder G_{dec} is a deconvolution neural network. The deconvolution process has no learning ability and can only visualize global contour or local expression features. The network architecture of G_{dec} is shown in Table 3.

Table 2. The architecture of the encoder.

Layer	Kernel/Stride	Output
Conv0	7 × 7/1	128 × 128 × 64
Conv1	5 × 5/2	64 × 64 × 64
Conv2	3 × 3/2	32 × 32 × 128
Conv3	3 × 3/2	16 × 16 × 256
Conv4	3 × 3/2	8 × 8 × 512
fc1	-	512
fc2	-	256

Table 3. The architecture of the decoder.

Layers	Kernel	Output
fc reshape	-	8 × 8 × 64
Deconv0	3 × 3	16 × 16 × 64
Deconv1	3 × 3	32 × 32 × 32
Deconv2	3 × 3	64 × 64 × 16
Deconv3	3 × 3	128 × 128 × 16
Deconv4	3 × 3	128 × 128 × 3

The architecture of D_1 is shown in Table 4. The size of the generated frontal face input into D_1 is 128 × 128 × 3, and it is changed to 64 × 64 × 64 after the Conv0 layer and to 1 × 1 × 1024 after the Conv1–Conv5 layers. The architectures of D_2 and D_3 are totally same to that of D_1 except for the sizes of input images. The size of the eye patch is 95 × 20, and that of the mouth-nose patch is 50 × 75.

Table 4. The architecture of the D_1.

Layer	Kernel/Stride	Output
Conv0	4 × 4/2	64 × 64 × 64
Conv1	4 × 4/2	32 × 32 × 128
Conv2	4 × 4/2	16 × 16 × 256
Conv3	4 × 4/2	8 × 8 × 512
Conv4	4 × 4/2	4 × 4 × 1024
Conv5	4 × 4/1	1 × 1 × 1024

3.2.3. The Loss Function Improved

We have improved the loss function of LD-GAN. Content loss L_{con} is added on the basis of adversarial loss L_{ck}. L_{con} consists of pixel loss L_P and symmetric loss L_S. Then, the loss function of G is:

$$L_G = L_{ck} + L_{con} = L_{ck} + L_P + L_S \quad (1)$$

L_P is the difference in pixels between a generated frontal face and an input profile image:

$$L_p = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \left| (I_s^f)_{i,j,k} - x_{i,j,k}^p \right| \quad (2)$$

The smaller L_P is, the closer the quality of the generated image will be to that of the input image, and so, L_P should be minimized. L_S is the Manhattan distance between the left and right sides of the generated frontal face:

$$L_s = \frac{1}{W/2 \times H} \sum_{i=1}^{W/2} \sum_{j=1}^{H} \left| (I_s^f)_{i,j,k} - (I_s^f)_{W-i,j,k} \right| \quad (3)$$

Calculating L_S accelerates the convergence of G. The adversarial loss is:

$$L_{ck} = lnD_k[(I_s^r)_k] + ln[1 - D_k(I_s^f)_k] \tag{4}$$

As D's ability to discriminate between true and false improves via training, G needs to compete against it to minimize its probability of discrimination. Thus, D needs to minimize L_{ck} for G. To sum up, the loss function of LD-GAN generator is:

$$L_G = \beta_1 L_p + \beta_2 L_s + \beta_3 L_{ck} \tag{5}$$

where β_1, β_2, and β_3 are the weights that affect the loss, and can be adjusted during training to achieve the best training results.

The D of LD-GAN does not involve content loss, and its training process only includes adversarial loss, which requires a weighted sum of three adversarial losses. The loss function of D_{group} is:

$$L_{D_{group}} = \omega_1 L_{c1} + \omega_2 L_{c2} + \omega_3 L_{c3} \tag{6}$$

where ω_1, ω_2 , and ω_3 are weighing hyper parameters. D identifies the true or false images generated by G and obtains a probability that the images will be judged as false; thus, D needs to maximize the adversarial loss.

4. Experimental Results and Discussions

The experiments were carried out on the Windows 10 operating system and the recognition methods were implemented using the Python language and PyTorch library. The experimental environment included an Intel(R) Core (TM) i7-10750H CPU @ 2.60 GHz processor, 16 GB memory, and GeForce GTX 1650Ti graphics card.

The effectivenesses of the proposed DSC-DenseNet and LD-GAN were verified on three public datasets. The final results of the combination of DSC-DenseNet and LD-GAN are demonstrated below.

4.1. Datasets

In experiments, we used the following public datasets:

Extended Cohn-Kanade Dataset (CK+) [29]: It is one of the most widely used expression datasets, and was released in 2010. There are 593 sequences in it, and each sequence begins with a neutral expression and proceeds to a peak expression. FER based on a static image often takes the last frames as samples. The eight included expressions are disgust, happiness, surprise, fear, anger, contempt, sadness, and neutral, as shown in Figure 8.

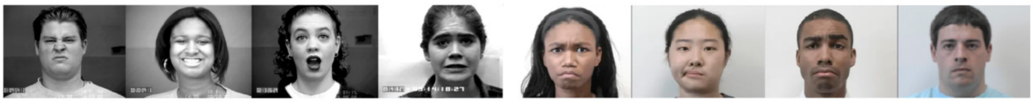

Figure 8. Eight expressions in the CK+ dataset.

Kaggle FER challenge dataset (Fer2013) [30]: All 35,887 examples are 48 × 48 gray images. The training set consists of 28,709 examples, and the validation and test set consists of 3589 examples. When compared to CK+, there are seven of the same expressions in Fer2013, except for contempt, as shown in Figure 9. Moreover, the examples in Fer2013 have deflection at different angles.

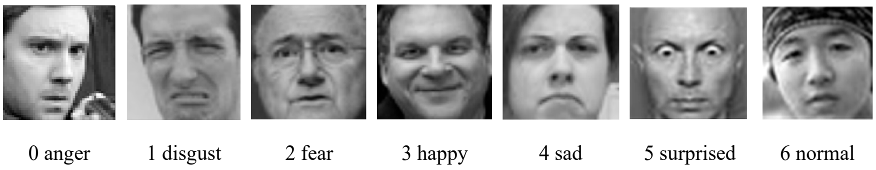

Figure 9. Seven expressions in the Fer2013 dataset.

Karolinska directed emotional faces (KDEF) [31]: It includes 4900 GRB images of size 562 × 762 in seven expressions. When compared to Fer2013, every expression in this dataset is represented with five different views, −90°, −45°, 0°, +45°, and +90°, as shown in Figure 10.

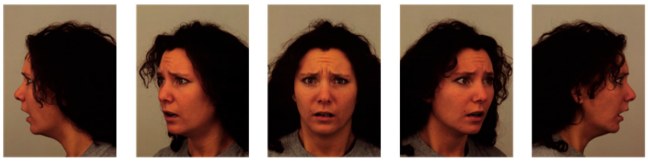

Figure 10. Samples at five views in the KDEF dataset.

4.2. *Preprocessing*

To ensure the effect of FER, we preprocessed the original images before experiments, including the data on face detection and alignment. We used the multitask cascaded convolutional networks (MTCNN) to hasten face detection and alignment [32,33].

We processed the last three frames in the labeled CK+ dataset expression sequences to 48 × 48 grayscale images. Since there were too few samples of contempt, and also in order to match the seven basic expressions across the datasets, contempt samples from CK+ were excluded from the experiments. Then, we expanded the number of samples to ten times their original number by using common methods for data augmentation such as scale augmentation, changing contrast and changing brightness, and flipping from left to right. To preserve an expression, the central area of the image in question should be maintained. Thus, random cropping or severe rescaling was not adopted by us. We found 10,500 samples from seven expression categories in total. In Fer2013, the numbers of most samples are much larger than those in CK+, but the former has insufficient disgust samples. Thus, we expanded the disgust class, and samples that did not contain faces or had severe facial occlusion were excluded. In KDEF, the sample size of each expression in every view is the same (140). We expanded this to 1400, and so the total number of utilized samples was 49,000.

4.3. *Experimental Results of DSC-DenseNet*

4.3.1. Experiments for Effectiveness of DSC-DenseNet

The network parameters during training were as follows: the epoch was 150 on CK+ and 250 on Fer2013, batch size was 32, initial learning rate was 0. 01, and learning rate decreased to 50% after each 8 epochs. We used 250 images from each expression category for testing on CK+ and 4000 images from all expression categories for testing on Fer2013. The confusion matrixes of FER percents are shown in Figures 11 and 12.

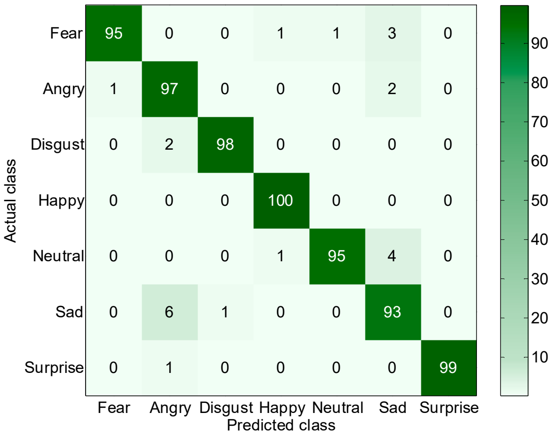

Figure 11. The confusion matrix of FER on the CK+ dataset.

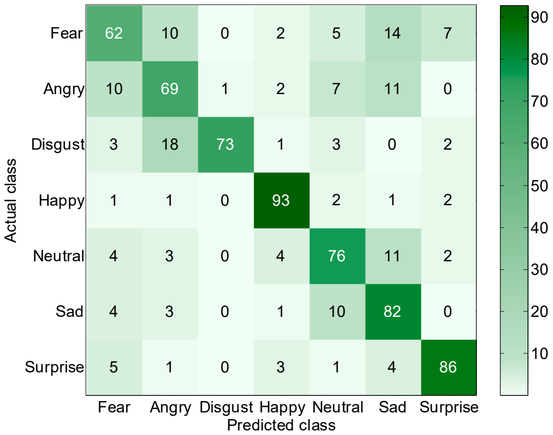

Figure 12. The confusion matrix of FER on the Fer2013 dataset.

The mean of the recognition rates on seven expressions, $R_{expression}$, was used as the final evaluation metric, and it was sometimes abbreviated as 'recognition rate' without causing confusion. It was defined thus:

$$Recognition\ rate = \frac{1}{7}\left(R_{Anger} + R_{Disgust} + \ldots + R_{Neutral}\right) \tag{7}$$

The experimental results showed that:

1. On the CK+ dataset, the recognition rate of our model was 96.7%. The percentages of recognition for the happiness, surprise, and disgust classes were the highest: 100%, 99%, and 98%, respectively. The reason for the good recognition results for the happiness class was that the features of happiness were more obvious than other emotions and thus it was not easily confused with other features. These results showed the same performance as other existing FER methods. The recognition rate for the sadness class was the lowest—92%—and 6% of sad expressions were misclassified as angry. The reason they were easily confused was that they both more or less involved frowning. Neutral and fear expressions were misclassified as sad in 4% and 3% of cases, respectively.
2. On the Fer2013 dataset, the recognition rate of our model was 77.3%. The percentages of recognition for the happiness and surprise classes were 93% and 86%, respectively. The recognition rates for the fear and anger classes were the lowest: 62% and 69%, respectively. The main classes that were confused with the fear class were sadness and anger, while the main classes that were confused with the anger class were fear and sadness.
3. For CK+, the frontal faces dataset, the recognition rate of our model could meet the practical requirements. For Fer2013, the dataset with profile faces, the recognition rate of our model was significantly reduced. Part of the reason for this was that facial occlusion affected recognition to some extent, although severely occluded samples were removed. Another reason was that there were multi-view images in the dataset, and facial deflection significantly affected the effectiveness of FER. This has also been the consensus among researchers, and it also indicates the necessity of studying facial pose normalization models in this paper.

4.3.2. Comparison of DSC-DenseNet with Other Lightweight Models

We performed comparison experiments on Fer2013 to compare our model to the classical lightweight classification models (SqueezeNet, ShuffleNet, ResNet, and MobileNet) and the state-of-art classification models (Separate-loss and RAN). The learning curves of some models are shown in Figure 13. We also compared our model to recently proposed FER models (Light-SE-ResNet and PGC-DenseNet). The FER recognition rate, params, and FLOPs of these models are shown in Table 5 (sorted by recognition rate).

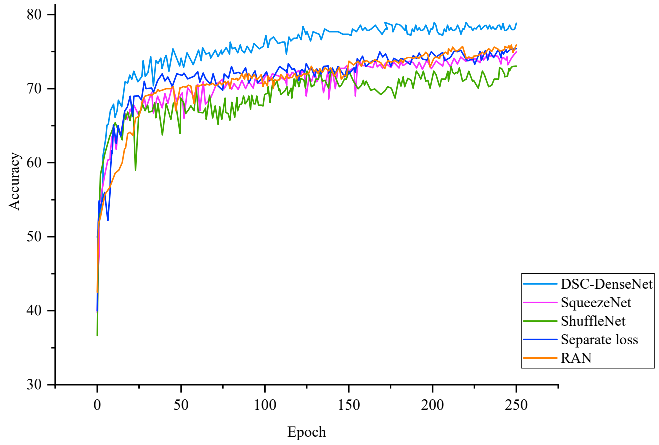

Figure 13. The learning curves of some lightweight models on the Fer2013 dataset.

Table 5. Comparison of lightweight models on the Fer2013 dataset.

Lightweight Model	Params (Million)	FLOPs (Billion)	Recognition Rate (95%CI)
MobileNetV3 (Large) [34]	4.2	0.02	63.9% ± 1.49%
Light-SE-ResNet [35]	5.1	0.06	67.4% ± 1.45%
ResNet18 [4]	11.2	0.09	69.3% ± 1.43%
ResNet50	23.5	0.21	70.1% ± 1.42%
ShuffleNet [10]	1.25	0.04	73.0% ± 1.38%
PGC-DenseNet [36]	0.26	0.12	73.3% ± 1.37%
SqueezeNet [9]	0.47	0.02	75.0% ± 1.34%
Separate-loss [37]	1.1	0.18	75.4% ± 1.33%
RAN [38]	1.2	0.14	76.0% ± 1.32%
DSC-DenseNet (Ours)	0.16	0.16	77.3% ± 1.30%

The experimental results showed that:

1. The FER recognition rate of our model on multi-view faces was 77.3% when the params value was 0.16M. Compared to the models with better accuracy (the lower half of Table 5), such as SqueezeNet and ShuffleNet, our model had higher FLOPs because of its concatenation of feature maps. However, its recognition rate was 2.3% and 4.3% higher than that of SqueezeNet and ShuffleNet, respectively. This could also be seen visually in their learning curves. On the other hand (the upper half of Table 5), it was shown that at the cost of accuracy, speed could be significantly improved. The FLOPs of MobileNetV3 and Light-SE-ResNet were extremely small.
2. Our model had a smaller params value to achieve approximate accuracy. Compared to Separate-loss and RAN, the two models with the closest accuracy to ours, the params value of DSC-DenseNet was equal to only about 15% of their params values, and the FLOPs value of DSC-DenseNet was between theirs. Therefore, our model achieved a practical recognition rate, meaning that the lightweight FER model proposed in this paper achieved a balance between the accuracy and performance requirements of the hardware platform.

4.4. Experimental Results of LD-GAN

4.4.1. Training Strategy

Compared to the true or false discrimination task in G, *D*'s true or false discrimination task was more difficult to train. In order to achieve a dynamic balance between the performances of *G* and D, the update frequency ratio between *G* and *D* was 1:2 in training. At the same time, a small learning rate given to *D* slowed down its convergence and avoided the ability unbalance between *D* and G, effectively preventing G's loss from increasing continuously and keeping its internal parameters from improving in the desired direction.

During the training process, we could not know when the performance of GAN would be optimal. Too much training may have produced negative effects which could have made the parameters unstable and damaged the effectiveness of the original model. Therefore, it was necessary to store the parameters when the model achieved excellent performance during training. After a certain number of epochs in the learning process, 10 random face images in the test set were compared with the generated faces. If the difference between two images increased abruptly during the process, the training was terminated because this would have indicated that the previously stable parameters had been destroyed.

During the training process, *G* and *D* were optimized using Adam optimizer. the learning rates for *G* and *D* were set to 0.001 and 0.0005. For each epoch, the learning rate was adjusted as an exponential decay with a decay parameter of 0.999. The maximum number of epochs was set to 500 and the batch size was set to 128. In order to generalize the network better, label smoothing was used; i.e., the labels 0 and 1 were replaced by random numbers in the range of 0–0.1 and 0.8–1 when the true or false judgment was made.

4.4.2. Experiments for Effectiveness of LD-GAN

We performed an experiment on the KDEF dataset, and the loss curve for this is shown in Figure 14. From the trend of the loss curve, it is clear that the training strategy used on LD-GAN was effective. The result of the posture normalization is shown in Figure 15.

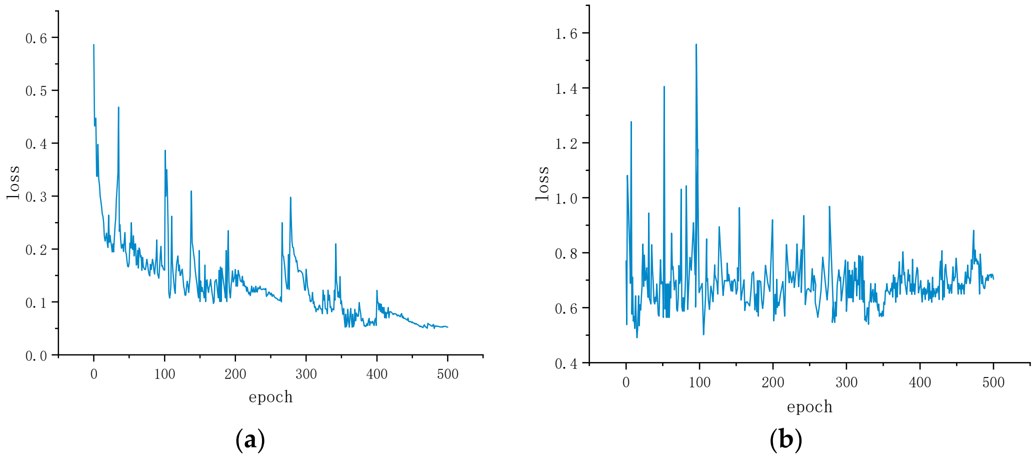

Figure 14. Training loss curves on the KDEF dataset. (**a**) Content loss L_{con}; (**b**) adversarial loss L_{ck}.

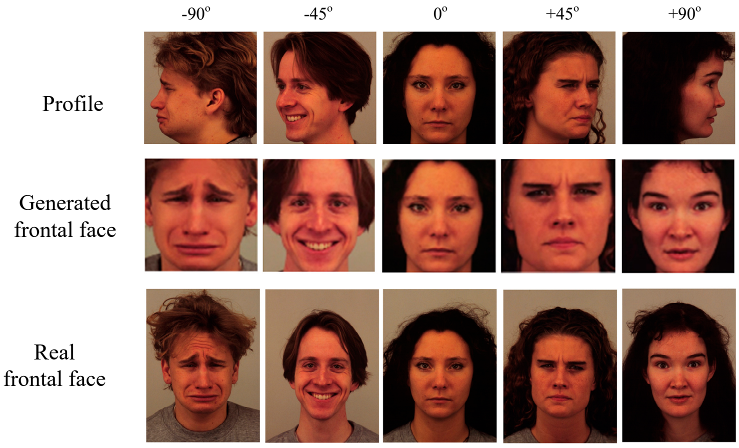

Figure 15. The result of the posture normalization on KDEF dataset.

In the study, the frontal images generated by LD-GAN were recognized by DSC-DenseNet. The mean FER recognition rate of each expression, at all views, was calculated, and the confusion matrix of seven facial expressions was derived, as is shown in Figure 16.

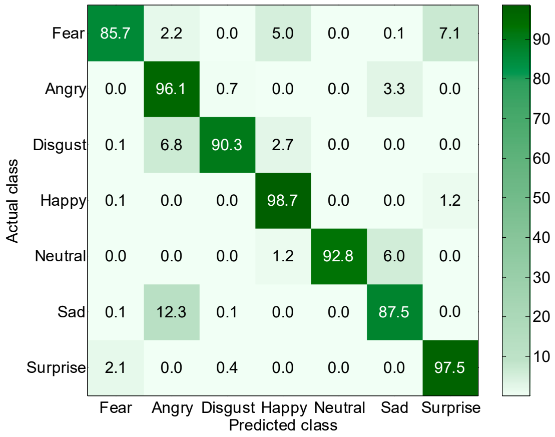

Figure 16. The confusion matrix of recognition rates.

The experimental results showed that:

1. The recognition rate for happy expressions was the highest: 98.7%. The rates of recognition for the surprise and anger classes were the second highest; these amounted to more than 96%. The recognition rates for the fear and sadness classes were lower: 85.7% and 87.5%, respectively. The mean recognition rate for the seven expressions was 92.7%.
2. When compared to the results without pose normalization (Figure 12), the method proposed significantly improved the recognition rate, and could also reduce the misclassification rate between different expressions to a lower level, thus meeting the needs of practical application.

4.4.3. Comparison of Our Model with Others

We performed comparison experiments on the KDEF dataset. The profiles in the ±45° and ±90° views were corrected to frontal faces by three GAN variants (FF-GAN, TP-GAN, and DR-GAN) and our model, and then FER was performed using DSC-DenseNet. We used 5-fold cross validation, and the results of this are shown in Table 6.

Table 6. Comparison of FER recognition rates, with 95% Cis, of the four models on the KDEF dataset.

View **Model**	**±90°**	**±45°**
FF-GAN [22]	83.5% ± 1.16%	89.6% ± 0.96%
TP-GAN [21]	83.6% ± 1.16%	88.9% ± 0.98%
DR-GAN [23]	85.4% ± 1.11%	90.7% ± 0.91%
LD-GAN (ours)	88.8% ± 0.99%	93.2% ± 0.79%

The experimental results showed that the FER rate of our method was 5.3%, 5.2%, and 3.4% higher than that of FF-GAN, TP-GAN, and DR-GAN, respectively, at ±90° views. At ±45° views, the FER rate of our method was 3.6%, 4.3%, and 2.5% higher than that of each of the three models, respectively.

4.4.4. Ablation Study

In order to generate facial images with more expression features, we used two pathways in G to generate local features that were closely related to the selected expressions. Two Local discriminators were added to D to preserve the local details and to improve the accuracy of expression classification. To verify the validity of our model, an ablation experiment was performed on the KDEF dataset.

To compare our model to LD-GAN, two Local discriminators were removed from the model. The normalization effects of the two models are illustrated in Figure 17. Figure 17a shows a sad-faced man's profile at a −90° view and a neutral-faced woman's profile at a +45° view. Compared to the real frontal faces in Figure 17d, the man's mouth does not show the drop it should have, and the woman's mouth in Figure 17b does show an excessive rise in the images generated by GAN without the Local discriminators. These errors increase the probability that the man's sad expression would be misclassified as neutral and the woman's neutral expression would be misclassified as happy in subsequent FER steps. In Figure 17c, which shows the faces generated by LD-GAN, the corners of the subjects' mouths are closer to those in the real frontal faces.

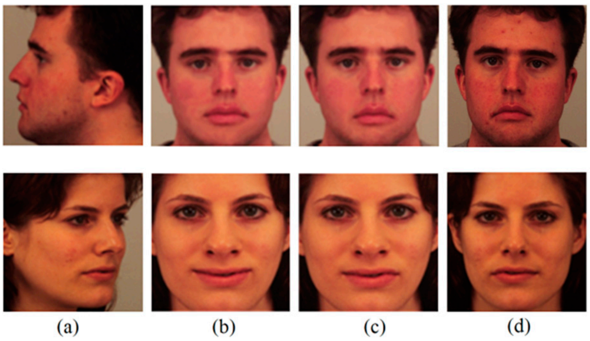

Figure 17. Comparison of face normalizations between two models. (**a**) Input profiles; (**b**) faces generated by GAN without Local discriminator; (**c**) faces generated by LD-GAN; (**d**) real frontal faces.

5. Conclusions

The lightweight DSC-DenseNet FER model proposed in this paper minimizes the number of parameters and the complexity of computation and achieves a useful FER rate. The LD-GAN face posture normalization model proposed improves the ability to reserve local features related to expression and can generate a face that is more conducive to FER. Experiments on multiple datasets show that the recognition rate for faces at multi-view is higher than 92% when combining LD-GAN and DSC-DenseNet.

In future research, we will investigate the impact of different lighting environments and local occlusion on our model and establish a lightweight FER method for natural scenes.

Author Contributions: Conceptualization, J.D. and Y.Z.; methodology, Y.Z., J.D. and L.F.; software, validation, Y.Z. and L.F.; writing—review and editing, J.D. and Y.Z.; supervision, J.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China under Grant NSFC-61671190.

Data Availability Statement: The code and data presented in this study are available at https://gitee.com/djw_Hrbust/fer.git (accessed on 1 June 2023).

Acknowledgments: We thank for Jinyang Li and Ziqi Zhao for their valuable comments and discussion.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
2. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-scale Image Recognition. *Comput. Sci.* **2014**, *56*, 1–14.
3. Szegedy, C.; Liu, W.; Jia, Y. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.

4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
7. Yao, A.; Cai, D.; Hu, P. HoloNet: Towards Robust Emotion Recognition in the Wild. In Proceedings of the 18th ACM International Conference on Multimodal Interaction, Tokyo, Japan, 12–16 November 2016.
8. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
9. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level Accuracy with 50× Fewer Parameters and <0.5 MB Model Size. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
10. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. Shufflenet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
11. Shen, H.; Meng, Q.; Liu, Y. Facial Expression Recognition by Merging Multilayer Features of Lightweight Convolution Networks. *Laser Optoelectron. Prog.* **2021**, *58*, 148–155.
12. Li, C.; Lu, Y. Facial expression recognition based on depthwise separable convolution. *Comput. Eng. Des.* **2021**, *42*, 1448–1454.
13. Gao, J.; Cai, Y.; He, Z. TP-FER: Facial expression recognition method of tri-path networks based on optimal convolutional neural network. *Appl. Res. Comput.* **2021**, *38*, 7.
14. Liang, H.; Lei, Y. Expression Recognition with Separable Convolution Channel Enhancement Features. *Comput. Eng. Appl.* **2022**, *58*, 184–192.
15. Zhang, P.; Kong, W.; Teng, J. Facial Expression Recognition Based on Multi-scale Feature Attention Mechanism. *Comput. Eng. Appl.* **2022**, *58*, 182–189.
16. Han, Z. Facial Expression Recognition under Various Facial Postures. Master's Thesis, Soochow University, Suzhou, China, 2020.
17. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Neural Information Processing Systems, Montreal, QC, Canada, 8–11 December 2014.
18. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
19. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
20. Generative Adversarial Network (GAN). Available online: https://www.geeksforgeeks.org/generative-adversarial-network-gan/ (accessed on 31 March 2023).
21. Huang, R.; Zhang, S.; Li, T.; He, R. Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
22. Yin, X.; Yu, X.; Sohn, K.; Liu, X.; Chandraker, M. Towards Large-Pose Face Frontalization in the Wild. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
23. Tran, L.; Yin, X.; Liu, X. Disentangled Representation Learning GAN for Pose-invariant Face Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
24. Hu, Y.; Wu, X.; Yu, B.; He, R.; Sun, Z. Pose-guided Photorealistic Face Rotation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
25. Hardy, C.; Le Merrer, E.; Sericola, B. MD-GAN: Multi-Discriminator Generative Adversarial Networks for Distributed Datasets. In Proceedings of the IEEE International Parallel and Distributed Processing Symposium, Rio de Janeiro, Brazil, 20–24 May 2019.
26. Lin, H.; Ma, H.; Gong, W.; Wang, C. Non-frontal Face Recognition Method with a Side-face-correction Generative Adversarial Networks. In Proceedings of the 3rd International Conference on Computer Vision, Image and Deep Learning, Changchun, China, 20–22 May 2022.
27. Iizuka, S.; Simo-serra, E.; Ishikawa, H. Globally ang Locally Consistent Image Completion. *ACM Trans. Graph.* **2017**, *36*, 1–14. [CrossRef]
28. Sharma, A.K.; Foroosh, H. Slim-CNN: A Light-Weight CNN for Face Attribute Prediction. In Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition, Buenos Aires, Argentina, 18–22 May 2020.
29. Extended Cohn-Kanade Dataset. Available online: http://www.pitt.edu/~emotion/ck-spread.htm (accessed on 19 April 2023).
30. Kaggle FER Challenge Dataset. Available online: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/overview (accessed on 19 April 2023).
31. Karolinska Directed Emotional Faces. Available online: http://www.emotionlab.se/kdef (accessed on 19 April 2023).
32. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2017**, *23*, 1499–1503. [CrossRef]

33. Xia, H.; Li, C. Face Recognition and Application of Film and Television Actors Based on Dlib. In Proceedings of the International Congress on Image and Signal Processing, Biomedical Engineering and Informatics, Suzhou, China, 19–21 October 2019.
34. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
35. Xie, Y. Research and Application of Facial Expression Recognition Method Based on Deep Learning Architecture. Master's Thesis, Wuhan Institute of Technology, Wuhan, China, 25 May 2022.
36. Yang, H. Facial Expression Recognition Method Based on Lightweight Convolutional Neural Network. Master's Thesis, Beijing University of Civil Engineering and Architecture, Beijing, China, 30 May 2020.
37. Li, Y.; Lu, Y.; Li, J.; Lu, G. Separate Loss for Basic and Compound Facial Expression Recognition in the Wild. In Proceedings of the Eleventh Asian Conference on Machine Learning, WINC AICHI, Nagoya, Japan, 17–19 November 2019.
38. Wang, K.; Peng, X.; Yang, J.; Meng, D.; Qiao, Y. Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition. *IEEE Trans Image Process.* **2020**, *29*, 4057–4069. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.